Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»
Τεχνολογίες Ενσωματωμένων Υπολογιστικών Συστημάτων

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Επιτάχυνση εφαρμογών Μηχανικής Όρασης με χρήση επεξεργαστή γραφικών** **Acceleration of Computer Vision applications using Graphics Processing Unit**
Ονοματεπώνυμο Φοιτητή	**Χρήστος Παπαζαχαρίου**
Πατρώνυμο	**Σπυρίδων**
Αριθμός Μητρώου	**ΜΠΣΠ/ 11003**
Επιβλέπων	**Μιχαήλ Ψαράκης, Επίκουρος Καθηγητής**

Ημερομηνία Παράδοσης: **Ιούλιος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Μιχαήλ Ψαράκης Επίκουρος Καθηγητής	Άγγελος Πικράκης Επίκουρος Καθηγητής	Παναγιώτης Κοτζανικολάου Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία έχει ως σκοπό να παρουσιάσει την σύζευξη δύο τεχνολογικών πεδίων. Τα γενικά πεδία αυτά είναι της Μηχανικής ή υπολογιστικής Όρασης και της Επιτάχυνσης Υλικού και πιο συγκεκριμένα της επιτάχυνσης μέσω Ετερογενούς Παράλληλης Επεξεργασίας με χρήση Επεξεργαστών Γραφικών για Προγραμματισμό Γενικού Σκοπού (General Purpose Graphics Processing Unit programming - GPGPU). Ειδικότερα, θα μελετηθεί το πρόβλημα της Μηχανικής Όρασης, το οποίο είναι ένα πολυσύνθετο αντικείμενο, σε πληθώρα των συνιστωσών του, με σκοπό να γίνει χρήση των τεχνολογιών επιτάχυνσης ως αρωγή για την υλοποίηση της λύσης του. Η εργασία χωρίζεται σε τρία λογικά τμήματα.

- Στο πρώτο γίνεται η παρουσίαση των επιμέρους τεχνολογικών πεδίων και επιστημονικών αρχών που διέπουν την Μηχανική Όραση, των παραδοσιακών τεχνικών σε επίπεδο θεωρητικό αλλά και εφαρμοσμένο που μέχρι σήμερα έχουν καθιερωθεί στον χώρο, καθώς και ορισμένων πρόσφατων, καινοτόμων αλλά αναγνωρισμένων προσεγγίσεων.
- Στο δεύτερο εκθέτονται οι τεχνολογίες που δύνανται να χρησιμοποιηθούν για να επιταχύνουν τις μεθόδους επίλυσης των ζητούμενων της Μηχανικής Όρασης.
- Στο τρίτο και τελευταίο τμήμα, παρουσιάζεται η υλοποίηση εφαρμογής που συνδυάζει τα δύο αυτά τεχνολογικά πεδία. Η εφαρμογή αποτελεί υλοποίηση του αλγορίθμου SIFT (Scale Invariant Feature Transform) με χρήση της γλώσσας CUDA της NVIDIA. Επίσης αναλύονται οι μέθοδοι, οι προκλήσεις και οι προτεινόμενες και υλοποιούμενες τεχνικές που χρησιμοποιήθηκαν. Τέλος, παρουσιάζονται οι πειραματικές μετρήσεις και τα συνεπαγόμενα αποτελέσματα των δοκιμών της υλοποίησης αυτής, τα οποία επιβεβαιώνουν την επίτευξη του στόχου της επιτάχυνσης και το όφελος της χρήσης της τεχνολογίας του Γενικού Σκοπού Προγραμματισμού Επεξεργαστών Γραφικών, σε εφαρμογές μηχανικής όρασης.

ABSTRACT

This thesis aims to present the conjunction of two technological fields. Those fields are Computer Vision and Hardware Acceleration, more precisely acceleration by means of Heterogeneous Parallel Processing using General Purpose Graphics Processing Unit programming (GPGPU). The problem of Computer Vision, which is a complex subject, will be studied in many of its constituents, with the goal of using acceleration technologies to assist the implementation of its solution. The thesis text is divided in three logical partitions.

- The firs one presents the respective technological fields and scientific principles that govern Computer Vision, the traditional techniques in a theoretical and applied level along with certain recent, innovative but nonetheless acknowledged approaches.
- The second part exhibits the technologies that can be used to speed up the solution methods of the Computer Vision demands.
- The third and last part presents the implementation of an application that combines the two fields. This application implements the SIFT algorithm (Scale Invariant Feature Transform) using NVIDIA's CUDA programming language, following an analysis of the methods, the challenges, as well as the suggested and implemented techniques. Finally, the experimental measurements and the entailing results of this implementation are presented, to confirm the attainment of the acceleration goal, as well as the benefit of the use of General Purpose GPU programming in Computer Vision applications.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά πρωτίστως τον κ. Ψαράκη Μιχάλη για την καθοδήγηση, την υποστήριξη και τις συμβουλές του καθ' όλη την διαδικασία υλοποίησης της εργασίας αυτής, αλλά και για τις υψηλές προδιαγραφές που έθεσε. Επίσης θα ήθελα να ευχαριστήσω όλο το ακαδημαϊκό προσωπικό του μεταπτυχιακού τμήματος για την συνεισφορά τους στην διεύρυνση του γνωστικού μου ορίζοντα και το υψηλό τους επιστημονικό επίπεδο που εμπνέει την επιδίωξη της αριστείας στους σπουδαστές τους.

Τέλος, θα ήθελα να αφιερώσω αυτή την εργασία στην μητέρα μου, αλλά και τους υπόλοιπους κοντινούς μου ανθρώπους, που με στήριξαν και ανέχτηκαν τις δυσκολίες που η προσπάθειά μου αυτή εναπόθεσε πάνω τους.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Πίνακας Εικόνων

ΕΙΣΑΓΩΓΗ

Στην εργασία αυτή εξετάζονται οι θεωρητικές αρχές δύο τεχνολογικών πεδίων, της Μηχανικής Όρασης και της Επιτάχυνσης Επεξεργασίας μέσω υλικού. Η Μηχανική Όραση αποτελεί την προσπάθεια να αναπτυχθούν αλγόριθμοι και εφαρμογές με την δυνατότητα να προσλαμβάνουν πληροφορία και δεδομένα από εικόνες και βίντεο με τρόπο που θα είναι δυνατόν ο υπολογιστής να επεξεργάζεται και να χειρίζεται το αντικείμενο των εικόνων σε επίπεδο λογικής παρόμοιο με του ανθρώπου. Αυτό επιτυγχάνεται με την ανάλυση της εικόνας κάνοντας χρήση εργαλείων της Επεξεργασίας Εικόνας και της Μηχανικής Μάθησης. Για να είναι δυνατή η χρήση των εξαγόμενων δεδομένων από την Μηχανική Μάθηση, ανάλογα με το πεδίο της εφαρμογής, πρέπει να γίνουν οι κατάλληλες μαθηματικές διεργασίες για την απαλοιφή περιττής πληροφορίας από την αρχική εικόνα και την τροποποίηση των δεδομένων σε απλοποιημένη μορφή ικανή να χρησιμοποιηθεί από συστήματα λήψης αποφάσεων.

Το κίνητρο για την επιλογή του πεδίου της Μηχανικής Όρασης ως αντικείμενο της εφαρμογής της εργασίας αυτής, είναι το γεγονός ότι η Μ.Ο. είναι ένα από τα πιο ραγδαία αναπτυσσόμενα στον τομέα της Πληροφορικής, και οι εφαρμογές του διεισδύουν καθημερινά σε ολοένα αυξανόμενο αριθμό των καθημερινών και επαγγελματικών δραστηριοτήτων της κοινωνίας. Το αντικείμενο αυτό έχει μεγάλο βαθμό μαθηματικής πολυπλοκότητας και πλήθος υπολογισμών, οι οποίοι ωστόσο, σε πολλές περιπτώσεις είναι επαναληπτικοί. Αυτό έχει ως αποτέλεσμα να περιορίζονται από την συμβατική αρχιτεκτονική υπολογιστών, κάτι που καθιστά την Μηχανική Όραση ιδανικό πεδίο εφαρμογής των τεχνολογιών Επιτάχυνσης Επεξεργασίας μέσω υλικού. Η επιτάχυνση στην συνέχεια παρέχει την δυνατότητα αξιοποίησης ακόμα πιο περίπλοκων και αποδοτικών αλγορίθμων καθώς καθίσταται δυνατή η αποδέσμευση τους από τους χρονικούς περιορισμούς που φέρουν. Υπάρχει πληθώρα δημοσιεύσεων και εργασιών που αποτελούν ένδειξη της τάσης να υιοθετούνται τεχνολογίες επιτάχυνσης για την υλοποίηση αλγορίθμων Μηχανικής Όρασης. Ενδεικτικά αναφέρονται μερικές κατηγορίες τέτοιων αλγορίθμων:

- Επεξεργασία Εικόνας [1][2].
- Ιστογράμματα HOG [3][4].
- Σημειακοί ανιχνευτές χαρακτηριστικών και οπτικής ροής [5][6][7][8][9].
- Ολοκληρωμένα πλαίσια εργασίας και βιβλιοθήκες λογισμικού για επεξεργασία εικόνας και Μηχανική Όραση [10][11][12].

Η περιπλοκότητα της παραγόμενης πληροφορίας από έναν αλγόριθμο Μηχανικής Όρασης, εξαρτάται από τον στόχο της εκάστοτε εφαρμογής. Για οπτική αναγνώριση γραφής, θα πρέπει για παράδειγμα ο αλγόριθμος να είναι σε θέση να φιλτράρει την εικόνα με τρόπο που θα είναι διακριτά σε επίπεδο δυαδικής εικόνας τα σχήματα των γραμμάτων, χωρίς να αλλοιωθούν ή να παραμορφωθούν, προκειμένου να δύναται ο ευφυής αλγόριθμος που θα τα επεξεργαστεί να τα αναγνωρίζει. Σε εφαρμογές αναγνώρισης και παρακολούθησης αντικειμένου, είναι απαραίτητο να προσδιοριστούν οι παράμετροι που καθορίζουν το επιθυμητό αντικείμενο και να μοντελοποιηθούν με τρόπο που θα είναι δυνατή η ταχεία αναζήτηση και ταυτοποίησή τους. Παλαιότερα, γινόταν χρήση μαθηματικών και γεωμετρικών προσδιορισμών για την αναγνώριση απλών αντικειμένων και σχημάτων, σε εικόνες έγχρωμες και μη. Πλέον οι πιο αποδοτικοί αλγόριθμοι κάνουν χρήση σημείων ενδιαφέροντος, προκειμένου να ταυτοποιήσουν ένα αντικείμενο, ακόμα και σε περιβάλλοντα υψηλού θορύβου, κίνησης και περιπλοκότητας. Σε αυτή την κατηγορία ανήκει ο αλγόριθμος που επιλέχτηκε ως αντικείμενο της εργασίας αυτής. Ο αλγόριθμος αυτός είναι ο Scale Invariant Feature Transform (SIFT) ή Μετασχηματισμός Χαρακτηριστικών Αμετάβλητων από Κλίμακα. Τα χαρακτηριστικά αυτά μπορούν να χρησιμοποιηθούν σε πληθώρα εφαρμογών σε συνδυασμό με την τακτοποίησή τους σε βάση μοντέλων για αξιοποίηση από αλγορίθμους Μηχανικής Μάθησης. Τέτοιες εφαρμογές είναι:

- Αναγνώριση απλών και σύνθετων αντικειμένων, ακόμα και πολλαπλών ανά εικόνα, ακόμα και σε περιπτώσεις επικάλυψης και ανεξάρτητα από συνθήκες φωτισμού, μεγέθυνσης και περιστροφής τους.
- Εντοπισμός αντικειμένων και χαρτογράφηση κινήσεων σε ρομποτικά, βιομηχανικά και στρατιωτικά συστήματα, αλλά και συστήματα ασφαλείας.
- Συρραφή εικόνων για παραγωγή πανοραμάτων, αεροφωτογραφιών ή χαρτών.
- Τρισδιάστατη μοντελοποίηση χώρου και παρακολούθηση κίνησης.
- Ιατρικές απεικονίσεις όπως η τρισδιάστατη τομογραφία.

Τα χαρακτηριστικά που παράγει από μία εικόνα ο SIFT είναι πολύ αξιόπιστα σε σχέση με πληθώρα αλγορίθμων που έχουν εμφανιστεί έχοντας ως έμπνευση την λειτουργία του. Ωστόσο, οι αλγόριθμοι αυτοί επιχειρούν να απλοποιήσουν ορισμένες από τις επιμέρους διαδικασίες του για να καταστούν υπολογιστικά ελαφρύτεροι και αναπόφευκτα παράγουν λιγότερο αξιόπιστα αποτελέσματα, ανταλλάσσοντας την σταθερότητα των χαρακτηριστικών με ταχύτητα εκτέλεσης και συνεπώς δεν χρήζουν παραλληλίας στον βαθμό του SIFT. Αυτός είναι και ο λόγος που επιλέχτηκε ο αλγόριθμος αυτός προκειμένου να αντιμετωπιστεί η πρόκληση της ταχύτητας επεξεργασίας χωρίς έκπτωση στην σταθερότητα και αξιοπιστία της λειτουργίας του. Ο στόχος αυτός επιτυγχάνεται με χρήση της τεχνολογίας CUDA της NVIDIA, η οποία ανήκει στο πεδίο του Γενικού Σκοπού Προγραμματισμού Επεξεργαστών Γραφικών (GPGPU), και έχει καθιερωθεί ως μία από τις επικρατέστερες τεχνολογίες Επιτάχυνσης Επεξεργασίας μέχρι σήμερα.

Ο αλγόριθμος SIFT παρουσιάζεται εκτενώς σε επόμενο κεφάλαιο αλλά η λειτουργία του συνοπτικά, είναι ως εξής:

- Μετατροπή της εικόνας σε μονοχρωματική (Grayscale).
- Διαδοχική σμίκρυνση της εικόνας σε οκτάβες μέσω Διγραμμικής Παρεμβολής (Bilinear Interpolation). Οι εικόνες που παράγονται έχουν πλευρές ίσης αναλογίας μεταξύ τους αλλά μισό μήκος από το προηγούμενο επίπεδο
- Δημιουργία Πυραμίδων Gauss με διαδοχική θόλωση κάθε οκτάβας χρησιμοποιώντας αυξανόμενο βαθμό θόλωσης σε κάθε επίπεδο. Παράγεται έτσι ένας αριθμός εικόνων ανά οκτάβα.
- Οι πυραμίδες θόλωσης μετατρέπονται σε πυραμίδες διαφορών DoG μεταξύ των συνεχόμενων επιπέδων θόλωσης ανά οκτάβα. Αυτό προσομοιώνει την λειτουργία ενός φίλτρου Laplacian of Gaussian και παράγει εικόνες που διαγράφονται οι ακμές και τα σημεία ενδιαφέροντος.
- Για κάθε επίπεδο διαφορών πραγματοποιείται η σύγκριση μεταξύ του κάθε pixel με τα άμεσα γειτονικά του αλλά και τις αντίστοιχες ομάδες pixel στο ανώτερο και κατώτερο επίπεδο DoG. Τα σημεία που επικρατήσουν ως μέγιστα η ελάχιστα κατά τις συγκρίσεις αυτές σημαίνονται ως ακρότατα (extrema).
- Τα ακρότατα στην συνέχεια ελέγχονται ώστε να μην ανήκουν σε ακμές, καθώς αυτές αποτελούν ασταθή χαρακτηριστικά όταν μεταβληθεί το μέγεθος του αντικειμένου. Τα εναπομείναντα σημεία αποτελούν τα φιλτραρισμένα σημεία ενδιαφέροντος ή keypoints, που εξάγει ο αλγόριθμος ως πληροφορία.
- Προεραιτικά, και για να επιτευχθεί η αμεταβλητότητα των σημείων αυτών στην περιστροφή, εξάγεται η πληροφορία της κατεύθυνσης της διαβάθμισής τους μέσω χρήσης ιστογραμμάτων προσανατολισμένων διαβαθμίσεων (HOG).

Η επιλογή της χρήσης CUDA για την επιτάχυνση του αλγορίθμου έγινε με βάση τα κριτήρια υψηλής παραλληλίας που ικανοποιεί η τεχνολογία αυτή, καθώς ο αλγόριθμος αυτός παρουσιάζει τμήματα κώδικα εξαιρετικά παραλληληλοποιήσιμα με αρθρωτή δομή και συνεπώς ήταν εφικτή η διαμέρισή του σε τμήματα χαμηλής μαθηματικής πολυπλοκότητας και υψηλής επαναληψιμότητας. Έτσι η μαζικά παράλληλη αρχιτεκτονική της GPU είναι ιδανική για την υλοποίησή του.

Ο βασικός στόχος της υλοποιημένης εφαρμογής είναι να πετύχει την μέγιστη επιτάχυνση με δεδομένες τις δυνατότητες του υλικού που χρησιμοποιήθηκε. Αυτό πραγματοποιείται με την αξιοποίηση του βαθύτερου επιπέδου παραλληλίας του αλγορίθμου, δηλαδή την επαναλαμβανόμενη επεξεργασία σημείου προς σημείο όλων των pixel της εκάστοτε εικόνας.

Τέλος παρουσιάζονται οι μετρήσεις και τα πειραματικά συμπεράσματα της υλοποίησης του αλγορίθμου SIFT σε παράλληλη μορφή με χρήση CUDA και επιβεβαιώνεται η επίτευξη του στόχου της επιτάχυνσης τόσο τμηματικά όσο και καθολικά, με παράγοντες επιτάχυνσης που σε ορισμένες περιπτώσεις ξεπερνούν τις 4200 και καθολική επιτάχυνση τάξεως μεγέθους εν συγκρίσει με την ακολουθιακή εκδοχή του αλγορίθμου σε γλώσσα C.

1. ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ (Computer Vision)

Ένα από τα πιο ραγδαία αναπτυσσόμενα πεδία της Πληροφορικής είναι αυτό της «Μηχανικής Όρασης» , ή αλλιώς γνωστή ως Υπολογιστική Όραση (Computer Vision-CV). Το αντικείμενο που πραγματεύεται είναι οι τρόποι λήψης, επεξεργασίας, ανάλυσης και ερμηνείας εικόνων στατικών ή βίντεο καθώς και η εξαγωγή πολυδιάστατων δεδομένων από τον πραγματικό κόσμο δια μέσου αισθητήρων εικόνας, με σκοπό την αριθμητική ή και συμβολική παράσταση πληροφορίας που στην συνέχεια αξιοποιούνται από συστήματα λήψης απλών ή και σύνθετων αποφάσεων. Είναι αλλιώς ο τρόπος με τον οποίον ένα υπολογιστικό σύστημα μπορεί να «αντιληφθεί» το αντικείμενο μίας εικόνας κατά τρόπο προσεγγιστικά όμοιο με αυτόν που το πραγματοποιεί ο άνθρωπος. Σε ορισμένες μάλιστα περιπτώσεις η αντίληψη αυτή μπορεί να είναι ανώτερη από του ανθρώπου χάρις τις προσαυξημένες δυνατότητες του συστήματος να συνδυάσει την πληροφορία προερχόμενη από την εικόνα, με πληροφορία προερχόμενη από άλλους τύπους αισθητηρίων, αλλά φυσικά και λόγω του πλεονεκτήματος που έχει η μηχανή εν συγκρίσει με τον άνθρωπο ως προς την ταχύτητα επεξεργασίας και αριθμητικών υπολογισμών.

Δεδομένης της ευκολίας που παρέχει η Μηχανική Όραση σε ένα σύστημα για την εισαγωγή πληροφορίας, γίνεται αυτονόητα πολύ ελκυστικό εργαλείο για γειτνιάζοντα επιστημονικά και τεχνολογικά πεδία όπως η Τεχνητή Νοημοσύνη, η Μηχανική Μάθηση και η Βιο-εμπνευσμένη πληροφορική [13][14]. Σε πολλές περιπτώσεις μάλιστα η ενσωμάτωσή της μέσα σε αυτά τα πεδία είναι τόσο οσμωτική που είναι δύσκολο να διαχωρίσει κανείς τις επιμέρους διαδικασίες. Οι εφαρμογές της όμως όπως θα δούμε, εκτείνονται σε πολυάριθμους κλάδους των επιστημών και της τεχνολογίας, καθώς επίσης η ανάπτυξη και εξέλιξη της εξαρτάται και καθορίζεται αμφίδρομα από την αρωγή πολλών από τα πεδία αυτά τα οποία την αξιοποιούν. Το γεγονός αυτό προοιωνίζει και εξηγεί την μέχρι σήμερα ταχύτατη εξέλιξή της καθώς και το αναπόφευκτο της μελλοντικής της θεαματικής εξάπλωσης, αλλά βεβαίως και την ανάγκη για την ορθολογιστική της χρήση, δεδομένων των κοινωνικών επιπτώσεων που αναμφίβολα έχει.

Ενδεικτικά αναφέρονται ορισμένα από τα πεδία εφαρμογών της Μηχανικής Όρασης καθώς και παραδείγματα συγκεκριμένων χρήσεών τους.

Στον τομέα της Ιατρικής Πληροφορικής χρησιμοποιείται για την παραγωγή ειδικών απεικονίσεων από δεδομένα που είναι δυσνόητα ή αναξιοποίητα στην πρωτογενή τους μορφή από τους ιατρούς και τους ερευνητές, όπως για παράδειγμα η τρισδιάστατη απεικόνιση σε περιπτώσεις τομογραφικών εξετάσεων (computed tomography) ή η απαλοιφή εικονικού θορύβου σε άλλες απεικονίσεις.

Στον τομέα της ρομποτικής [15], αλλά και της στρατιωτικής βιομηχανίας είναι επίσης προφανής η διευκόλυνση που παρέχει η Μηχανική Όραση σε περιπτώσεις όπου απαιτείται αυτόματη αναγνώριση και ανάκτηση στόχων στον χώρο ακόμα και σε περιπτώσεις ταχύτατης κίνησης (computer guided munitions), αλλά ακόμη και σε εφαρμογές κατασκοπικής και αναγνωριστικής φύσεως.

Στον τομέα της αστρονομίας [2] όπως και της μετεωρολογίας η δυνατότητα επεξεργασίας τεράστιου όγκου πληροφορίας που εξάγεται από εικόνες ενισχύεται σημαντικά από τα εργαλεία της Μηχανικής Όρασης.

Πολλά από τα εργαλεία και τις μεθόδους της Μηχανικής Όρασης είναι επίσης απαραίτητα για την δισδιάστατη αλλά και τρισδιάστατη χαρτογράφηση χώρου τόσο γεωλογικών, γεωγραφικών όσο και αρχαιολογικών εφαρμογών, κάτι που επιτυγχάνεται με μεθόδους συρραφής εικόνων και τρισδιάστατη απεικόνιση [16].

Η φωτογραφία και η κινηματογράφηση επίσης αξιοποιούν εφαρμογές της Μηχανικής Όρασης [17] σε περιπτώσεις όπως η σταθεροποίηση Βίντεο και η απαλοιφή φαινομένων ανάκλασης και κακού φωτισμού. Πολλές σύγχρονες κάμερες μάλιστα περιλαμβάνουν τέτοια συστήματα για την αυτόματη ρύθμιση των παραμέτρων λήψης τους (auto calibration, auto filtering, DSP).

Η Μηχανική Όραση χρησιμοποιείται σε πολλαπλά στάδια της αλυσίδας παραγωγής βιομηχανικών μονάδων τόσο για την αυτοματοποίηση διαδικασιών σε συνδυασμό με εξελιγμένα αλλά και απλά ρομποτικά συστήματα, όπως επίσης και στον ποιοτικό έλεγχο [18].

Ένας ακόμη τομέας που αξιοποιεί τους καρπούς της Μηχανικής Όρασης είναι αυτός της φυσικής ασφάλειας χώρων, σε εξελιγμένα συστήματα κλειστών κυκλωμάτων ασφαλείας και συναγερμού [19]. Κατ' επέκταση οι αστυνομικές αρχές και άλλες κρατικές υπηρεσίες μπορούν να επωφελούνται από την χρήση όμοιων συστημάτων [5], κάτι το οποίο πέρα από τα πολλαπλά και αυτονόητα οφέλη (ρύθμιση κυκλοφορίας, πυρασφάλεια, νόμιμη αστυνόμευση, έγκαιρη παρέμβαση) περιπλέκει τις επιπτώσεις της χρήσης αυτής της τεχνολογίας στο κοινωνικό και πολιτικό περιβάλλον.

Τέλος οι εφαρμογές στον οικιακό αυτοματισμό [20], την ψυχαγωγία και την καθημερινή εργασία είναι πολυπληθείς. Τέτοιες είναι, συστήματα όπως το kinect της Microsoft για την αναγνώριση κινήσεων και την χρήση τους ως δεδομένα εισόδου σε συστήματα αλληλεπίδρασης με τον άνθρωπο, εφαρμογές οπτικής αναγνώρισης χαρακτήρων (OCR) [21] για την αξιοποίηση της χειροκίνητης γραφής και την αυτόματη ψηφιοποίηση εντύπων εγγράφων προς επεξεργασία, μέχρι και όχι-πλέον-φουτουριστικά συστήματα τρισδιάστατου βίντεο, εικονικής πραγματικότητας και προσαυξημένης πραγματικότητας (Google glass).

1.1 Επίπεδα της Μηχανικής Όρασης

Η γενική διαδικασία μιας εφαρμογής Μηχανικής Όρασης ανάλογα βέβαια και με το δεδομένο πρόβλημα που καλείται να λύσει, μπορεί να διαχωριστεί σε διακριτά βήματα ή επίπεδα. Σε κάποιες πιο σύνθετες περιπτώσεις η διάκριση τους είναι δύσκολη, όπως επίσης και η ακολουθιακή τους ιεραρχία. Σε άλλες πιο απλές η εξελικτικά απλοποιημένες κάποια από αυτά παραλείπονται η συγχωνεύονται. Η περιπλοκότητα της σύνθεσης αυτών των επιμέρους επιπέδων υπαγορεύεται από τη σκοπιά της εφαρμογής και του προβλήματος που καλείται να λύσει. Για παράδειγμα, αν το ελεγχόμενο περιβάλλον στο οποίο θα κληθεί να λειτουργήσει το σύστημα, προσφέρει μικρά επίπεδα περιττής πληροφορίας και θορύβου, η παράλειψη αλγορίθμων που εξυπηρετούν την απαλοιφή τους ενδείκνυται προς όφελος της ταχύτητας και της αξιόπιστης επαναληψιμότητας, ή απλά για την μείωση του κόστους και των πιθανοτήτων σφάλματος λόγω περιπλοκότητας. Αν το περιβάλλον του συστήματος περιέχει θόρυβο ή απαιτεί περίπλοκες ευφυείς αποφάσεις, αναπόφευκτα απαιτείται η σύνθεση περισσότερων αλγορίθμων από τα διαφορετικά επίπεδα της Μηχανικής όρασης. Ακολουθεί αναφορά τους και συνοπτική επεξήγησή των βασικών αρχών που τα διέπουν.

- Επεξεργασία εικόνας (image processing)
- Ανίχνευση αντικειμένου ή στόχου (object detection) και όραση σε ενδιάμεσο στάδιο
- Μηχανική μάθηση – υπολογιστική νοημοσύνη (machine learning)
  - Αναγνώριση ταυτοποιημένου αντικειμένου ή στόχου (object recognition) (υπόθεση)
  - Κατηγοριοποίηση στόχου ή αντικειμένου (object classification) και όραση σε υψηλό επίπεδο
- Παρακολούθηση κινούμενου αντικειμένου / στόχου (object tracking) ή κινούμενου παρασκηνίου (φόντου) (optical flow)

1.2 Επεξεργασία εικόνας

Αυτό το πεδίο [22] αφορά τις διαδικασίες που εφαρμόζονται σε μία στατική ψηφιακή εικόνα ακόμα και αν αυτή αποτελεί μέρος μιας συνεχόμενης ροής εικόνων (βίντεο). Οι διαδικασίες αυτές συνήθως, μοντελοποιούνται μαθηματικά μέσω πινάκων και αριθμητικών ακολουθιών και με την χρήση φίλτρων που συχνά εξηγούνται με διαφορικό / ολοκληρωτικό λογισμό και απειροστικό διανυσματικό λογισμό, μαθηματική μορφολογία και στατιστική. Αναφορικά σε αυτή την κατηγορία ανήκουν οι ακόλουθες θεωρητικές αρχές και οι αντίστοιχοι αλγόριθμοι.

1.2.1 Φίλτρα

Τα γραμμικά και μη φίλτρα, αποτελούν το πιο σύνηθες βήμα που προηγείται του οποιουδήποτε αλγορίθμου μηχανικής όρασης και μάλιστα πολύ συχνά παρεμβάλλονται σε διάφορα σημεία της αλυσίδας συνδεδεμένων αλγορίθμων (pipelined approach) για να μειώσουν την δημιουργία θορύβου, να ρυθμίσουν φωτομετρικές παραμέτρους και να βελτιώσουν την συνολική αξιοπιστία της εφαρμογής. Επειδή κυρίως αποτελούν σύνθετες μαθηματικές διεργασίες που εφαρμόζονται σε όλα τα pixel της εικόνας, αυξάνουν τον υπολογιστικό φόρτο και καθιστούν την χρήση τους σε εφαρμογές πραγματικού χρόνου λιτή. Σε αυτόν τον τομέα όπως θα δούμε αργότερα, είναι και η βασικότερη συνεισφορά της παράλληλης επεξεργασίας και της επιτάχυνσης επεξεργασίας δεδομένων ειδικότερα. Η βάση των γραμμικών φίλτρων είναι κυρίως η ίδια όπως και στην ψηφιακή επεξεργασία σήματος (μονοδιάστατου), δηλαδή η Συνέλιξη και η ανάλυση Fourier.

1.2.2 Επεξεργασία Σημείου προς Σημείο

Στην επεξεργασία μίας εικόνας με αυτόν τον τρόπο, ο μετασχηματισμός αλλάζει μόνο ένα pixel σε ένα κανάλι [1]. Έστω $f(x,y)$ η τιμή του pixel σε συντεταγμένες x και y στην εικόνα, $g(x,y)$ ένας μετασχηματισμός (φίλτρο) της $f(x,y)$ και T μία συνάρτηση του $f(x,y)$. Τότε η T χαρτογραφεί την $f(x,y)$ στην $g(x,y)$ και ο μετασχηματισμός επηρεάζει μόνο το pixel στην θέση (x,y) ως ακολούθως:

$$g(x,y) = T[f(x,y)]$$

Παραδείγματα τέτοιας επεξεργασίας είναι οι αρνητικές εικόνες, η μετατροπή σε μονοχρωματική εικόνα, οι μετατροπές φωτεινότητας, δηλαδή εκλάμπρυνση (brightening) και συσκότιση εικόνας, καθώς και τα φίλτρα κατωφλίωσης.

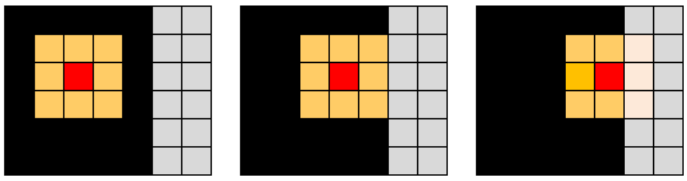
1.2.3 Συνέλιξη 2 διαστάσεων /Χωρική Συνέλιξη (Spatial Convolution)

Όπως π.χ. στην συνέλιξη ενός διακριτού σήματος εφαρμόζουμε την μέθοδο του ολισθαίνοντος παραθύρου που αντιστοιχεί στο φίλτρο, έτσι και εδώ έχουμε ένα παράθυρο 2 διαστάσεων το οποίο λέγεται πυρήνας (kernel) και πρέπει να πραγματοποιήσει σάρωση όλης της εικόνας. Η διακριτή συνέλιξη 2 διαστάσεων (χωρική συνέλιξη) για εικόνα $f[x,y]$ με πυρήνα $g[x,y]$ ορίζεται ως εξής:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g\,[x-n_1, y-n_2]$$

Σε αυτό το στάδιο, επειδή χρησιμοποιείται πυρήνας, που ονομάζεται και κορμός, γύρω από το εξεταζόμενο pixel πρέπει να ληφθούν υπόψη το μέγεθος του πυρήνα και να γίνουν οι σχετικές παραδοχές για την συμπεριφορά του φίλτρου στα άκρα της εικόνας.

πυρήνας (παράθυρο συνέλιξης) 3x3



1-1-i Πρόβλημα Συνέλιξης στα άκρα εικόνας.

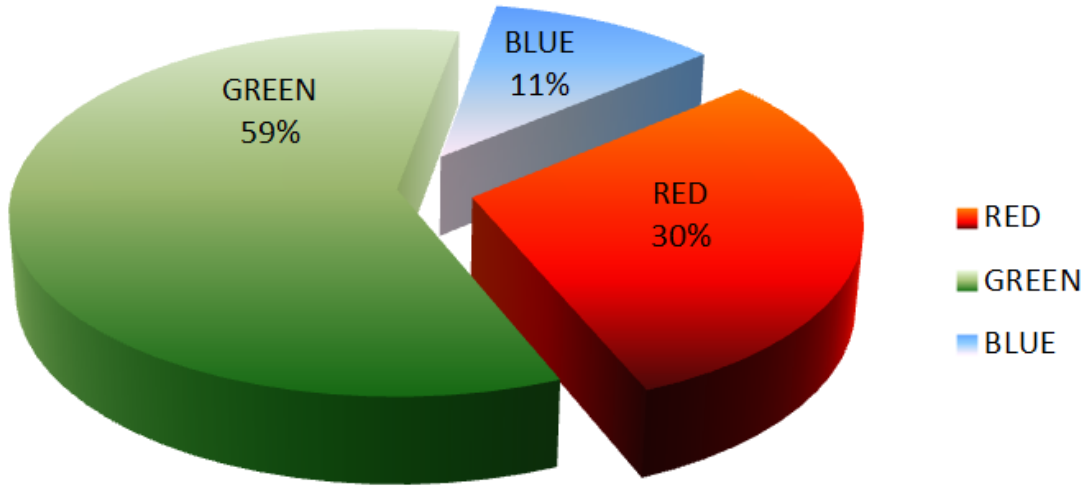
1.2.4 Φίλτρο Μονοχρωματικής Μετατροπής (Grayscale filter)

Το φίλτρο αυτό είναι μία συνάρτηση που αποφέρει την φωτεινή ένταση έναντι του χρώματος. Συχνά αναφέρεται λανθασμένα ως ασπρόμαυρο φίλτρο. Ένας τρόπος για να πραγματοποιηθεί αυτό είναι ο μέσος όρος των τριών χρωματικών καναλιών RGB.

$$GrayI(x,y) = \frac{Red(x,y) + Green(x,y) + Blue(x,y)}{3}$$

Σύμφωνα με ορισμένους ερευνητές η εξίσωση αυτή πρέπει να τροποποιείται με συγκεκριμένους δείκτες προκειμένου να ανταποκρίνεται καλύτερα στον βαθμό ευαισθησίας του ανθρώπινου ματιού στις διαφορετικές χρωματικές συχνότητες. Το μέγεθος αυτό ονομάζεται Φωταύγεια (Luminescence) και η αναλογία μεταξύ των χρωμάτων συνήθως ρυθμίζεται από μία παράμετρο που ονομάζεται Gamma.

$$Gray(x,y) = 0.3Red(x,y) + 0.59Green(x,y) + 0.11Blue(x,y)$$



1-ii Αναλογία Φωταύγειας RGB

1.2.5 Αρνητική Εικόνα

Αυτό το φίλτρο αντιστρέφει τα επίπεδα έντασης της εικόνας. Είναι χρήσιμο στις μονοχρωματικές εικόνες όταν χρειάζεται να βρεθούν συγκεκριμένα χαρακτηριστικά ενδιαφέροντος και το επικρατές χρώμα είναι το λευκό ή το μαύρο όπως για παράδειγμα στις ακτινογραφίες. Έστω D το χρωματικό βάθος της εικόνας σε bit, τότε η ένταση του χρώματος της εικόνας βρίσκεται εντός του διαστήματος $[0, 2^D - 1]$, π.χ. για 8 bit έχουμε εύρος τιμών από 0 έως 255. Και έτσι το φίλτρο περιγράφεται ως εξής.

$$Neg(x,y) = 2^D - 1 - f(x,y)$$

1.2.6 Φίλτρο Εκλάμπρυνσης (Brightening Filter)

Σε αυτή την περίπτωση αναβαθμίζονται οι τιμές των pixel σε μεγαλύτερες μέσω συνάρτησης που εφαρμόζεται πάνω στην $f(x,y)$ ή με σταθερές $a > 1, b > 0$ όπως στην ακόλουθη εξίσωση.

$$g(x,y) = af(x,y) + b$$

Ένας τρόπος να γίνει αυτός ο μετασχηματισμός είναι με χρήση ημιτονικής συνάρτησης, όπου k είναι η γωνιακή συχνότητα, μ είναι το μέγιστο πλάτος και $I(x,y)$ είναι η αρχική εικόνα.

$$Bright(x,y) = \mu \sin\left(k\, I(x,y)\right), \mu \in [0, \frac{\pi}{2}]$$

Η προηγούμενη εξίσωση ομαλοποιείται λόγω του εύρους τιμών βάσει χρωματικού βάθους D.

$$Bright(x,y) = \Lambda \sin\left(\frac{\pi\, I(x,y)}{2\Lambda}\right), \forall\, \mu = \Lambda \in [0, 2^D - 1], k = \frac{\pi}{2\Lambda}$$

1.2.7 Φίλτρο Συσκότισης

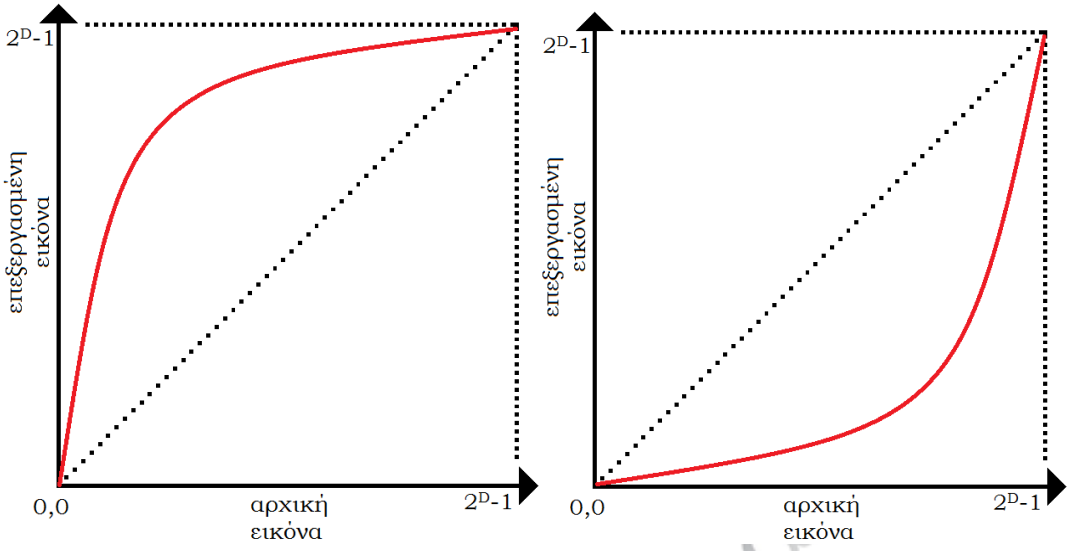
Σε περιπτώσεις όπου η εικόνα εμφανίζεται πολύ φωτεινή λόγω έκθεσης στο φώς, μπορεί να επεξεργαστεί για να γίνει πιο σκούρα. Γι αυτόν τον σκοπό μπορεί να χρησιμοποιηθεί μία συνημιτονική συνάρτηση στο ίδιο εύρος του φίλτρου εκλάμπρυνσης. Η λειτουργία του μειώνει τις τιμές έντασης και ως αποτέλεσμα σκουραίνει το χρώμα της αρχικής εικόνας και εκφράζεται με την ακόλουθη μορφή.

$$Dark(x,y) = \mu\,(1 - \cos\left(kI(x,y)\right))$$

Η ομαλοποίηση όμοια με την ημιτονική συνάρτηση για την ψηφιακή εικόνα γίνεται όπως ακολούθως.

$$Dark(x,y) = \Lambda(1 - \cos\left(\frac{\pi\, I(x,y)}{2\Lambda}\right)), \forall\, \Lambda \in [0, 2^D - 1]$$

Γραφικά τα δύο φίλτρα φωτεινότητας (εκλάμπρυνσης και συσκότισης) παρίστανται ως εξής.



1-iii Φίλτρα Φωτεινότητας (Brightness Filters)

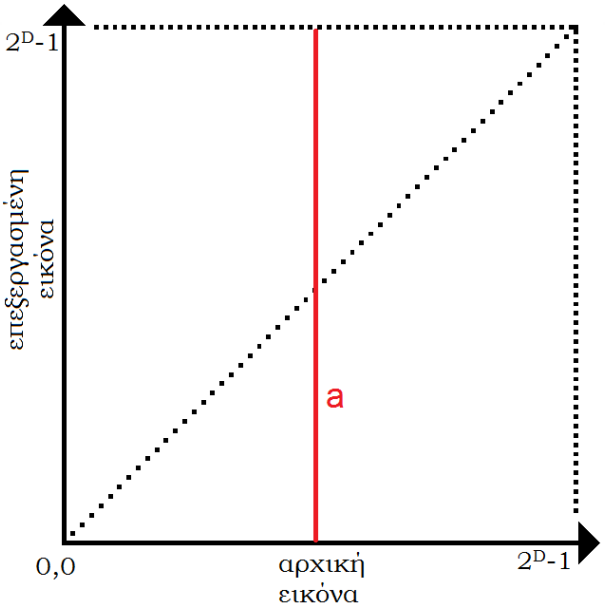
1.2.8 Φίλτρα Αντίθεσης

Ορισμένες φορές η εφαρμογή φίλτρων φωτεινότητας δεν αρκεί για να τονίσει τις λεπτομέρειες μίας εικόνας. Κατά συνέπεια θα πρέπει να πραγματοποιήσουμε επιπλέον χειρισμούς όπως:

- Αύξηση της έντασης των φωτεινότερων τόνων και μείωση των σκοτεινότερων.
- Αύξηση της έντασης των σκοτεινότερων τόνων και μείωση των φωτεινότερων.

Αυτός ο τύπος επεξεργασίας ονομάζεται Αντίθεση (contrast). Η Αντίθεση επιτυγχάνεται με χρήση ενός κατωφλίου έντασης I_T πάνω από το οποίο οι φωτεινοί τόνοι αυξάνονται και κάτω από αυτό μειώνονται, ή αντίστροφα. Στην περίπτωση που οι τόνοι πάνω από το κατώφλι μετασχηματίζονται στην μέγιστη δυνατή τιμή και οι τόνοι κάτω από αυτό στην ελάχιστη, έχουμε ένα φίλτρο Υψηλής Αντίθεσης, το αποτέλεσμα του οποίου είναι μία εικόνα δύο χρωματικών τόνων, γνωστή και ως δυαδική εικόνα. Αυτή η περίπτωση εκφράζεται ως ακολούθως:

$$Bin(x,y) = \begin{cases} 0 & ,\forall\, I(x,y) \le I_T \\ 2^D - 1 & ,\forall\, I(x,y) > I_T \end{cases}$$

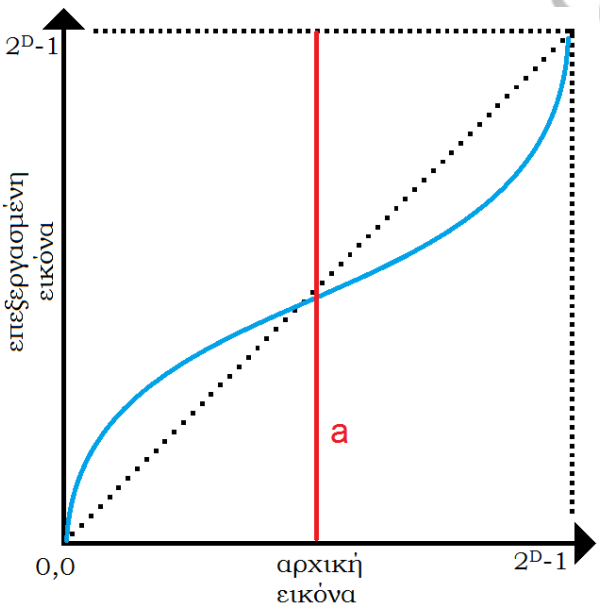


1-iv Φίλτρο υψηλής αντίθεσης-δυαδικό (High contrast filter -binary)

Σε πάρα πολλές περιπτώσεις, η αντίθεση θα βελτιώσει τον τόνο χωρίς την ακρότητα της δυαδικής εικόνας. Συνήθεις τεχνικές αντίθεσης είναι οι εξής:

- Αντίστροφη ημιτονοειδής συνάρτηση.

$$InvSin(x,y) = I(x,y) - \lambda \sin\left(\frac{2\pi\, I(x,y)}{\Lambda}\right)$$



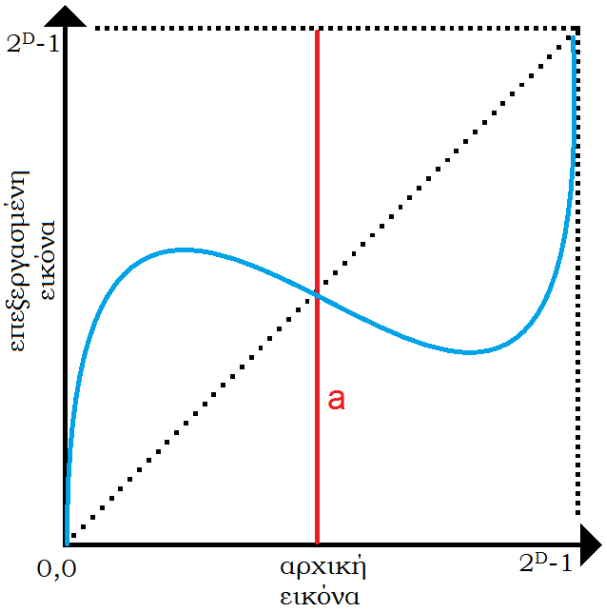
1-v Φίλτρο αντίθεσης: εφαπτομένης (Contrast filter: tangent)

- Υπερβολική συνάρτηση εφαπτομένης, για την αύξηση της αντίθεσης

$$HyTan(x,y) = \frac{\Lambda}{2}\left[1 - \tan h\left[a(I(x,y) - \frac{a\Lambda}{2})\right]\right], \forall\, a > 0$$

- Ημινονική συνάρτηση, για την μείωση της αντίθεσης.

$$sin\,(x,y) = I(x,y) + \lambda \sin\left(\frac{2\pi\, I(x,y)}{\Lambda}\right), \forall\, \lambda\ \in [0{,}40)$$



1-vi Φίλτρο Αντίθεσης: ημιτονικό (Contrast filter: sine)

1.2.9 Χρωματική Αντιπαράθεση (dithering)

Στις περιπτώσεις που είθισται να απαλείψουμε το σφάλμα κβαντισμού / δειγματοληψίας για να αποφύγουμε την δημιουργία ανεπιθύμητων υφών, χρησιμοποιούμε σκόπιμα την «επιμόλυνση» της εικόνας με τεχνητό θόρυβο. Η τεχνική αυτή ονομάζεται Χρωματική Αντιπαράθεση (dithering). Κάποιες φορές μάλιστα αυτό έχει αποτέλεσμα στην εμφάνιση της εικόνας ως οφθαλμαπάτη μεγαλύτερης ανάλυσης ή ψηφιακού βάθους και συνεπώς χρησιμοποιείται ευρέως στις τεχνολογίες συμπίεσης και απεικόνισης εικόνας και βίντεο.

1.2.10 Θόλωση Gauss (Gaussian Blur)

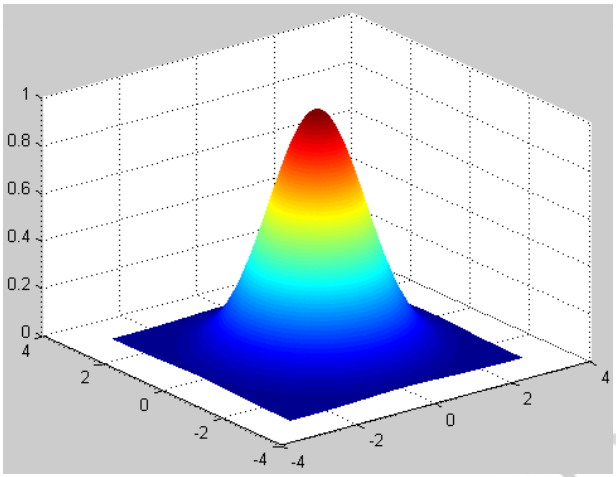
Το πιο κοινά χρησιμοποιούμενο φίλτρο στην υπολογιστική όραση πιθανώς να είναι αυτό της Θόλωσης ή εξομάλυνσης Gauss (Gaussian Blur) και μαθηματικά αντιστοιχεί με συνέλιξη της εικόνας με μία συνάρτηση Gauss [23], ή αλλιώς μετασχηματισμός Weierstrass.

Δεδομένου του ότι ο μετασχηματισμός Fourier μίας συνάρτησης Gauss, παράγει εκ νέου συνάρτηση Gauss, η Θόλωση ουσιαστικά είναι ένα χαμηλοπερατό φίλτρο [24]. Μία ακόμη χρήση της συνάρτησης Gauss είναι σε αλγόριθμους συναρτήσεων πυραμιδικής σμίκρυνσης (και μεγέθυνσης) εικόνας [25], όπου με παρόμοια τεχνική εξάγεται ο μέσος όρος γειτονικών pixel (Median filter) για την ομαδοποίησή τους σε νέα pixel της γεωμετρικά-αναλογικά μικρότερης εικόνας (Gaussian-Laplacian pyramid). Συνοπτικά η Θόλωση Gauss υλοποιείται ως εξής:

- Υπολογίζουμε τα βάρη του μονοδιάστατου παραθύρου G'_n
- Φιλτράρουμε τους άξονες X και Y μονοδιάστατα με μέγεθος παραθύρου $2N+1$ για την ταυτόχρονη σάρωση των 2 αξόνων και σ συντελεστή μείωσης διακύμανσης.
-

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

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



1-vii 2D Gaussian Distribution

1.2.11 Τεχνικές Βελτίωσης (Image Restoration)

Σε πολλές περιπτώσεις η εικόνα την οποία καλείται να αναλύσει η υπολογιστική όραση είναι αλλοιωμένη από θόρυβο ή κακές συνθήκες φωτισμού, ανακλαστικότητας και λήψης και συνεπώς η διαδικασία της εξαγωγής πληροφορίας δυσχεραίνεται. Η λύση δίνεται με την επεξεργασία εικόνας κάνοντας χρήση συγκεκριμένων τεχνικών βελτίωσης [26]. Η αξιολόγηση των τεχνικών αυτών γίνεται σε ελεγχόμενο πειραματικό περιβάλλον όπου η προσθήκη των αλλοιώσεων γίνεται τεχνητά και συνεπώς έχουμε πρόσβαση στην ιδανική εικόνα, ή συγκρίνοντας τα αποτελέσματά τους στην πράξη. Για την αξιολόγηση αυτή χρησιμοποιούμε συνήθως τα ακόλουθα κριτήρια:

- Λόγος σήματος – θορύβου (Signal to Noise Ratio- SNR) το οποίο υπολογίζεται σε decibel (dB) ως εξής:

$$SNR = 10\log_{10}(\frac{\overline{signal\ power}}{\overline{noise\ power}})$$

- Λόγος κορυφής σήματος – θορύβου (Signal to Noise Ratio- SNR) το οποίο είναι και το πιο αντικειμενικό μέτρο εκτίμησης της ποιότητας της βελτιωμένης, ανακατασκευασμένης εικόνας. Υπολογίζεται επίσης σε decibel (dB) ως εξής:

$$PSNR = 10\log_{10}\frac{\psi_{max}}{\sigma_e^2}$$

Όπου ψ_{max} είναι η μέγιστη ένταση pixel της εικόνας και σ_e^2 είναι το Σφάλμα Μέσου Τετραγώνου (Mean Square Error – MSE) μεταξύ δύο εικόνων f_1και f_2 διαστάσεων *M*N* pixels και εκφράζεται ως εξής:

$$\sigma_\varepsilon^2 = MSE = \frac{1}{MN}\sum_i\sum_j (f_1(i,j) - f_2(i,j))^2$$

Δηλαδή συνδυαστικά:

$$PSNR = 10\log_{10}\frac{\psi_{max}}{\frac{1}{MN}\sum_i\sum_j (f_1(i,j) - f_2(i,j))^2}$$

Μία από τις παλαιότερες τεχνικές βελτίωσης και σημαντικά διαδεδομένη είναι αυτή του N. Wiener. Το φίλτρο Wiener υλοποιεί μία εκτίμηση της αποκατεστημένης εικόνας που ελαχιστοποιεί την εξής συνάρτηση στατιστικού σφάλματος:

$$e^2 = E[(f_{original} - f_{restored})^2]$$

Όπου E: γνωστός τελεστής μέσης τιμής (expected value operator) και είναι ισοδύναμο του MSE και η επίλυση της συνάρτησης αυτής στο πεδίο της χωρικής συχνότητας εκφράζεται ως εξής:

$$F_{restored}(u,v) = [\frac{1}{H(u,v)}\cdot\frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}}]\cdot G(u,v)$$

Ο λόγος $\frac{S_n(u,v)}{S_f(u,v)}$ είναι ο λόγος του θορύβου προς την ισχύ του σήματος και είναι αντίστροφο του SNR.

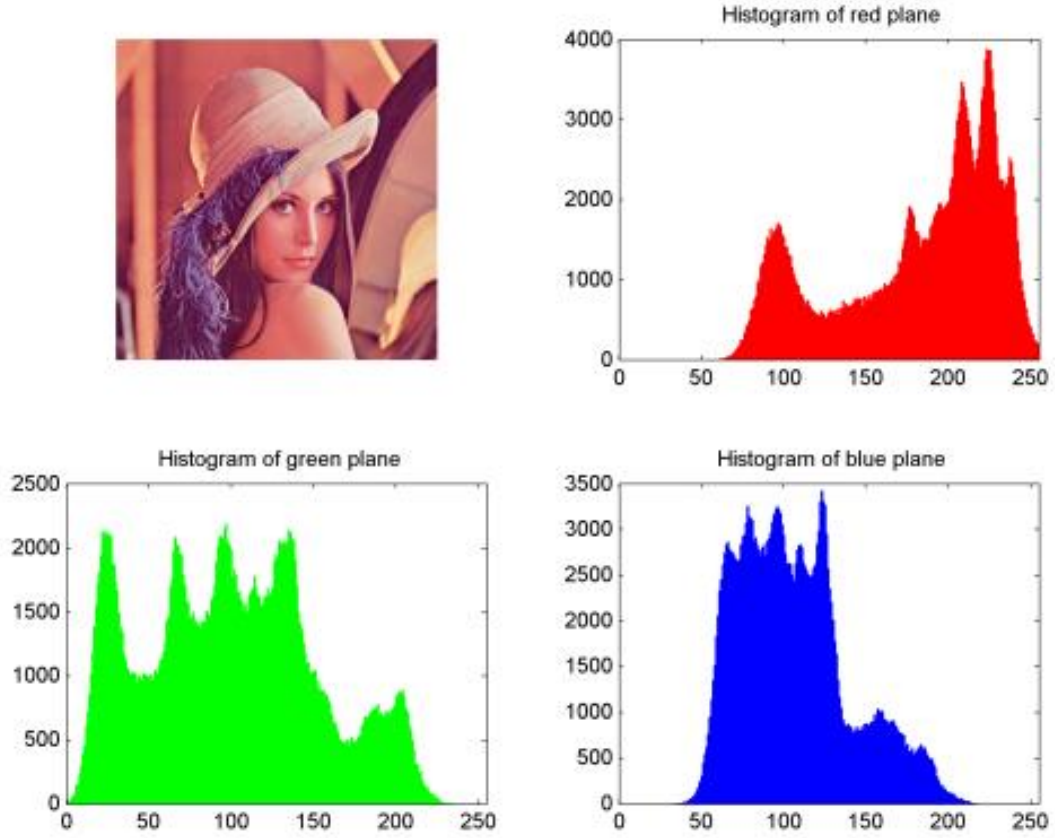
Μία ακόμη τεχνική αποκατάστασης είναι το φίλτρο περιορισμένων ελαχίστων τετραγώνων (constrained least-squares - CLS) $H_{cls}(u,v)$ που εκφράζεται πιο εύκολα στο πεδίο του διακριτού Fourier:

$$H_{cls}(u,v) = \frac{D^*(u,v)}{D^*(u,v)D(u,v) + aC^*(u,v)C(u,v)}$$

Όπου a είναι μία παράμετρος που ικανοποιεί το MSE ή υπολογίζεται από άλλες αναλυτικές μεθόδους.

1.2.12 Ιστογράμματα (Histograms)

Τα ιστογράμματα είναι ένα εργαλείο που χρησιμοποιείται παραδοσιακά στην ανάλυση εικόνας για να εξαχθούν συμπεράσματα και στατιστικά που αφορούν την φωτεινότητα, την αντίθεση και τους χρωματικούς τόνους των διαφορετικών καναλιών σε μη μονόχρωμες εικόνες. Στην επεξεργασία εικόνας, συχνά πραγματοποιείται εξισορρόπηση ιστογράμματος (Histogram equalization), προκειμένου να καταστεί εκμεταλλεύσιμο όλο το ψηφιακό βάθος της εικόνας, να βελτιωθεί η αντίθεση (contrast) όταν είναι επιθυμητό και να διευκολυνθούν οι περεταίρω διαδικασίες φιλτραρίσματος και κατωφλίωσης . Στο επίπεδο μάλιστα της αναγνώρισης αντικειμένου, ενίοτε γίνεται χρήση των ιστογραμμάτων για την σύγκριση και αντιστοίχιση / ταύτιση του ανιχνευόμενου αντικειμένου με το δείγμα της υπόθεσης. Ένα καλό παράδειγμα της χρήσης του ιστογράμματος στην υπολογιστική όραση, είναι ο εντοπισμός κινούμενου αντικειμένου σε στατικό φόντο πραγματοποιώντας σάρωση με μερικά ιστογράμματα στις επιμέρους περιοχές της εικόνας, προσδιορίζοντας έτσι την περιοχή στην οποία υπήρξε διαφοροποίηση στην κατανομή του ιστογράμματος. Επίσης, μη συχνοτικά ιστογράμματα χρησιμοποιούνται για την κατηγοριοποίηση δεδομένων κατά την ανάλυση μίας εικόνας, όπως στην περίπτωση των Ιστογραμμάτων Προσανατολισμού Διαβάθμισης (Histogram of Oriented Gradients -HOG) ενός σημείου ενδιαφέροντος.



1-viii Ιστογράμματα των καναλιών RGB εικόνας "Lena.jpg".

Συνοπτικά η μέθοδος της εξίσωσης ιστογράμματος [27] περιγράφεται ως εξής:

- Αρχικά έχουμε: $r = f(x, y)$
- Θεωρούμε r τυχαία μεταβλητή με Συνάρτηση Πυκνότητας Πιθανότητας $p_r(r)$
- Μετασχηματίζουμε σε $s = T(r)$
- Ιδιότητες του T:
  Αν $r_1 < r_2$ τότε και $T(r_1) < T(r_2)$
  Αν $0 < r < 1$ τότε και $0 < T(r) < 1$
- Η Συνάρτηση Πυκνότητας Πιθανότητας $p_s(s)$ δίνεται από

$$p_s(s) = \left[\frac{p_r(r)}{\left|\frac{ds}{dr}\right|}\right]_{r=T^{-1}(s)}$$

- Ο μετασχηματισμός αθροιστικής κατανομής

$$s = T(r) = \int_0^r p_r(\tau)d\tau\,, 0 < r < 1$$

οδηγεί σε ομοιόμορφη Συνάρτηση Πυκνότητας Πιθανότητας ps(s) με τα ζητούμενα χαρακτηριστικά [28].

- Επειδή συνήθως το r είναι διακριτό χρησιμοποιείται ο εξής μετασχηματισμός

$$s_k = T(r_k) = \sum_{i=0}^{k} \frac{n_i}{n}, 0 < r_k < 1$$

1.2.13 Μετατροπές χρωματικών χώρων (color space conversion)

Οι μετατροπές μεταξύ χρωματικών χώρων (color spaces) είναι σημαντικό εργαλείο για την προετοιμασία μίας εικόνας, προκειμένου να διαφοροποιηθούν χαρακτηριστικά που είναι απαιτούμενα από την εκάστοτε εφαρμογή. Το ανθρώπινο μάτι είναι σε θέση να δεχθεί έγχρωμη εικόνα, η οποία κατά σύμβαση στον υπολογιστή κωδικοποιείται με συγκεκριμένους τρόπους χρωματικής οργάνωσης που ονομάζονται χρωματικά μοντέλα. Τέτοια μοντέλα είναι το RGB (Red Green Blue),CMYK (Cyan Magenta Yellow Key) και δημιουργούν ουσιαστικά πίνακες που στα αντίστοιχα «κανάλια» τους περιέχουν την ένταση του συγκεκριμένου συνιστούντος χρώματος για την σύνθεση της εικόνας. Σε ορισμένες περιπτώσεις ωστόσο, είναι πιο εύκολη η εξαγωγή πληροφορίας από εικόνες μονοχρωματικές (greyscale), καθότι προσφέρονται για πιο άμεση εξαγωγή φωτεινότητας και ιστογραμμάτων. Οι ίδιες πληροφορίες είναι δυνατόν να εξαχθούν από έγχρωμη εικόνα, αλλά με αυξημένο υπολογιστικό φόρτο. Σε άλλες περιπτώσεις είθισται να χρησιμοποιούμε παραλλαγμένες μορφές απεικόνισης έγχρωμης RGB εικόνας που ευνοούν την ανίχνευση διακριτών σχημάτων [29] και την διαφοροποίηση αντικειμένων στον ψευδο-τρισδιάστατο χώρο. Τέτοιες παραστάσεις είναι οι δύο πιο διαδεδομένες παραστάσεις «κυλινδρικών συντεταγμένων» HSV (Hue Saturation Value) HSL (Hue Saturation Lightness). Θεωρητικά η μετατροπή ενός ιδανικού χρωματικού μοντέλου σε ένα άλλο, θα έπρεπε να παράγει απόλυτα όμοιες εικόνες, όμως στο κβαντισμένο περιβάλλον του υπολογιστή, οι περιορισμοί χρωματικού / ψηφιακού βάθους δειγματοληψίας (bit depth) δημιουργούν αλλοιώσεις μεταξύ τους, κάτι το οποίο εκμεταλλευόμαστε συχνά για να αξιοποιήσουμε την ικανότερη για το πρόβλημά μας παράσταση. Ορισμένες φορές μάλιστα το αποτέλεσμα είναι καλύτερο, αν εσκεμμένα και με στοχευμένη αυθαιρεσία παραστήσουμε τα δεδομένα του ενός χρωματικού χώρου σε αναντιστοιχία με τον άλλον. Αυτό γίνεται ακόμα πιο επιτακτικό, όταν πραγματοποιούμε πολυφασματική ανάλυση της εικόνας (multispectral image analysis) ή φασματική περιγραφή της.

Προσεγγιστικά μία μετατροπή από RGB σε HSV γίνεται ως εξής:

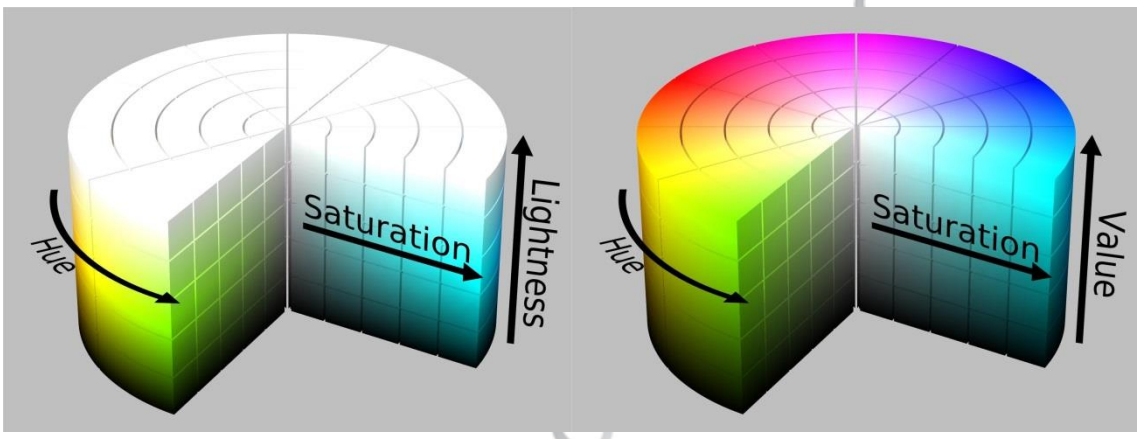
$$V = max(R, G, B)$$

$$S = \frac{255\ (V - \min(R, G, B))}{V}, \forall\ V \neq 0$$

$$S = 0, \forall\ V = 0$$

$$H = \begin{cases} \frac{60(G - B)}{S}, \forall\ V = R \\ 180 + \frac{60(G - B)}{S}, \forall\ V = G \\ 240 + \frac{60(R - G)}{S}, \forall\ V = B \end{cases}$$

$$H = H + 360, \forall\ H < 0$$



1-ix Παράσταση χρωματικών χώρων HSV – HSL (https://mhstekkomp.files.wordpress.com/2011/05/hsl-hsv-cilinder-diagram.jpg)

1.3 Ανίχνευση Αντικειμένου

1.3.1 Κατάτμηση Εικόνας - Δυαδική Εικόνα (Segmentation - binary image)

Στις πιο απλές εφαρμογές υπολογιστικής όρασης, συνήθως η εξαγωγή της πληροφορίας επιτυγχάνεται όταν μετά από την επεξεργασία της εικόνας με κάποια γραμμικά φίλτρα, μετατροπές χρωματικού χώρου, ή και κάποιες άλλες πιο πρωτόγονες μεθόδους επεξεργασίας, όπως για παράδειγμα φίλτρο υψηλής αντίθεσης, καταλήγουμε στην διαδικασία μίας λογικής κατωφλίωσης των pixel καθολικά ή τοπικά (global / local binarization) και εξάγουμε μία «δυαδική εικόνα» που απεικονίζει τις περιοχές ενδιαφέροντος σύμφωνα με τα χαρακτηριστικά «κατωφλίου» (threshold). Μετά τον εντοπισμό των περιοχών, γίνεται η κατάτμηση (segmentation), δηλαδή ο διαχωρισμός τους από την υπόλοιπη εικόνα. Η κατάτμηση είναι ένα σημαντικό στάδιο της Ανάλυσης Εικόνας, ώστε να ξεχωρίσουν οι περιοχές με χρήσιμη πληροφορία. Σε υψηλότερα επίπεδα της Μηχανικής Όρασης μάλιστα, η κατάτμηση χρησιμοποιείται για την αναγνώριση αντικειμένου και την παρακολούθησή του σε ροές εικόνων, πάντα σαν δομικό στοιχείο περισσότερο σύνθετων αλγορίθμων. Συνήθως η δυαδική εικόνα, είτε παριστά το απομονωμένο μοναδικό αντικείμενο ενδιαφέροντος της εφαρμογής μας, ή μας δίνει ένα σύνολο από συμπαγή σχήματα (blob detection) τα οποία μέσω μαθηματικής μορφολογίας μπορούμε να ερμηνεύσουμε και να αξιοποιήσουμε ως πολλαπλά ή και διαφορετικά αντικείμενα.

Η εξαγωγή μίας δυαδικής εικόνας μπορεί να γίνει πολύ απλά, είτε θέτοντας ένα κατώφλι σε μία παράμετρο της εικόνας όπως π.χ. η φωτεινότητα των pixel, αλλά και πιο σύνθετα κριτήρια που συχνά έχουν να κάνουν με τον συνυπολογισμό γειτονικών pixel και εφαρμόζοντας προϋποθέσεις υστέρησης ή πραγματοποιώντας πιο σύνθετες μαθηματικές διαδικασίες.

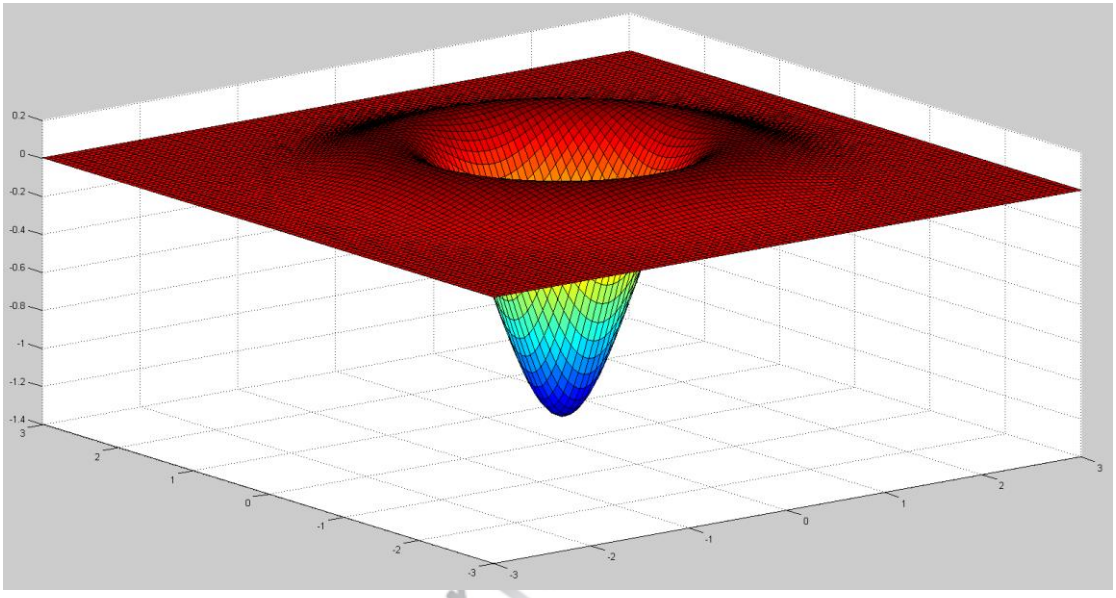
1.3.2 Ανίχνευση Ακμών (Edge Detection)

Ένα ζητούμενο της κατάτμησης είναι συχνά η ανίχνευση ακμών. Σε εικόνες δυαδικού ή μονόχρωμου χρωματικού χώρου αυτό υλοποιείται απλά με την μέθοδο των τελεστών 1ης παραγώγου Sobel και τελεστών κλίσης (gradient operators) **.** Σε πιο σύνθετες καταστάσεις γίνεται χρήση τελεστών Laplace 2ης παραγώγου που ορίζονται ως εξής:

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

Οι παράγωγοι 2ου βαθμού χρησιμοποιούνται επίσης όταν γίνεται αναζήτηση κλειστού «περιγράμματος» (contours) για τον προσδιορισμό μορφολογικών χαρακτηριστικών.

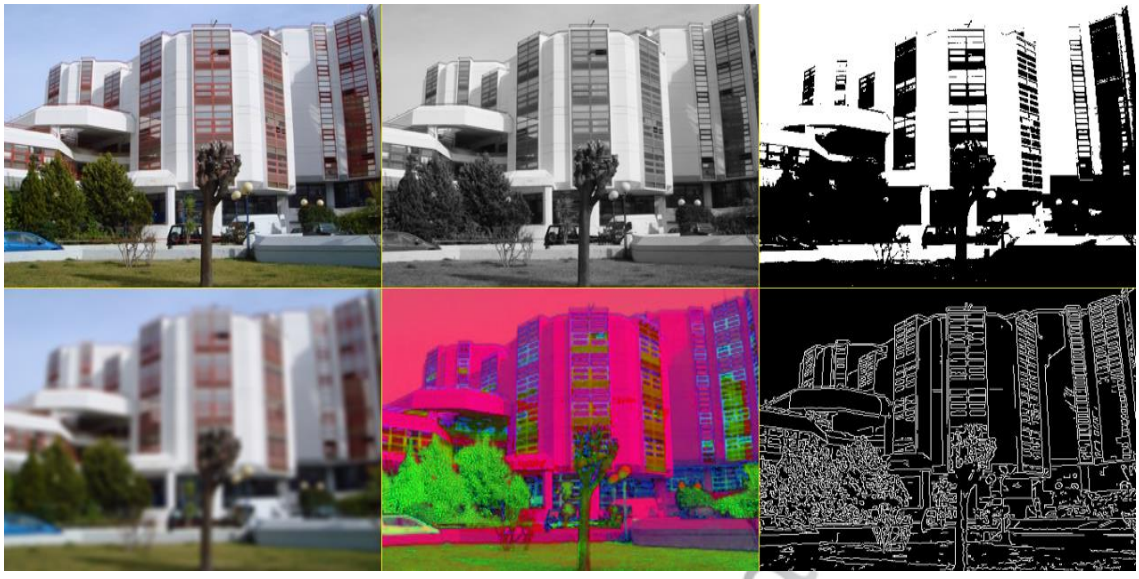
Σε περιπτώσεις που απαιτείται ακόμα μεγαλύτερη αξιοπιστία έναντι υπολογιστικού κόστους, ενδείκνυται η χρήση του ευφυούς αλγορίθμου ανίχνευσης ακμών Canny Edge Detector [30]**,** έναν από τους πιο διαδεδομένους αλγορίθμους της κατηγορίας του.



1-x Κατανομή Laplacian of Gaussian

Ενδεικτικά, η μέθοδος Canny Edge Detector λειτουργεί ως εξής:

- Εξομαλύνεται την εικόνα για απαλοιφή θορύβου κατά προτίμηση με φίλτρο Θόλωσης Gauss (Gaussian Blur) ή άλλο αντίστοιχο.
- Υπολογίζεται οι μερικές παράγωγοι της έντασης για τους άξονες x και y. Να σημειωθεί επίσης ότι σε εικόνες πολλαπλών καναλιών, η ένταση υπολογίζεται προσεγγιστικά από το άθροισμα συγκεκριμένων αναλογιών (στο RGB π.χ. έχουμε Luminance =0.2126*R + 0.7152*G + 0.0722*B) Εδώ χρησιμοποιούνται οι τελεστές Laplace και εφόσον έχει ακολουθήσει η θόλωση Gauss ο αλγόριθμος καλείται Laplacian of Gaussian (LoG filter) [31].
  $I_x = G_\sigma^x * I$ και $I_y = G_\sigma^y * I$
- Υπολογίζεται το μέτρο του τελεστή κλίσης (gradient) για κάθε pixel.
  $M(x,y) = |\nabla I| = \sqrt{I_x^2 + I_y^2}$
  Σε αυτό το στάδιο μπορεί να προσδιοριστεί ο τελεστής προσανατολισμού του gradient μέσω τριγωνομετρικού υπολογισμού του τόξου εφαπτομένης του παραπάνω διανύσματος έντασης (compass operator).
- Απαλείφονται τα pixel εκείνα που δεν είναι τα τοπικά μέγιστα του μέτρου στην κατεύθυνση του διανύσματος του τελεστή κλίσης.
- Πραγματοποιούμε κατωφλίωση με «υστέρηση».
  - Διαλέγονται τα pixel για τα οποία $M > T_h$(υψηλό κατώφλι).
  - Συλλέγονται τα pixel για τα οποία $M > T_l$((χαμηλό κατώφλι) και τα οποία είναι γειτνιάζοντα ήδη συλλεγμένων σημείων ακμής.



1-xi Αποτελέσματα φίλτρων:
Πάνω: Αρχική εικόνα (RGB), μονοχρωματική εικόνα (greyscale), δυαδική εικόνα (binary)
Κάτω: Θόλωση Gauss, χρωματικός χώρος HSV, Ανίχνευση ακμών Canny

1.3.3 Μαθηματική Μορφολογία (Morphological Operations)

Το μαθηματικό πεδίο της Μορφολογίας έχει συνεισφέρει στην επεξεργασία εικόνας μεθόδους, όπως η «διάβρωση» (erosion) και «διαστολή» (dilation), το γεωμετρικό άνοιγμα και κλείσιμο, αλλά και η σχηματική απεικόνιση με πολυγωνικούς ή γραμμικούς σκελετούς και συνδεόμενες ελλείψεις.

Η βασική προσέγγιση της Μορφολογίας είναι η σάρωση μίας εικόνας, συνήθως δυαδικής, με ένα απλό, προκαθορισμένο σχήμα, εξάγοντας συμπεράσματα για την ομοιότητα ή διαφορετικότητα των σχημάτων στην εικόνα. Το προκαθορισμένο αυτό σχήμα ονομάζεται δομικό στοιχείο και αποτελεί και αυτό συνήθως δυαδική εικόνα. Έστω E ένας ευκλείδειος συνεχής γεωμετρικός χώρος, ή ένα πλέγμα ακεραίων και A μία δυαδική εικόνα εντός του E.

Η διαδικασία της μορφολογικής διάβρωσης (erosion) έχει το αποτέλεσμα της λέπτυνσης των γραμμών και την ανάδειξη των πιο λεπτών χαρακτηριστικών της εικόνας, κάτι που την καθιστά ευαίσθητη στον θόρυβο και πραγματοποιείται ως εξής:

$$A \ominus \mathrm{B} = \{\mathrm{z} \in \mathrm{E} | B_z \subseteq \mathrm{A}\},$$

όπου B_zείναι η ερμηνεία του B από το διάνυσμα z που σημαίνει

$$B = \{b + z | b \in B\}, \forall z \in E$$

Άλλη έκφραση της διάβρωσης του A από το B είναι η εξής:

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

Η διαδικασία της μορφολογικής διαστολής (dilation) έχει το αντίθετο αποτέλεσμα από την διάβρωση. Η διαστολή της εικόνας A από δομικό στοιχείο B πραγματοποιείται ως εξής:

$$A \oplus B = \bigcup_{b \in B} A_b$$

Επίσης, για $B^S = \{x \in E | -x \in B\}$ *όπου* B^s *συμμετρικό του* B , έχουμε:

$$A \oplus B\{z \in E | (B^s)_z \cap A \neq \emptyset\}$$

Η διαδικασία του ανοίγματος της εικόνας A από το δομικό στοιχείο B πραγματοποιείται όταν αρχικά γίνει διαστολή της A από το B και στην συνέχεια διάβρωση της παραγόμενης εικόνας από το B. Το κλείσιμο είναι η ακριβώς αντίστροφη ακολουθιακή εκτέλεση από το άνοιγμα.

Πιο εξελιγμένες παραλλαγές των αλγορίθμων της μαθηματικής μορφολογίας αναλαμβάνουν ανάλογες διαδικασίες για εικόνες μονοχρωματικές ή και έγχρωμες. Για τις μονοχρωματικές εικόνες οι οποίες είναι συναρτήσεις χαρτογράφησης ενός ευκλείδειου χώρου ή πλέγματος εντός του συνόλου των πραγματικών αριθμών ($\mathbb{R}$), τα αντίστοιχα δομικά στοιχεία ονομάζονται δομικές συναρτήσεις. Έτσι λοιπόν για εικόνα $f(x)$ και δομική συνάρτηση $b(x)$ έχουμε τους ακόλουθους τύπους:

Μονοχρωματική διαστολή:

$$(f \oplus b)(x) = \underset{y \in E}{supremum} [f(y) + b(x-y)]$$

Μονοχρωματική διάβρωση:

$$(f \ominus b)(x) = \underset{y \in E}{infimum} [f(y) - b(y-x)]$$

1.3.4 Μετασχηματισμός Hough (Hough Transform)

Μία ακόμη κατηγορία αλγορίθμων επεξεργασίας εικόνας που χρησιμοποιούνται στην Μηχανική Όραση είναι οι «Μετασχηματισμοί Hough». Στόχο έχουν συνήθως την ανίχνευση γραμμών και κύκλων άλλα σε πιο γενικευμένη μορφή μπορούν να ανιχνεύσουν διάφορα μη προσδιορισμένα σχήματα.

Γενικά ο μετασχηματισμός αυτός μπορεί να μας δώσει την μετατροπή κάθε μη μηδενικού pixel στην εικόνα σε μία ημιτονοειδή συνάρτηση στο πεδίο Hough, καθώς και την οπισθοπροβολή Hough κάθε σημείου που στο πεδίο Hough μετατρέπεται σε ευθεία γραμμή στην εικόνα (Hough Backprojection).

Για μία συνάρτηση $A(x,y)$, ο μετασχηματισμός Hough ορίζεται ως:

$$H(\theta,\rho) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} A(x,y)\delta(\rho - x\cos\theta - y\,sin\,\theta)dx\,dy$$

Όπου δ είναι η συνάρτηση delta – Dirac. Με $A(x,y)$, κάθε σημείο (x,y) στην αρχική εικόνα A μετασχηματίζεται σε μία ημιτονοειδή συνάρτηση $\rho = xcos\theta - ysin\theta$ όπου ρ είναι η κάθετη απόσταση μεταξύ προέλευσης της γραμμής σε μία γωνία $\theta \in 0 \le \theta \le \pi$, κάτι που μπορεί να επιστρέψει και αρνητικές τιμές.

Σημεία που βρίσκονται πάνω στην ίδια ευθεία στην εικόνα , θα παράγουν ημιτονοειδείς συναρτήσεις που τέμνονται ένα κοινό σημείο στον μετασχηματισμό Hough. Για τον αντίστροφο μετασχηματισμό ή οπισθοπροβολή, κάθε σημείο στο πεδίο Hough μετατρέπεται σε ευθεία γραμμή στην εικόνα.

Συνήθως, η συνάρτηση Hough χρησιμοποιείται με δυαδικές εικόνες. Στην περίπτωση αυτή ή $H(\theta,\rho)$ δίνει τον συνολικό αριθμό ημιτονοειδών που διασχίζουν το σημείο (θ,ρ), και συνεπώς τον συνολικό αριθμό σημείων που αποτελούν την ευθεία γραμμή στην αρχική εικόνα. Διαλέγοντας ένα κατώφλι T για την $H(\theta,\rho)$, και χρησιμοποιώντας τον αντίστροφο μετασχηματισμό Hough δύναται να φιλτραριστεί η εικόνα έτσι ώστε να διατηρήσει μόνο γραμμές που περιέχουν τουλάχιστον T σημεία.

Ο διακριτός τύπος για τον μετασχηματισμό Hough μίας εικόνας A_{mn} διαστάσεων M και N, με δείκτες συντεταγμένων $m \in [0, M-1]$ και $n \in [0, N-1]$ είναι ο ακόλουθος:

$$H(\theta,\rho) = \sum_m \sum_n A_{mn}\,\delta(\rho,[\rho'])$$

Όπου [ρ'] είναι η στρογγυλοποίηση στον πλησιέστερο ακέραιο, και:

$$\rho' = (m\Delta x + x_{min})\cos\theta + (n\Delta y + y_{min})\sin\theta$$

Τα pixel θεωρούνται ότι έχουν διαστήματα Δx και Δy στις x και y κατευθύνσεις. Η delta συνάρτηση ορίζεται ως εξής:

$$\delta(\rho,[\rho']) = \begin{cases} 1 & \forall\ \rho = [\rho'] \\ 0 & \forall\ \rho \neq [\rho'] \end{cases}$$

Ο αντίστροφος μετασχηματισμός B_{mn} περιέχει όλες τις ευθείες γραμμές που δίνονται από τα σημεία (θ,ρ) που παράγονται από την $H(\theta,\rho)$. Η διακριτή μαθηματική έκφραση είναι:

$$B_{mn} = \begin{cases} \sum_{\theta}\sum_{\rho} H(\theta,\rho)\delta(n,[am+b]) & \forall\ |sin\theta| > \frac{\sqrt{2}}{2} \\ \sum_{\theta}\sum_{\rho} H(\theta,\rho)\delta(m,[a'n+b']) & \forall\ |sin\theta| \leq \frac{\sqrt{2}}{2} \end{cases}$$

Επίσης οι κλίσεις και η μετατόπιση δίνεται από:

$$\alpha = -\frac{\Delta x}{\Delta y}\frac{cos\theta}{sin\theta}, \qquad b = \frac{\rho - x_{min}cos\theta - y_{min}\ sin\theta}{\Delta y\ sin\theta}$$

$$\alpha' = \frac{1}{\alpha}, \qquad b' = \frac{\rho - x_{min}cos\theta - y_{min}\ sin\theta}{\Delta\chi\ cos\theta}$$

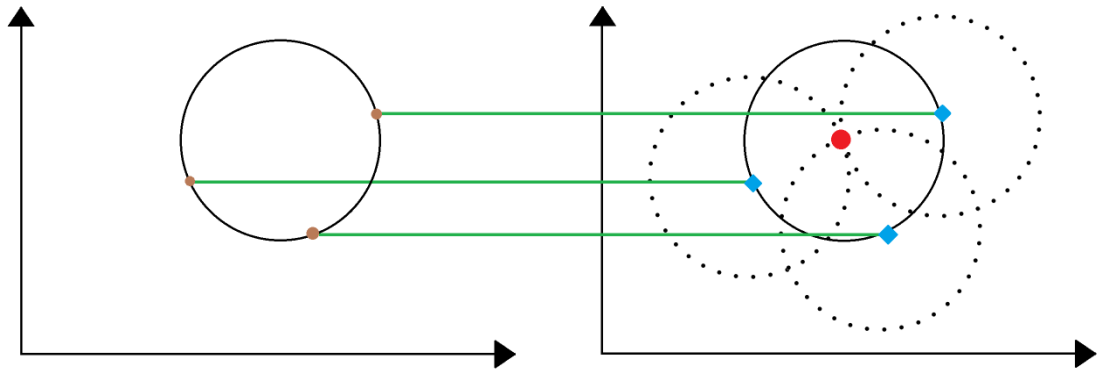
Ο μετασχηματισμός Hough για εύρεση κύκλων (Circle Hough Transform – CHT) [32] είναι μία επέκταση της φιλοσοφίας του μετασχηματισμού για εύρεση γραμμών. Χρησιμοποιείται για να προσδιορίσει τις παραμέτρους του κύκλου όταν είναι γνωστό ένα πλήθος σημείων που συμπίπτουν με την περίμετρο. Ένας κύκλος με ακτίνα R και κέντρο (a,b) μπορεί να περιγραφεί με τις παραμετρικές εξισώσεις:

$$x = a + R\ cos\ (\theta)$$
$$y = b + R\ sin\ (\theta)$$

Όταν η γωνία θ σαρώνει όλη την έκταση των 360^o, τα σημεία (x,y) διαγράφουν /ανιχνεύουν την περίμετρο ενός κύκλου. Αν μία εικόνα περιέχει πολλά σημεία, κάποια από τα οποία εμπίπτουν στην περίμετρο κύκλων, τότε το έργο του προγράμματος αναζήτησης είναι να βρει τριάδες (a,b,R) που περιγράφουν κάθε κύκλο. Το γεγονός ότι ο παραμετρικός χώρος είναι τρισδιάστατος, σημαίνει ότι άμεση υλοποίηση της τεχνικής Hough θα έχει μεγάλο κόστος μνήμης και χρόνου.

Αν οι αναζητούμενοι κύκλοι είναι δεδομένης ακτίνας, ή έστω εντός ενός σχετικά περιορισμένου φάσματος τιμών ακτίνας, τότε η αναζήτηση μπορεί να υποβαθμιστεί σε δισδιάστατη ή μικρή σειρά δισδιάστατων. Ο στόχος είναι να βρεθούν οι (a,b) συντεταγμένες των κέντρων.

Το σμήνος των σημείων (a,b) στον παραμετρικό χώρο εμπίπτουν σε έναν κύκλο ακτίνας R με κέντρο στο (x,y). Το πραγματικό κεντρικό σημείο θα είναι κοινό σε όλους τους παραμετρικούς κύκλους και μπορεί να βρεθεί με έναν πίνακα συσσώρευσης Hough.



1-xii Hough Circles

Η ίδια τεχνική μπορεί να αποκαλύψει πολλαπλούς κύκλους με την ίδια ακτίνα. Αν η ακτίνα δεν είναι γνωστή, τότε το σμήνος των σημείων στον τρισδιάστατο παραμετρικό χώρο θα εμπέσει στην επιφάνεια ενός κώνου [33]. Κάθε σημείο (x,y) στην περίμετρο ενός κύκλου θα παράγει μία κωνική επιφάνεια στον παραμετρικό χώρο. Η τριάδα (a,b,R) θα αντιστοιχεί στην σημειακή συσσώρευση όπου το μεγαλύτερο πλήθος επιφανειών κώνων τέμνονται.

Υφίσταται επίσης και Γενικευμένος Μετασχηματισμός Hough [34] για εύρεση αυθαίρετων σχημάτων, τρισδιάστατων σχημάτων (με ασύμφορα υψηλό υπολογιστικό κόστος) καθώς και σύνθετων σχημάτων του οποίου η έκφραση είναι:

$$R_{\Phi} = T_S \left\{ T_\theta \left[\bigcup_{k=1}^{N} R_{S_k}(\Phi) \right] \right\} \forall\, \sigma\chi\acute{\eta}\mu\alpha\, S \ni [S_1, S_2, \dots S_N]$$

1.3.5 Ροπές (moments)

Ο υπολογισμός των ροπών μίας εικόνας [26] δίνει χαρακτηριστικά παρόμοια με την αντίστοιχη προσέγγιση στην Μηχανική. Η ροπή πρώτης τάξεως δίνει την ένταση του pixel κατ' αντιστοιχία με το κέντρο της μάζας στην μηχανική, η δεύτερης τάξεως δίνει τους δύο βασικούς άξονες διαγωνιοποίησης όπως στην μηχανική δίνονται οι άξονες αδράνειας. Συνεπώς οι ροπές βοηθούν στον προσδιορισμό του γεωμετρικού σχήματος που εικονίζεται και είναι ιδιαίτερα χρήσιμες σε περιπτώσεις φιλτραρισμένων και δυαδικών εικόνων.

Για μία δισδιάστατη συνάρτηση f(x,y) η ροπή τάξεως (p+q) ορίζεται ως:

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q\, f(x,y) dx\, dy \;\forall\, p.q \in \mathbb{N}$$

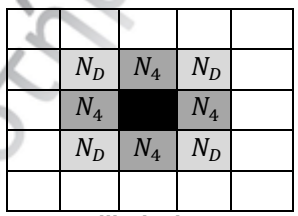
Για μία μονοχρωματική εικόνα με ένταση $I(x,y)$, οι ροπές υπολογίζονται από:

$$M_{ij} = \sum_x \sum_y x^i y^i I(x,y)$$

1.3.6 Εξαγωγή ορίων συνεκτικών περιοχών

Ένα pixel με συντεταγμένες (x,y) έχει 4 άμεσους γείτονες που απαρτίζουν το σύνολο:

$$N_4(p) = \{(x+1,y), (x-1,y), (x,y+1), (x,y-1)\}$$



1-xiii Pixel area

Και 4 διαγώνιους γείτονες που αποτελούν το σύνολο:

$$N_D(p) = \{(x+1,y+1), (x-1,y+1), (x+1,y-1), (x-1,y-1)\}$$

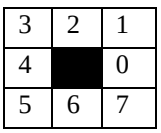
Κατά συνέπεια ένα pixel έχει συνολικά 8 γειτνιάζοντα :

$$N_8(p) = N_4(p) \cup N_D(p)$$

Συνήθως σε μία δυαδική εικόνα υφίστανται δύο μορφές συνεκτικότητας δύο pixel p και q:

- 4-συνεκτικότητα : $q \in N_4(p)$
- 8-συνεκτικότητα: $q \in N_8(p)$

Για την σήμανση των περιγραφόμενων pixel ακολουθείται η μέθοδος Freeman γνωστή και ως «κώδικας αλύσου» (chain coding) [35].



1-xiv Chain Code

Χρησιμοποιώντας τον κώδικα αυτό, σαρώνεται η εικόνα στους 2 άξονες της και χαρτογραφούνται τα pixel που έχουν γειτονική σχέση με σκοπό την εξαγωγή πληροφορίας περιγραμμάτων (contours), κάτι που είναι χρήσιμο για την αναγνώριση αντικειμένων.

Η «υπογραφή» ενός περιγράμματος [26] είναι ένας τρόπος παράστασης ενός δισδιάστατου αντικειμένου σε μία διάσταση, θεωρώντας ότι αυτή η παράσταση είναι ευκολότερη για την αξιοποίηση της από υπολογιστικές διαδικασίες. Έχουν αναπτυχθεί διάφορες μέθοδοι υπολογισμού υπογραφής. Η πιο κοινή είναι ο υπολογισμός της απόστασης από το προσεγγιστικό κέντρο του αντικειμένου προς το εκάστοτε σημείο του περιγράμματος, ως συνάρτηση της γωνίας του διανύσματος κατεύθυνσης προς το σημείο αυτό. Μία τέτοια υπογραφή, σε ένα σύστημα ορθογωνίων συντεταγμένων, εξαρτάται από την θέση, την κλίση (rotation) και το μέγεθος του αντικειμένου. Η θέση αποκαθίσταται εύκολα θέτοντας το προσεγγιστικό κέντρο του σχήματος στο κέντρο των συντεταγμένων αναφοράς (x-y offset) και το μέγεθος μπορεί επίσης σχετικά εύκολα να αποκατασταθεί διερευνώντας κάποιες προφανείς αναλογίες. Η δυσκολία βρίσκεται στην αποκατάσταση της περιστροφής του αντικειμένου. Αυτή η δυσκολία δύναται να αντιμετωπιστεί με μεταφορά του προβλήματος στο επίπεδο του μετασχηματισμού Fourier:

- Έστω $f(x)$ συνεχής συνάρτηση και $F(u)$ ο μετασχηματισμός της.

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi ux}dx$$

Και ο διακριτός μετασχηματισμός (DFT) της διακριτής μορφής της.

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{-j\frac{1}{N}2\pi ux}$$

- Χρησιμοποιώντας την ιδιότητα μετάφρασης του μετασχηματισμού Fourier που εκφράζεται στα ζεύγη, έχουμε:

$$f(x)e^{\frac{j2\pi u_0 x}{N}} \Leftrightarrow F(u-u_0)$$

Και

$$f(u)e^{\frac{j2\pi u_0 x}{N}} \Leftrightarrow F(x-x_0)$$

Αν εκφράσουμε την $F(u)$ στην εκθετική μορφή $F(u) = |F(u)|e^{j\Phi(u)}$, παρατηρούμε ότι η γωνία φάσης $\Phi(u)$ αλλάζει με την μετάφραση της $f(x)$, ενώ το φάσμα ενίσχυσης παραμένει σταθερό. Αυτό σημαίνει ότι η περιστροφή του αντικειμένου κατά γωνία φ, ισούται με μετάφραση της κυκλικής υπογραφής περιγράμματος του σχήματος με την ίδια γωνία. Συμπεραίνουμε λοιπόν, ότι το φάσμα ισχύος της μετασχηματισμένης κατά Fourier υπογραφής του περιγράμματος ενός αντικειμένου είναι ένας αμετάβλητος περιγραφέας, τηρουμένης της θέσεως, περιστροφής και αναλογίας μεγέθους του σχήματος του.

1.3.7 Υφή εικόνας

Στην επεξεργασία εικόνας, ένα χρήσιμο σύστημα ανάλυσης είναι αυτό της υφής (image texture). Γενικά αποτελεί ένα σύνολο μετρικών εργαλείων, που δίνουν την χωρική διάταξη της έντασης μίας ολόκληρης ή μερικής εικόνας. Οι υφές είναι, είτε δημιουργημένες τεχνητά, ή φυσικά διαμορφωμένες και δύνανται να συνεισφέρουν στην κατάτμηση και την ταξινόμηση της εικόνας. Η ανάλυση της υφής γίνεται με προσέγγιση είτε δομημένη ή στατιστική. Η στατιστική προσέγγιση είναι η ευκολότερη να μοντελοποιηθεί και χρησιμοποιείται ευρέως δεδομένης της ακανόνιστης διαρρύθμισης που τείνουν να έχουν οι φυσικές υφές.

Αρχικά γίνεται χρήση της ανίχνευσης ακμών μαζί με τον γεωμετρικό τους προσανατολισμό για τον προσδιορισμό των pixel που εν τέλει αντικατοπτρίζουν την πολυπλοκότητα της υφής. Θεωρώντας μία περιοχή μεγέθους N pixel, η ανίχνευση των ακμών της παράγει για κάθε σημείο, το μέγεθος της διαβάθμισης $Mag(p)$ και τον προσανατολισμό της $Dir(p)$.

Η οξύτητα των ακμών στην μονάδα εμβαδού ορίζεται:

$$F_{edge} = \frac{|\{p|Mag(p)| > T\}|}{N} \text{, για } T \text{ κατώφλι}$$

Αντίστοιχα, υπολογίζουμε τους πίνακες συνεμφάνισης**,** οι οποίοι συλλαμβάνουν αριθμητικά χαρακτηριστικά μίας υφής χρησιμοποιώντας χωρικές σχέσεις παρομοίων μονοχρωματικών τόνων. Αυτά τα αριθμητικά χαρακτηριστικά είναι χρήσιμα για την παράσταση, σύγκριση και ταξινόμηση υφών. Ακολούθως παρουσιάζονται μερικά από τα χαρακτηριστικά που παράγει ένας κανονικοποιημένος πίνακας συνεμφάνισης. Για $p[i,j]$ αντίστοιχο πίνακα μονοχρωματικής χωρικής εξάρτησης, και Ng το πλήθος των διακριτών μονοχρωματικών τόνων στην διακριτή εικόνα, ισχύει:

$$\text{Γωνιακή } 2^{\eta} \text{ ροπή } (\text{στροφορμή}) = \sum_i \sum_j p[i,j]^2$$

$$\text{Αντίθεση } (Contrast) = \sum_{n=0}^{Ng-1} n^2 \left\{ \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p[i,j] \right\}, \forall |i-j| = n$$

$$\text{Συσχέτιση } (Correlation) = \frac{\sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (ij) p[i,j] - \mu_x \mu_y}{\sigma_x \sigma_y}$$

$$\text{Εντροπία} = -\sum_i \sum_j p[i,j] \log(p[i,j])$$

1.4 Αναγνώριση Αντικειμένου

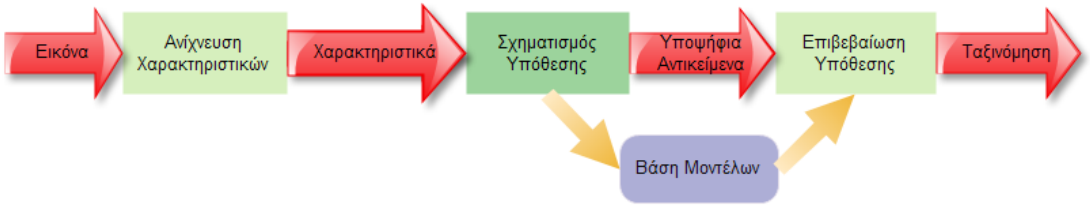
Ένα σύστημα αναγνώρισης αντικειμένου βρίσκει αντικείμενα στον πραγματικό κόσμο από μία εικόνα, χρησιμοποιώντας μοντέλα αντικειμένων τα οποία είναι γνωστά εκ των προτέρων. Το έργο αυτό ενέχει σημαντικές δυσκολίες που δεν είναι προφανείς. Οι εγκέφαλοι ζωντανών οργανισμών πραγματοποιούν αυτήν την διαδικασία φαινομενικά άκοπα και ακαριαία. Η αλγοριθμική περιγραφή αυτού του έργου για την υλοποίηση σε μηχανές εντούτοις, χρειάζεται περισσότερη προσπάθεια. Ακολούθως θα παρουσιαστούν τα διάφορα βήματα αυτής της διαδικασίας, καθώς και μερικές από τις πιο διαδεδομένες τεχνικές που χρησιμοποιούνται σε πολλαπλές εφαρμογές. Θα εκτεθούν οι διάφοροι τύποι διεργασιών αναγνώρισης, που καλείται ένα σύστημα αναγνώρισης να πραγματοποιήσει και θα αναλυθεί η περιπλοκότητα των διεργασιών αυτών, καθώς επίσης και οι προσεγγίσεις που είναι επωφελείς στις διαφορετικές φάσεις μίας τέτοιας διεργασίας.

Το πρόβλημα της αναγνώρισης, μπορεί να οριστεί ως πρόβλημα σήμανσης (labeling) βασισμένης πάνω σε μοντέλα γνωστών αντικειμένων. Τυπικά, δεδομένης μίας εικόνας η οποία εμπεριέχει ένα η περισσότερα αντικείμενα (ή φόντο) ενδιαφέροντος και ένα σύνολο σημάνσεων που αντιστοιχούν σε ένα σύνολο γνωστών μοντέλων του συστήματος, το σύστημα θα πρέπει να αναθέσει σωστές σημάνσεις σε περιοχές , η σύνολα περιοχών εντός της εικόνας. Το πρόβλημα της αναγνώρισης είναι στενά συνδεδεμένο με αυτό της κατάτμησης. Δηλαδή, χωρίς τουλάχιστον μία μερική αναγνώριση αντικειμένων, η κατάτμηση είναι ανέφικτη, και χωρίς κατάτμηση, η αναγνώριση αντικειμένου είναι αδύνατη.

1.4.1 Δομικά Συστατικά

Ένα σύστημα αναγνώρισης πρέπει να έχει τα ακόλουθα δομικά συστατικά.

- Βάση Μοντέλων (modelbase)
- Ανιχνευτής Χαρακτηριστικών (feature detector-extractor)
- Υπόθεση (hypothesizer)
- Επιβεβαίωση υπόθεσης (verifier)



1-xv αναγνώριση αντικειμένου

Η «βάση μοντέλων» περιέχει όλα τα γνωστά στο σύστημα μοντέλα (ακόμα και αν είναι μόνο ένα μοντέλο). Η πληροφορία στην βάση μοντέλων, εξαρτάται από την προσέγγιση στην αναγνώριση. Μπορεί να ποικίλει από ποιοτική ή λειτουργική περιγραφή, έως ακριβή γεωμετρική πληροφορία επιφάνειας. Σε πολλές περιπτώσεις μάλιστα, τα μοντέλα των αντικειμένων είναι αφηρημένα διανύσματα χαρακτηριστικών.

Ένα χαρακτηριστικό είναι κάποια ιδιότητα του αντικειμένου που κρίνεται αρκετά σημαντική για την περιγραφή και ταυτοποίηση του αντικειμένου σε σχέση με άλλα αντικείμενα. Τέτοια χαρακτηριστικά είναι συνήθως το μέγεθος, το χρώμα [36] και το σχήμα [37].

Ο «ανιχνευτής χαρακτηριστικών» διενεργεί με συγκεκριμένους τελεστές σε εικόνες και αναγνωρίζει θέσεις χαρακτηριστικών, που βοηθούν στον σχηματισμό της υπόθεσης. Τα χαρακτηριστικά που χρησιμοποιούνται από το σύστημα, εξαρτώνται από τον τύπο των αντικειμένων προς αναγνώριση και την εσωτερική οργάνωση της βάσης μοντέλων. Χρησιμοποιώντας τα ανιχνευμένα χαρακτηριστικά της εικόνας, η μηχανή υποθέσεων αναθέτει πιθανότητες σε αντικείμενα παρόντα στην σκηνή. Αυτό το βήμα χρησιμοποιείται, για να μειώσει τον χώρο αναζήτησης για την αναγνώριση χρησιμοποιώντας συγκεκριμένα γνωρίσματα. Η βάση μοντέλων οργανώνεται, κάνοντας χρήση κάποιου συστήματος ευρετηρίου για να διευκολύνει την απαλοιφή πιθανών εσφαλμένων υποψηφίων αντικειμένων από την θεώρηση. Έπειτα, η μηχανή επιβεβαίωσης χρησιμοποιεί μοντέλα αντικειμένων, για την διαπίστωση της ορθότητας της υπόθεσης και την τελειοποίηση της προσέγγισης της πιθανότητας των αντικειμένων. Στην συνέχεια το σύστημα, βάσει όλων των στοιχείων, επιλέγει το αντικείμενο με την μεγαλύτερη πιθανότητα ως το σωστό αντικείμενο. Αναλυτικά οι μέθοδοι ανίχνευσης χαρακτηριστικών, θα παρουσιαστούν σε επόμενο κεφάλαιο.

Όλα τα συστήματα αναγνώρισης αντικειμένων χρησιμοποιούν μοντέλα είτε προφανώς είτε αφανώς και επιστρατεύουν ανιχνευτές βάσει αυτών των μοντέλων. Ο σχηματισμός υπόθεσης και η επιβεβαίωση της διαφέρουν στην σημασία τους σε διαφορετικές προσεγγίσεις της αναγνώρισης. Κάποια συστήματα χρησιμοποιούν μόνο την αρχική υπόθεση και στην συνέχεια επιλέγουν το αντικείμενο με την μεγαλύτερη πιθανότητα να είναι το σωστό αντικείμενο. Σε αυτήν την κατηγορία ανήκουν αλγόριθμοι ταξινόμησης προτύπων. Πολλά συστήματα τεχνητής νοημοσύνης από την άλλη, εξαρτώνται ελάχιστα από την υπόθεση και αποδίδουν περισσότερο υπολογιστικό έργο στην φάση της επιβεβαίωσης. Σε κάποιες μάλιστα περιπτώσεις η υπόθεση παρακάμπτεται εξολοκλήρου.

Ένα σύστημα αναγνώρισης, πρέπει να επιλέγει τα κατάλληλα εργαλεία και τεχνικές για την υλοποίηση των βημάτων που προαναφέρθηκαν. Κατά την σχεδίαση ενός τέτοιου συστήματος πρέπει να λαμβάνονται υπόψη τα εξής κεντρικά θέματα:

- Παράσταση αντικειμένου- μοντέλου: Αυτό το θέμα αφορά τον τρόπο με τον οποίο τα αντικείμενα θα παρασταθούν στην βάση μοντέλων. Προσδιορίζονται τα σημαντικά χαρακτηριστικά, στατικά ή δυναμικά, τα οποία πρέπει να δεσμεύονται από το μοντέλο. Για κάποια αντικείμενα, πιθανόν να υφίστανται γεωμετρικοί περιγραφείς και να είναι επαρκείς, ενώ μία διαφορετική κλάση αντικειμένου μπορεί να εξαρτάται από πιο γενικά η λειτουργικά χαρακτηριστικά. Η παράσταση ενός αντικειμένου θα πρέπει να συλλαμβάνει όλη την σχετική πληροφορία χωρίς πλεονασμούς και θα πρέπει να οργανώνει την πληροφορία αυτή με τρόπο τέτοιο που θα επιτρέπει την γρήγορη πρόσβαση από διαφορετικά συστατικά του συστήματος αναγνώρισης.
- Εξαγωγή χαρακτηριστικών: Αυτή αφορά την επιλογή των χαρακτηριστικών που θα πρέπει να ανιχνευθούν και τον τρόπο αξιόπιστης ανίχνευσης τους. Τα περισσότερα χαρακτηριστικά μπορούν να αναχθούν υπολογιστικά σε δισδιάστατες εικόνες αλλά στην πραγματικότητα σχετίζονται με τρισδιάστατα χαρακτηριστικά αντικειμένων. Λόγω της φύσης της διαδικασίας σχηματισμού εικόνας, κάποια χαρακτηριστικά είναι ευκολότερο να υπολογιστούν αξιόπιστα, ενώ άλλα είναι πιο δύσκολα.

- Αντιστοίχιση μοντέλου- χαρακτηριστικού: Αυτό το θέμα αφορά το ταίριασμα των μοντέλων με τα αναγνωρισμένα χαρακτηριστικά στην εικόνα. Στα περισσότερα έργα αναγνώρισης αντικειμένου, συνυπάρχουν πολλά χαρακτηριστικά και πολλά αντικείμενα. Μία εξαντλητική προσέγγιση, θα λύσει το πρόβλημα της αναγνώρισης, αλλά μπορεί να είναι υπερβολικά αργή για να θεωρηθεί χρήσιμη. Η αξιοπιστία των ανιχνευμένων χαρακτηριστικών και η αποδοτικότητα της τεχνικής αντιστοίχισης, είναι σημαντικοί παράγοντες που πρέπει να λαμβάνονται υπόψη, όταν αναπτύσσεται μία στρατηγική αντιστοίχισης, σε συσχέτιση πάντα με το μέγεθος του πεδίου ορισμού του προβλήματος. Είθισται επίσης να συνυπολογίζονται συγκεκριμένες απαιτήσεις χρόνου εκτέλεσης της εφαρμογής, αλλά φυσικά και οι πιθανοί λογιστικοί περιορισμοί κόστους, ενέργειας και φυσικού μεγέθους του συστήματος.
- Σχηματισμός υπόθεσης: Σε αυτό το επίπεδο, το θέμα είναι η πιθανότητα ενός συνόλου αντικειμένων να επιλεγούν βάσει της αντιστοίχισης χαρακτηριστικών με το μοντέλο, αλλά και ο τρόπος που μπορεί να ανατεθούν πιθανότητες σε κάθε υποψήφιο αντικείμενο. Αυτό το βήμα ουσιαστικά είναι μία ευρετική προσέγγιση (heuristics) για να μειωθεί το μέγεθος του χώρου αναζήτησης. Αυτό το βήμα χρησιμοποιεί γνώση του πεδίου ορισμού της εφαρμογής για να αναθέσει κάποια μέτρα πιθανότητας ή βεβαιότητας στα διάφορα αντικείμενα στο πεδίο. Αυτά τα μέτρα αντικατοπτρίζουν την πιθανότητα παρουσίας αντικειμένων, βάσει των ανιχνευμένων χαρακτηριστικών.
- Επιβεβαίωση υπόθεσης αντικειμένου: Κατά λογική συνέχεια του παραπάνω σημείου, σε αυτό το βήμα επιλέγεται το αντικείμενο με την χρήση του μοντέλου μέσα από ένα σύνολο υποψηφίων αντικειμένων στην δεδομένη εικόνα. Αυτό αφορά κάθε διαφορετικού τύπου αντικείμενο έναντι του αντίστοιχου μοντέλου. Για κάθε ικανή υπόθεση, πρέπει να εξεταστεί η παρουσία αντικειμένου για να επιβεβαιώσει ή να απορρίψει την ύπαρξη του.

Ανάλογα με την περιπλοκότητα του προβλήματος, ένα η περισσότερα από τα στάδια της αρχικής ακολουθίας μπορεί να καταστεί τετριμμένο. Για παράδειγμα, η αναγνώριση αντικειμένου βάσει αναγνώρισης προτύπου (pattern recognition) δεν χρήζει αντιστοίχισης αντικειμένου μοντέλου ή επιβεβαίωση υπόθεσης αντικειμένου. Αναθέτει απευθείας πιθανότητες στα αντικείμενα και επιλέγει αυτό με μεγαλύτερο δείκτη πιθανότητας.

1.4.2 Περιπλοκότητα της αναγνώρισης αντικειμένου

Η ποιότητα μίας εικόνας γενικά μπορεί να εξαρτάται από τις συνθήκες φωτισμού, τις παραμέτρους της κάμερας και την τοποθέτηση της. Δεδομένου του ότι ένα αντικείμενο πρέπει να αναγνωριστεί από εικόνες μίας σκηνής που περιέχει πολλαπλές οντότητες, η περιπλοκότητα της αναγνώρισης εξαρτάται από πολυάριθμους παράγοντες. Μία ποιοτική προσέγγιση για τον προσδιορισμό της πολυπλοκότητας αυτής, πρέπει να λαμβάνει υπόψη τους ακόλουθους παράγοντες:

- Σταθερότητα προσκηνίου: Η περιπλοκότητα της σκηνής εξαρτάται από το αν οι εικόνες έχουν ληφθεί υπό όμοιες συνθήκες φωτισμού, παρασκηνίου, οπτικής γωνίας κ.α. με το μοντέλο. Υπό δραστικά διαφορετικές συνθήκες, η απόδοση διαφορετικών αλγορίθμων ανίχνευσης πλήττεται σημαντικά [38]. Η φύση του παρασκηνίου, η παρουσία άλλων αντικειμένων και ο φωτισμός πρέπει να συμπεριληφθούν στον καθορισμό των χαρακτηριστικών που μπορούν να ανιχνευθούν αξιόπιστα και αποδοτικά.
- Χώρος εικόνας-μοντέλου: Σε ορισμένες εφαρμογές, οι εικόνες λαμβάνονται με τρόπο τέτοιο όπου τρισδιάστατα αντικείμενα μπορούν να παρασταθούν δισδιάστατα. Τα μοντέλα τέτοιων περιπτώσεων παρίστανται με χρήση δισδιάστατων χαρακτηριστικών. Αν τα μοντέλα είναι τρισδιάστατα και δεν μπορούν να αγνοηθούν οι επιδράσεις της προοπτικής, τότε η κατάσταση περιπλέκεται περισσότερο. Σε αυτήν την περίπτωση, τα χαρακτηριστικά ανιχνεύονται σε δισδιάστατο χώρο εικόνας, ενώ τα αντικείμενα βρίσκονται σε τρισδιάστατο. Συνεπώς το ίδιο τρισδιάστατο χαρακτηριστικό μπορεί να εμφανιστεί ως διαφορετικό χαρακτηριστικό σε μία εικόνα. Αυτό είναι και από τα πιο συνήθη και σημαντικά προβλήματα της αναγνώρισης σε δυναμικές εικόνες λόγω της κίνησης του αντικειμένου. Η λύση δίνεται περιορίζοντας και καθορίζοντας ακριβώς το πεδίο της εφαρμογής.

- Πλήθος αντικειμένων στην βάση μοντέλων: Όταν ο αριθμός των αντικειμένων είναι πολύ μικρός, περιττεύει το στάδιο της υπόθεσης και πιθανώς να αρκεί μία εξαντλητική σειριακή αντιστοίχιση. Η υπόθεση καθίσταται σημαντική σε περιπτώσεις μεγάλου αριθμού αντικειμένων. Ο φόρτος της επιλογής κατάλληλων χαρακτηριστικών για την αναγνώριση αντικειμένου αυξάνεται ταχύτατα σε αναλογία με την αύξηση του πλήθους των αντικειμένων. Μία μικρή βάση μοντέλων ή μοντέλων με λίγα-ελλιπή χαρακτηριστικά καθιστά την εφαρμογή ευαίσθητη σε εσφαλμένη αναγνώριση αντικειμένων (false positives). Αντίθετα, υπερβολικό πλήθος χαρακτηριστικών οδηγεί σε πιθανή απώλεια θετικών αναγνωρίσεων (false negatives) και μεγαλύτερο υπολογιστικό φόρτο. Επίσης, όταν μία βάση μοντέλων περιέχει πολυάριθμα μοντέλα προς αναγνώριση, πρέπει να οργανωθεί σε δομές (π.χ. δένδρα) ικανές να κάνουν επαγωγική προσέγγιση της αντιστοίχισης για να βελτιώσει τον υπολογιστικό φόρτο και να ελαχιστοποιηθεί η λανθασμένη αναγνώριση παρομοίων αντικειμένων.
- Πλήθος αντικειμένων στην εικόνα και η πιθανότητα έμφραξης (occlusion): Όταν υπάρχει μόνο ένα αντικείμενο στην εικόνα, πιθανώς να είναι εξολοκλήρου ορατό. Αυξάνοντας τον αριθμό των αντικειμένων στην εικόνα, αυξάνεται ο κίνδυνος της έμφραξης. Η έμφραξη, το κρύψιμο δηλαδή ενός αντικειμένου μερικώς η πλήρως από άλλα αντικείμενα ή εμπόδια στον χώρο του εκτός οπτικού πεδίου της κάμερας, είναι ένα από τα σημαντικότερα σε πολλούς υπολογισμούς εικόνας. Η έμφραξη έχει ως αποτέλεσμα την απουσία αναμενόμενων χαρακτηριστικών ή την δημιουργία απρόοπτων άλλων. Γι αυτό τον λόγο πρέπει να συνυπολογίζεται στην διαδικασία της επιβεβαίωσης υπόθεσης. Γενικά η δυσκολία του έργου της αναγνώρισης αυξάνει με τον αριθμό αντικειμένων στην εικόνα. Δυσκολίες στην κατάτμηση της εικόνας, συχνά οφείλονται στην παρουσία πολλαπλών αλληλοφραγμένων αντικειμένων στην εικόνα.

Για να αντιμετωπιστούν οι διάφοροι παράγοντες που επηρεάζουν την αναγνώριση αντικειμένου, είθισται να κατηγοριοποιήσουμε το πρόβλημα της αναγνώρισης στις ακόλουθες κλάσεις:

- Δισδιάστατη αναγνώριση.
- Τρισδιάστατη αναγνώριση.
- Αναγνώριση κατάτμησης

1.4.3 Δισδιάστατη αναγνώριση

Σε πολλές εφαρμογές η εικόνα λαμβάνεται από απόσταση ικανή να καταστήσει την προβολή της σκηνής κατά προσέγγιση γεωμετρικά ορθή. Αν τα αντικείμενα είναι πάντα σε μία σταθερή θέση στην σκηνή, τότε μπορούν να θεωρηθούν δισδιάστατα. Σε αυτές τις εφαρμογές, δύναται να χρησιμοποιηθεί βάση μοντέλων δύο διαστάσεων. Έτσι υφίστανται οι εξής δύο περιπτώσεις.

- Τα αντικείμενα δεν θα υποστούν έμφραξη, όπως στις περιπτώσεις απομακρυσμένης αίσθησης και πολυάριθμων βιομηχανικών εφαρμογών.
- Τα αντικείμενα πιθανόν να επικαλυφθούν από άλλα αντικείμενα ενδιαφέροντος ή να είναι μερικώς ορατά όπως στην θεωρητική περίπτωση ενός κάδου με εξαρτήματα.

Σε κάποιες περιπτώσεις, ενώ τα αντικείμενα είναι μακριά, πιθανώς να εμφανιστούν σε διαφορετικές θέσεις, με αποτέλεσμα πολλαπλές σταθερές εικόνες. Τέτοιες περιπτώσεις μπορούν επίσης να θεωρηθούν εν γένει προβλήματα δισδιάστατης αναγνώρισης αντικειμένου.

1.4.4 Τρισδιάστατη αναγνώριση

Όταν οι εικόνες των αντικειμένων δύνανται να ληφθούν αυθαίρετα από διαφορετικές οπτικές γωνίες, τότε το αντικείμενο πιθανώς να εμφανιστεί πολύ διαφορετικό σε δύο διαφορετικές προοπτικές. Για την αναγνώριση αντικειμένου με χρήση τρισδιάστατων μοντέλων, πρέπει να συνυπολογιστεί η επιρροή της προοπτικής και το οπτικό πεδίο της εικόνας. Το γεγονός ότι τα μοντέλα είναι τρισδιάστατα και οι εικόνες περιέχουν μόνο δισδιάστατη πληροφορία, επηρεάζει την προσέγγιση της αναγνώρισης αντικειμένου. Εξακολουθεί να είναι εξέχουσας σημασίας η έμφραξη και ο διαχωρισμός των αντικειμένων. Ένας κοινά χρησιμοποιούμενος συμβιβασμός επιτυγχάνεται με την ομαδοποίηση των μοντέλων που αντιπροσωπεύουν το ίδιο αντικείμενο αλλά από διαφορετικές προοπτικές.

Για τις περιπτώσεις τρισδιάστατης αναγνώρισης, είθισται να οριστεί ο τύπος πληροφορίας που χρησιμοποιείται στο έργο της αναγνώρισης. Οι δύο διαφορετικές περιπτώσεις είναι οι ακόλουθες.

- Ένταση: Δεν είναι ρητά διαθέσιμη οποιαδήποτε πληροφορία για την επιφάνεια σε εικόνες έντασης. Με χρήση τιμών έντασης, αναγνωρίζονται χαρακτηριστικά σε αντιστοιχία με την τρισδιάστατη δομή του αντικειμένου.
- Εικόνες 2.5 διαστάσεων: Σε πολλές εφαρμογές είναι διαθέσιμες, ή δύνανται να υπολογιστούν αναπαραστάσεις επιφάνειας με σχετικό προς τον παρατηρητή σύστημα συντεταγμένων από τις εικόνες. Αυτή η πληροφορία είναι χρήσιμη στην αναγνώριση αντικειμένου καθώς δίνει την απόσταση από διαφορετικά σημεία στην εικόνα προς ένα συγκεκριμένο σημείο.
- Στερεοσκοπική Όραση [39]: Εμπνευσμένη από την φύση, αυτή η μέθοδος αναγνώρισης βασίζεται στην διαφορική εικόνα που λαμβάνεται από δύο αισθητήρες με μικρή απόσταση μεταξύ τους. Η διαφορά στην λαμβανόμενη εικόνα, συνήθως δίνει με μεγάλη ευκολία περιγράμματα αντικειμένων που βρίσκονται σε απόσταση από το παρασκήνιο τους και βοηθά στην εκτίμηση της απόστασης αυτής. Η μέθοδος αυτή, είναι ιδιαίτερα αποδοτική στις περιπτώσεις, που το αντικείμενο έχει χρώμα ή υφή παραπλήσιο με το παρασκήνιο του και συνεπώς η εξαγωγή ορίων με χρήση των συνηθισμένων αλγορίθμων ανίχνευσης ακμών είναι αναποτελεσματική.

1.4.5 Αναγνώριση Κατάτμησης

Όταν οι εικόνες έχουν ήδη κατατμηθεί σε διαφορετικά αντικείμενα και διαφοροποιηθεί από το παρασκήνιο τους, είναι εύκολη η αναγνώρισή τους. Γενικά το πρόβλημα της αναγνώρισης είναι άμεσα συνδεδεμένο με το πρόβλημα της κατάτμησης.

1.4.6 Αναπαράσταση αντικειμένου

Σε γενικές γραμμές, ως αντικείμενο μπορεί να οριστεί οτιδήποτε ενδιαφέρον για ανάλυση. Για παράδειγμα, βάρκες στην θάλασσα, ψάρια στο ενυδρείο, οχήματα στον δρόμο, αεροπλάνα στον αέρα, άνθρωποι στον δρόμο, αποτελούν σύνολα αντικειμένων που ενδέχεται να είναι αρκετά σημαντικά ώστε να ανιχνευθούν και να παρακολουθηθούν σε ένα συγκεκριμένο πεδίο εφαρμογής. Ειδικότερα για την παρακολούθηση, τα αντικείμενα αναπαρίστανται με βάση το σχήμα και άλλα χαρακτηριστικά της εμφάνισης τους. Οι πιο κοινοί τρόποι αναπαράστασης ενός αντικειμένου είναι οι ακόλουθοι:

- Σημεία: Το αντικείμενο παρίσταται από ένα σημείο που ονομάζεται «κεντροειδές», δηλαδή το κέντρο βάρους (γεωμετρικά), ή από ένα σύνολο σημείων που αντιπροσωπεύουν επιμέρους σημεία γεωμετρικής βαρύτητας. Γενικά η παράσταση με σημεία, είναι κατάλληλη για αναγνώριση και παρακολούθηση αντικειμένων που καταλαμβάνουν μικρές περιοχές στην εικόνα.
- Αρχέγονα γεωμετρικά σχήματα: Το σχήμα του αντικειμένου παρίσταται με ορθογώνια και ελλείψεις μεταξύ άλλων. Η αναγνώριση αντικειμένων αυτής της παράστασης, συνήθως μοντελοποιείται με μεθόδους μετάφρασης, συσχετισμένου ή προβολικού μετασχηματισμού ομογραφίας (όπως για παράδειγμα η ομογραφία RANSAC [7]). Παρόλο που τα αρχέγονα γεωμετρικά σχήματα είναι καταλληλότερα για την παράσταση απλών συμπαγών αντικειμένων, μπορούν να χρησιμοποιηθούν και για την παρακολούθηση άλλων.
- Σκιαγραφία και περίγραμμα (silhouette and contours): Η παράσταση περιγραμμάτων καθορίζει το όριο ενός αντικειμένου. Η περιοχή εσωτερικά του περιγράμματος ονομάζεται σκιαγραφία του αντικειμένου. Αυτός ο τύπος παράστασης είναι κατάλληλος για αντικείμενα με περίπλοκα και μη συμπαγή σχήματα.
- Αρθρωτά μοντέλα σχήματος: Αυτά τα αντικείμενα συντίθενται από μέλη τα οποία συνδέονται σε αρθρώσεις. Ένα τέτοιο παράδειγμα, είναι το ανθρώπινο σώμα του οποίου τα αρθρωτά μέλη είναι ο κορμός, τα πόδια, τα χέρια το κεφάλι κ.λπ. τα οποία συνδέονται με τις αρθρώσεις. Η σχέση μεταξύ των μελών, περιγράφεται με κινηματικά μοντέλα. Προκειμένου να παρασταθεί ένα αρθρωτό αντικείμενο, πρέπει αρχικά να μοντελοποιηθούν τα συστατικά μέλη χρησιμοποιώντας κυλίνδρους ή ελλείψεις.

- Σκελετικά μοντέλα: Ο σκελετός του αντικειμένου μπορεί να εξαχθεί εφαρμόζοντας μετασχηματισμό μεσαίου άξονα από την μαθηματική μορφολογία. Αυτό το μοντέλο χρησιμοποιείται συνήθως ως παράσταση σχήματος για την αναγνώριση αντικειμένων, τόσο αρθρωτών, όσο και συμπαγών.

Υπάρχουν πολυάριθμοι τρόποι παράστασης των χαρακτηριστικών εμφάνισης ενός αντικειμένου. Αξιοσημείωτο είναι, ότι οι παραστάσεις σχήματος μπορούν να συνδυαστούν με την παράσταση εμφάνισης προς παρακολούθηση. Στο γενικό πλαίσιο της παρακολούθησης, κάποιες συνήθεις παραστάσεις εμφάνισης είναι οι ακόλουθες.

- Πιθανοτικές πυκνότητες εμφάνισης αντικειμένου. Οι εκτιμήσεις πυκνότητας πιθανότητας της εμφάνισης του αντικειμένου, παρίστανται είτε παραμετρικά μέσω συναρτήσεων Gauss, ή με μίξη συναρτήσεων Gauss [40], άλλα και μη παραμετρικά όπως τα παράθυρα Parzen και τα ιστογράμματα. Οι πιθανοτικές πυκνότητες χαρακτηριστικών αντικειμένου όπως χρώμα και υφή [41] μπορούν να υπολογιστούν από περιοχές εικόνας, που έχουν προσδιοριστεί από σχηματικά μοντέλα, όπως εσωτερική επιφάνεια μίας έλλειψης ή ενός περιγράμματος.
- Πρότυπα περιγράμματος: Αυτά δημιουργούνται χρησιμοποιώντας απλά γεωμετρικά σχήματα ή σκιαγραφίες. Το πλεονέκτημά τους είναι ότι φέρουν χωρικά δεδομένα επιπροσθέτως των εμφανισιακών χαρακτηριστικών. Προϋπόθεση είναι η κωδικοποίηση της εμφάνισης μέσω μίας και μόνο προοπτικής.
- Ενεργά μοντέλα εμφάνισης: Αυτά δημιουργούνται από την ταυτόχρονη μοντελοποίηση του σχήματος και της εμφάνισης. Σε γενικές γραμμές, το σχήμα του αντικειμένου προσδιορίζεται από ένα σύνολο οροσήμων. Όμοια με την παράσταση με πρότυπα περιγράμματος, τα ορόσημα μπορούν να εμπεριέχονται στο όριο του αντικειμένου ή εναλλακτικά εντός της περιοχής του αντικειμένου. Για κάθε ορόσημο, αποθηκεύεται ένα διάνυσμα με μορφή χρώματος, υφής ή μέτρου διαβάθμισης. Τα ενεργά μοντέλα εμφάνισης χρήζουν μίας εκπαιδευτικής φάσης, όπου το σχήμα και η συσχετιζόμενη εμφάνιση εκπαιδεύονται από ένα σύνολο δειγμάτων χρησιμοποιώντας για παράδειγμα την ανάλυση κύριου συστατικού [42].
- Μοντέλα εμφάνισης πολλαπλών προοπτικών: Από αυτά κωδικοποιούνται διάφορες οπτικές γωνίες ενός αντικειμένου. Μία προσέγγιση για να παρασταθούν διαφορετικές προοπτικές ενός αντικειμένου, είναι η δημιουργία ενός υποχώρου των δεδομένων προοπτικών. Τέτοιες προσεγγίσεις είναι για παράδειγμα η Ανάλυση Κυρίου Συστατικού (Principal Component Analysis – PCA) και η Ανάλυση Ανεξάρτητου Συστατικού (Independent Component Analysis – ICA). Έχουν χρησιμοποιηθεί και οι δύο για παράσταση εμφάνισης και σχήματος.

Μία ακόμη προσέγγιση εκπαίδευσης των διαφορετικών προοπτικών ενός αντικειμένου, είναι η εκπαίδευση ενός συνόλου από «ταξινομείς», όπως για παράδειγμα οι Μηχανές Υποστήριξης Διανύσματος (Support Vector Machine – SVM) [43] ή τα στατιστικά δίκτυα Bayes (Bayesian Networks) [44].

1.5 Αλγόριθμοι Ανίχνευσης και Αναγνώρισης Αντικείμενου

Κάθε μέθοδος παρακολούθησης αντικειμένου, χρειάζεται έναν μηχανισμό ανίχνευσης αντικειμένου είτε σε κάθε καρέ, η όταν το αντικείμενο εμφανιστεί για πρώτη φορά στο βίντεο. Μία συνήθης προσέγγιση για την ανίχνευση αντικειμένου είναι η χρήση της πληροφορίας από μονά καρέ. Ωστόσο κάποιες μέθοδοι ανίχνευσης αντικειμένου, κάνουν χρήση χρονικών πληροφοριών, που έχουν υπολογισθεί από αλληλουχίες εικόνων για να ελαχιστοποιήσουν το πλήθος των εσφαλμένων ανιχνεύσεων [45]. Αυτές οι πληροφορίες είναι συνήθως στην μορφή διαφορών μεταξύ διαδοχικών καρέ, επισημαίνοντας τις περιοχές των εικόνων αυτών που παρουσιάζουν μεταβολές. Τροφοδοτώντας τις περιοχές αυτές της εικόνας, καθίσταται καθήκον του αλγορίθμου παρακολούθησης, να πραγματοποιήσει την αντιστοιχία από το ένα καρέ στο επόμενο, δημιουργώντας έτσι τροχιές.

Οι πιο διαδεδομένες μέθοδοι ανίχνευσης, μαζί με κάποια από τα αντιπροσωπευτικότερα έργα τους ανά κατηγορία, συνοψίζονται ακολούθως:

- Ανιχνευτές σημείων:
  - Ανιχνευτής Moravec [46]
  - Ανιχνευτής Harris [47]
  - Μετασχηματισμός Χαρακτηριστικών Ανεξαρτήτως Κλίμακας /Scale Invariant Feature Transform-SIFT [48][49]
  - Ανιχνευτής Σημείων Ανεξαρτήτως Συσχετισμού /Affine Invariant Point Detector [50]
- Κατάτμηση:
  - Μέση Μετατόπιση /Mean-shift [51]
  - Περικομμένου Γραφήματος /Graph-cut [52]
  - Ενεργά Περιγράμματα /Active Contours [53]
- Μοντελοποίηση Παρασκηνίου:
  - Μίξη συναρτήσεων Gauss /Mixture Of Gaussians –MOG [54]
  - Ιδιο-παρασκηνίου /Eigenbackground [55]
  - Wall flower [56]
  - Παρασκήνιο Δυναμικής Υφής / Dynamic Texture Background [57]
- Μοντελοποίηση Παρασκηνίου:
  - Μηχανές Διανυσμάτων Υποστήριξης /Support Vector Machines –SVM [58]
  - Νευρωνικά Δίκτυα [59]
  - Προσαρμοστική Ενίσχυση /Adaptive Boosting- Adaboost [60]

1.5.1 Ανίχνευση σημείων

Οι ανιχνευτές σημείων, χρησιμοποιούνται για να βρίσκουν σημεία ενδιαφέροντος σε εικόνες που παρουσιάζουν εκφραστικές υφές στις σχετικές τοπικότητές τους. Τα σημεία ενδιαφέροντος, έχουν εκτεταμένα χρησιμοποιηθεί στο πλαίσιο προβλημάτων όπως της κίνησης, της στερεοσκοπίας και της παρακολούθησης. Μία ελκυστική αρετή των σημείων ενδιαφέροντος είναι η αμεταβλητότητα τους σε αλλαγές στον φωτισμό και στην προοπτική της κάμερας. Στην βιβλιογραφία ορισμένοι από τους συχνότερα χρησιμοποιούμενους ανιχνευτές σημείων περιλαμβάνουν τον Τελεστή Ενδιαφέροντος Moravec, τον Ανιχνευτή Σημείων Ενδιαφέροντος Harris, τον ανιχνευτή KLT [39] και φυσικά τον SIFT. Ο Τελεστής του Moravec, προκειμένου να εντοπίσει σημεία ενδιαφέροντος, υπολογίζει τις αλλαγές στις εντάσεις της εικόνας με έναν πυρήνα σάρωσης (παράθυρο) 4X4 σε κατευθύνσεις οριζόντιες, κάθετες, διαγώνιες και αντιδιαγώνιες και επιλέγει το ελάχιστο από τις τέσσερεις παραλλαγές ως αντιπροσωπευτική τιμή για το δεδομένο παράθυρο. Ένα σημείο κηρύσσεται ενδιαφέρον εφόσον η παραλλαγή της έντασης είναι η τοπικά μέγιστη σε ένα χώρο 12X12.

Ο ανιχνευτής Harris υπολογίζει τις παραγώγους πρώτου βαθμού της εικόνας (I_x, I_y), στις κατευθύνσεις x, y για να επισημάνει τις κατευθυντικές παραλλαγές έντασης, και στην συνέχεια ένας πίνακας δεύτερων ροπών που κωδικοποιεί την παραλλαγή, αξιολογείται για κάθε pixel σε μικρότερη γειτνίαση.

$$M = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}$$

Ένα σημείο ενδιαφέροντος ταυτοποιείται χρησιμοποιώντας την ορίζουσα και το ίχνος της ροπής M, το οποίο μετρά την παραλλαγή στην τοπική περιοχή R, όπου k σταθερά.

$$R = \det(M) - k \cdot tr(M)^2$$

Τα σημεία ενδιαφέροντος σημειώνονται κατωφλιώνοντας την R αφού εφαρμοστεί καταστολή των σημείων που δεν βρίσκονται σε σημαντικές ακμές (nonmaxima suppression).

Τα σημεία ενδιαφέροντος σημειώνονται κατωφλιώνοντας την R αφού εφαρμοστεί καταστολή των σημείων που δεν βρίσκονται σε σημαντικές ακμές (nonmaxima suppression).

Ο ίδιος πίνακας ροπών M χρησιμοποιείται στο βήμα ανίχνευσης σημείων ενδιαφέροντος του αλγορίθμου KLT. Η βεβαιότητα του σημείου ενδιαφέροντος R, υπολογίζεται χρησιμοποιώντας την ελάχιστη ιδιοτιμή του M, λ_{min} . Υποψήφια σημεία ενδιαφέροντος επιλέγονται κατωφλιώνοντας την R . Μεταξύ των υποψηφίων σημείων, ο KLT απαλείφει τα υποψήφια που βρίσκονται χωρικά παραπλησίως.

Ποσοτικά, τόσο ο Harris, όσο και ο KLT δίνουν έμφαση στις μεταβολές της έντασης με πολύ παρόμοια μέτρα. Για παράδειγμα, η R στον Harris σχετίζεται με το χαρακτηριστικό πολυώνυμο που χρησιμοποιείται για να βρεθούν οι ιδιοτιμές του M.

$$\lambda^2 + \det(M) - \lambda \cdot tr(M) = 0$$

Αντίθετα ο KLT υπολογίζει τις ιδιοτιμές απευθείας. Στην πράξη, και οι δύο αλγόριθμοι τείνουν να βρουν τα ίδια σημεία ενδιαφέροντος. Η βασική διαφορά είναι το επιπλέον κριτήριο που επιβάλλει ο KLT για την προκαθορισμένη απόσταση μεταξύ ανιχνευμένων σημείων ενδιαφέροντος.

Στην θεωρία, ο πίνακας Mείναι αμετάβλητος τόσο σε περιστροφή όσο και σε μετάφραση. Ωστόσο, δεν είναι αμετάβλητος σε συσχετισμένους ή προβολικούς μετασχηματισμούς. Προκειμένου να εγκαθιδρύσει έναν συμπαγή τρόπο ανίχνευσης σημείων ενδιαφέροντος υπό διαφορετικούς μετασχηματισμούς, το 2004 ο David G. Lowe εισήγαγε τον αλγόριθμο SIFT (Scale Invariant Feature Transform) [48] [49], ο οποίος αποτελείται από τέσσερα βήματα, τα οποία θα αναλυθούν στο κεφάλαιο της εφαρμογής. Εν συντομία παρουσιάζονται ακολούθως.

- Αρχικά ορίζεται ένας χώρος κλίμακας, συνελίσσοντας την εικόνα με φίλτρα Gauss σε διαφορετικές κλίμακες (αναλογίες). Οι συνελιγμένες εικόνες χρησιμοποιούνται για να παράγουν εικόνες διαφορών Gauss (Difference-of-Gaussians). Τα υποψήφια σημεία ενδιαφέροντος επιλέγονται στην συνέχεια από τα ελάχιστα και τα μέγιστα αυτών των διαφορών DoG.
- Στην συνέχεια αναβαθμίζονται οι θέσεις κάθε υποψηφίου σημείου, παρεμβάλλοντας τις χρωματικές τιμές αξιοποιώντας γειτονικά pixel.
- Ακολούθως, απαλείφονται υποψήφια σημεία ενδιαφέροντος με χαμηλή αντίθεση, καθώς και αυτά που βρίσκονται πάνω σε ακμές.
- Τελικά, στα εναπομείναντα σημεία ενδιαφέροντος, ανατίθενται προσανατολισμοί βάσει των κορυφών στα ιστογράμματα των κατευθύνσεων διαβάθμισης σε μικρή γειτνίαση με το εκάστοτε υποψήφιο σημείο.

Ο ανιχνευτής SIFT παράγει μεγαλύτερο αριθμό σημείων ενδιαφέροντος, συγκριτικά με τους άλλους ανιχνευτές σημείων ενδιαφέροντος [61]. Αυτό οφείλεται στο γεγονός, ότι συσσωρεύονται τα σημεία ενδιαφέροντος σε διαφορετικές κλίμακες και διαφορετικές αναλύσεις (πυραμίδα Gaussian-Laplacian). Εμπειρικά, έχει δειχθεί από ερευνητές, ότι ο αλγόριθμος SIFT ξεπερνά σε επιδόσεις τους περισσότερους ανιχνευτές σημείων ενδιαφέροντος και είναι πιο ανθεκτικός σε παραμορφώσεις εικόνας [62]. Για τον λόγο αυτό επιλέχτηκε ο αλγόριθμος αυτός για να υλοποιηθεί στο πειραματικό μέρος αυτής της εργασίας.

Από την διάδοση του αλγορίθμου SIFT (Scale Invariant Feature Transform) έχουν προταθεί και εφαρμοστεί πολλαπλοί αλγόριθμοι [63][7] με παρόμοια ή επεκταμένη φιλοσοφία και προφανή, χαρακτηριστική απόπειρα πρωτοτυπίας στην ονοματολογία τους βάσει αρχικών γραμμάτων [64]. Μεταξύ των πιο αξιοσημείωτων είναι οι ακόλουθοι:

- SURF (Speeded Up Robust Features)
- BRISK (Binary Robust Invariant Scalable Keypoints) [65]
- FREAK (Fast Retina Keypoint) [66]
- BRIEF (Binary Robust Independent Elementary Features) [63]
- FAST (Features from Accelerated Segment Test)
- ORB (Oriented FAST and Rotated BRIEF) [67]

Συγκεκριμένα ο αλγόριθμος SURF [68] ο οποίος προστατεύεται και με πατέντα ως πνευματική ιδιοκτησία για εμπορική χρήση, είναι ο πιο καθιερωμένος σήμερα λόγω της ταχύτητας και της αξιοπιστίας του. Συνεπώς, η δημιουργία παραλλαγών του, που τον υλοποιούν σε βελτιωμένη, ή και παράλληλη μορφή [8] [69] είναι συχνό φαινόμενο,. Συνοπτικά η λειτουργία του είναι η ακόλουθη:

- Αρχικά εφαρμόζεται ένας ανιχνευτής Fast-Hessian υλοποιώντας την ορίζουσα του Hessian πίνακα για την τοποθεσία και την κλίμακα. Δηλαδή για ένα σημείο $x = (\chi, \upsilon)$ και αρχική εικόνα I, ο πίνακας αυτός για κλίμακα μεγέθους σ ορίζεται ως εξής:

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

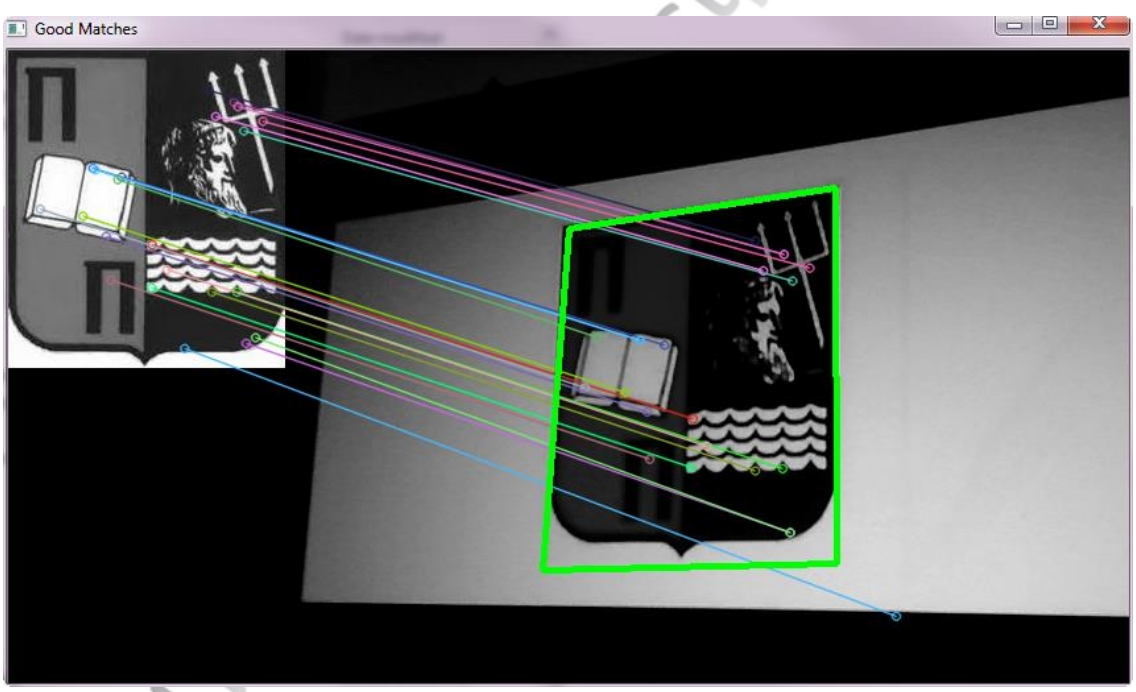
Όπου $L_{xx}(x,\sigma)$ είναι η συνέλιξη της δεύτερης τάξης μερικής Gauss παραγώγου:

$$\frac{\partial^2}{\partial x^2} g(\sigma)$$

και ομοίως για $L_{xy}(x,\sigma), L_{xy}(x,\sigma), L_{yy}(x,\sigma)$.

- Στην συνέχεια πραγματοποιείται ένας περιγραφέας SIFT λόγω της υψηλής απόδοσής του, αλλά με παραλειπόμενα ορισμένα γνωρίσματα που προσδίδουν περιπλοκότητα και υπολογιστικό φόρτο. Αρχικά διαρρυθμίζεται μία επαναλήψιμη γωνιακή τοποθέτηση, βάσει της πληροφορίας από μια κυκλική περιοχή γύρω από το σημείο ενδιαφέροντος. Ακολούθως, κατασκευάζεται μια τετράγωνη περιοχή ευθυγραμμισμένη με την επιλεγμένη γωνία τοποθέτησης. Στην συνέχεια εξάγεται ο περιγραφέας. Αυτό γίνεται χωρίζοντας αρχικά την περιοχή σε μικρότερα τετράγωνα τμήματα 4×4. Αυτό διατηρεί σημαντικές χωρικές πληροφορίες. Για κάθε τμήμα υπολογίζονται κάποια απλά χαρακτηριστικά με διάκενο πλέγμα 5×5. Έπειτα οι απαντήσεις-κυμάτια d_x, d_y που παρήγαγε ο Haar [6] για την κάθε περιοχή, αθροίζονται και σχηματίζουν μία νέα εισαγωγή στο διάνυσμα χαρακτηριστικών. Προκειμένου να συμπεριληφθούν πληροφορίες για την πολικότητα των αλλαγών έντασης, εξάγεται το άθροισμα των απολύτων τιμών για τις απαντήσεις. Κάθε τμήμα περιέχει ένα τετραδιάστατο διάνυσμα περιγραφέα v για την υποκείμενη δομή έντασης.
- $$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

Αυτό έχει ως αποτέλεσμα το διάνυσμα περιγραφέα για όλες τις υποπεριοχές 4×4 να έχει συνολικό μήκος 64.



1-xvi SURF: αναγνώριση λογότυπου σε ζωντανό βίντεο.

1.5.2 Ανίχνευση Χαρακτηριστικών

Για την αναγνώριση αντικειμένων χρησιμοποιούνται πολλοί τύποι χαρακτηριστικών. Τα περισσότερα χαρακτηριστικά είναι βασισμένα είτε σε περιοχές ή σε όρια μίας εικόνας. Θεωρείται ότι μία περιοχή ή ένα κλειστό όριο – περίγραμμα, αντιστοιχεί σε μία οντότητα που είναι είτε αντικείμενο ή μέρος αντικειμένου. Κάποια από τα πιο κοινώς χρησιμοποιούμενα χαρακτηριστικά αναφέρονται συνοπτικά στις ακόλουθες ενότητες.

1.5.3 Καθολικά Χαρακτηριστικά (Global Features)

Αυτά είναι συνήθως κάποια γνωρίσματα μίας περιοχής στις εικόνες, όπως το εμβαδό / μέγεθος, η περίμετρος, οι περιγραφείς Fourier και οι ροπές (moments). Καθολικά χαρακτηριστικά μπορούν να εξαχθούν είτε από μία περιοχή, λαμβάνοντας υπόψη όλα τα σημεία εντός της περιοχής, ή μόνο εκείνα που βρίσκονται στο όριό της. Σε κάθε περίπτωση, η πρόθεση είναι να βρεθούν περιγραφείς που αποκτώνται συνυπολογίζοντας όλα τα σημεία, τις περιοχές τους, τα χαρακτηριστικά έντασης, και τις χωρικές συσχετίσεις τους.

- *Τοπικά Χαρακτηριστικά (Local Features).*

  Τα τοπικά χαρακτηριστικά είναι συνήθως, στο όριο ενός αντικειμένου ή αντιπροσωπεύουν μία διακριτή μικρή έκταση μίας περιοχής. Η καμπυλότητα και άλλες συγγενικές ιδιότητες όπως τμήματα ορίων και γωνίες, συχνά χρησιμοποιούνται ως τοπικά χαρακτηριστικά. Η καμπυλότητα ενδέχεται να είναι η καμπυλότητα ενός ορίου, ή δύναται να υπολογισθεί για μία επιφάνεια. Η επιφάνεια αυτή μπορεί να παρίσταται από την ένταση ή να ανήκει στον χώρο των 2.5 διαστάσεων, στον γεωμετρικό χώρο, δηλαδή που ενώ παρίσταται σε δύο διαστάσεις, μπορεί και παρέχει την αντίληψη βάθους. Σημεία υψηλής καμπυλότητας είθισται να καλούνται γωνίες και παίζουν σημαντικό ρόλο στην αναγνώριση αντικειμένου.
- *Σχεσιακά Χαρακτηριστικά (Relational Features).*

  Αυτά βασίζονται στις σχετικές θέσεις διαφορετικών οντοτήτων, είτε περιοχές, κλειστά περιγράμματα, ή τοπικά χαρακτηριστικά. Αυτά τα γνωρίσματα συνήθως περιέχουν την απόσταση μεταξύ χαρακτηριστικών και τις μετρήσεις σχετικού προσανατολισμού. Είναι συνεπώς, ιδιαίτερα χρήσιμα στον καθορισμό σύνθετων χαρακτηριστικών, αξιοποιώντας πολλαπλές περιοχές ή τοπικά χαρακτηριστικά στην εικόνα. Στις περισσότερες περιπτώσεις, η σχετική θέση των οντοτήτων είναι αυτό που καθορίζει τα αντικείμενα. Το ακριβώς ίδιο γνώρισμα, σε μία ελάχιστα παραλλαγμένη σχέση , μπορεί να εξάγει εντελώς διαφορετικό αντικείμενο.

Σε πολλές περιπτώσεις μπορούν να χρησιμοποιηθούν χαρακτηριστικά για την περιγραφή ενός αντικειμένου τόσο Καθολικά όσο και Τοπικά. Οι συσχετίσεις μεταξύ αντικειμένων, μπορούν να χρησιμοποιηθούν για να σχηματίσουν σύνθετα / μικτά χαρακτηριστικά.

1.5.4 Αφαίρεση Παρασκηνίου (Background Subtraction)

Ανίχνευση αντικειμένων μπορεί να επιτευχθεί χτίζοντας μία αναπαράσταση της σκηνής που ονομάζεται Μοντέλο Παρασκηνίου, και στην συνέχεια βρίσκοντας αποκλίσεις από το μοντέλο για κάθε συνεχόμενο καρέ μίας ακολουθίας βίντεο. Οποιαδήποτε σημαντική αλλαγή στην περιοχή της εικόνας από το μοντέλο παρασκηνίου, σηματοδοτεί ένα κινούμενο αντικείμενο. Τα pixel που αποτελούν τις περιοχές που υφίστανται αλλαγή, σηματοδοτούνται για περεταίρω ανάλυση και επεξεργασία. Συνήθως, συνδεδεμένοι συστατικοί αλγόριθμοι εφαρμόζονται για να αποκτηθούν συνδεδεμένες περιοχές που αντιστοιχούν στο αντικείμενο. Αυτή η διαδικασία ονομάζεται Αφαίρεση Παρασκηνίου.

Η διαφοροποίηση των χρονικά διαδοχικών καρέ έχει μελετηθεί διεξοδικά από την δεκαετία του 1970. Ωστόσο, η ευρεία διάδοσή της επήλθε μετά το έργο των Wren et al. το 1997 [70]. Προκειμένου να εκπαιδευτεί για τις βαθμιαίες αλλαγές σε βάθος χρόνου, ο αλγόριθμος που πρότεινε ο Wren μοντελοποιεί το χρώμα κάθε pixel, $I(x,y)$, ενός στατικού παρασκηνίου με μία μοναδική τρισδιάστατη συνάρτηση Gauss $I(x,y) \sim N\big(\mu(x,y), \Sigma(x,y)\big)$ στον χρωματικό χώρο YUV. Οι παράμετροι του μοντέλου, το μέσο $\mu(x,y)$ και η συνδιακύμανση $\Sigma(x,y)$, εκπαιδεύονται από τις χρωματικές παρατηρήσεις σε διάφορα διαδοχικά καρέ. Εφόσον το μοντέλο παρασκηνίου έχει παραχθεί, για κάθε pixel $x,y)$ στην εικόνα εισόδου, η πιθανότητα του χρώματός του να έρχεται από τις περιοχές $N\big(\mu(x,y)\big), \Sigma(x,y)$ υπολογίζεται, και τα pixel που αποκλίνουν από το μοντέλο παρασκηνίου επισημαίνονται ως ανήκοντα στο προσκήνιο της εικόνας, δηλαδή το αντικείμενο. Ωστόσο, μία μονή συνάρτηση Gauss, δεν είναι επαρκές μοντέλο για υπαίθριες εικόνες, μιας και πολλαπλά χρώματα μπορούν να παρατηρηθούν σε συγκεκριμένες θέσεις λόγω σχετικής κίνησης του αντικειμένου, σκιές ή ανακλάσεις. Μία σημαντική βελτίωση στην μοντελοποίηση παρασκηνίου, επιτυγχάνεται χρησιμοποιώντας πολυτροπικά στατιστικά μοντέλα για να περιγραφεί το χρώμα του παρασκηνίου ανά pixel. Για παράδειγμα, οι Stauffer και Grimson το 2000 [54] χρησιμοποίησαν μίξη συναρτήσεων Gauss (MoG), για να μοντελοποιήσουν το χρώμα των pixel. Σε αυτήν την μέθοδο, κάθε pixel στο τρέχον καρέ ελέγχεται σε σχέση με το μοντέλο παρασκηνίου, συγκρίνοντάς το με κάθε Gauss συνάρτηση στο παρασκήνιο μέχρι να βρεθεί η αντίστοιχη εκεί. Εφόσον βρεθεί, αναθεωρούνται το μέσο και η διακύμανση της αντιστοιχούσας συνάρτησης Gauss, ειδάλλως, μία νέα συνάρτηση εισάγεται στην μίξη συναρτήσεων MoG ίση με το τρέχον χρώμα pixel, καθώς και μία αρχική διακύμανση. Κάθε pixel ταξινομείται, βάσει του κατά πόσον η αντίστοιχη κατανομή, αντιπροσωπεύει την διαδικασία του παρασκηνίου.

Μία ακόμη προσέγγιση, είναι η ενσωμάτωση χωρικής πληροφορίας σκηνής, έναντι της αποκλειστικής χρήσης πληροφορίας βάσει χρώματος. Οι Elgammal και Davis το 2000 [71] χρησιμοποίησαν μη παραμετρική εκτίμηση πυκνότητας πυρήνα για να μοντελοποιήσουν το παρασκήνιο ανά pixel. Ως εκ τούτου, η μέθοδός τους μπορεί να διαχειριστεί τρεμάμενη εικόνα από την κάμερα ή μικρές κινήσεις στο παρασκήνιο.

Οι Li και Leung το 2002 [72], συγχώνευσαν τα χαρακτηριστικά υφής και χρώματος για να πραγματοποιήσουν αφαίρεση παρασκηνίου πάνω σε μπλοκ pixel μεγέθους 5X5. Επειδή η υφή δεν διαφοροποιείται σημαντικά με τις αλλαγές της φωτεινότητας, η μέθοδος αυτή είναι πιο αξιόπιστη σε τέτοιες συνθήκες. Οι Toyama et al. το 1999 [56] πρότειναν έναν αλγόριθμο τριών βαθμίδων, για να επιλύσουν το πρόβλημα της αφαίρεσης παρασκηνίου. Πέρα από την αφαίρεση σε επίπεδο pixel, χρησιμοποιούν την πληροφορία σε επίπεδο καρέ και περιοχής της εικόνας. Στο επίπεδο pixel προτείνεται η χρήση φίλτρου Wiener, για να υλοποιήσει στοχαστικές προβλέψεις του αναμενόμενου χρώματος παρασκηνίου. Σε επίπεδο περιοχής, συμπληρώνονται ομογενείς περιοχές του προσκηνίου (αντικειμένου). Σε επίπεδο καρέ, εφόσον τα περισσότερα από τα pixel της τρέχουσας εικόνας επιδεικνύουν απότομη αλλαγή, θεωρείται ότι τα μοντέλα παρασκηνίου βάσει χρώματος pixel δεν είναι πλέον έγκυρα. Σε αυτό το σημείο, είτε ανταλλάσσεται το μοντέλο παρασκηνίου με προηγούμενο, ή αρχικοποιείται εκ νέου.

Μία εναλλακτική προσέγγιση για την αφαίρεση παρασκηνίου, είναι η αναπαράσταση των διαφοροποιήσεων έντασης μίας ακολουθίας pixel σε μία εικόνα, ως διακριτές καταστάσεις που αντιστοιχούν στα συμβάντα του περιβάλλοντος. Για παράδειγμα, για την ανίχνευση αυτοκινήτων σε έναν αυτοκινητόδρομο, τα pixel της εικόνας μπορούν να είναι στην κατάσταση «παρασκήνιο», «προσκήνιο (αυτοκίνητο)» ή «σκιά». Οι Rittscher et al. [73]χρησιμοποιούν HMM (Hidden Markov Models) μοντέλα, για την ταξινόμηση μικρών κορμών μίας εικόνας, ως υπαγόμενους σε αυτές τις κατηγορίες. Στην περίπτωση συμβάντος του περιβάλλοντος, οι Stenger et al. το 2001 [74] χρησιμοποιούν τα μοντέλα HMM, για την ανίχνευση αλλαγών φωτισμού σε δωμάτιο. Το πλεονέκτημα της χρήσης αυτών των μοντέλων, είναι ότι συγκεκριμένα συμβάντα που είναι δύσκολο να μοντελοποιηθούν σωστά με μη εποπτευόμενες προσεγγίσεις μοντελοποίησης παρασκηνίου, μπορούν πλέον να εκπαιδευτούν με χρήση δειγμάτων.

Αντί να μοντελοποιούνται οι παραλλαγές αυτόνομων pixel, οι Oliver et al. το 2000 [55] πρότειναν μία ολιστική προσέγγιση, χρησιμοποιώντας αποσύνθεση ιδιοχώρου. Για k εισερχόμενα καρέ, $I^i: i = 1 \dots k$, μεγέθους $n \times m$, ένας πίνακας παρασκηνίου B μεγέθους $k \times l$ διαμορφώνεται επικαλύπτοντας m σειρές σε κάθε καρέ διαδοχικά, όπου $l = n \times m$, και η ιδιοτιμή της αποσύνθεσης εφαρμόζεται στην συνδιακύμανση του $B, C = B^T B$. Το παρασκήνιο στη συνέχεια, παρίσταται μέσω των περιγραφικότερων n ιδιοδιανυσμάτων, u_i , όπου $i < n < k$, τα οποία περικλείουν όλες τις πιθανές συνθήκες φωτισμού στο οπτικό πεδίο. Συνεπώς, η προσέγγιση αυτή είναι λιγότερο ευαίσθητη σε αυτές. Τα αντικείμενα προσκηνίου, ανιχνεύονται προβάλλοντας την τρέχουσα εικόνα στον ιδιοχώρο και βρίσκοντας τις διαφορές μεταξύ των ανακατασκευασμένων και των πραγματικών εικόνων.

Οι προαναφερθείσες μέθοδοι φέρουν τον περιορισμό της αναγκαιότητας στατικού παρασκηνίου. Το 2003 οι Monnet et al. [75] και οι Zhong και Sclaroff [57] απευθύνθηκαν σε αυτό το πρόβλημα. Και οι δύο μέθοδοι καταφέρνουν να αντιμετωπίσουν χρονικά μεταβαλλόμενα παρασκήνια, όπως κύματα σε νερό, κινούμενα σύννεφα κ.α.. Μοντελοποιούν τις περιοχές της εικόνας ως αυτοπαλίνδρομες διαδικασίες κινούμενου μέσου (ARMA) που παρέχουν έναν τρόπο να εκπαιδευτεί και να προβλεφθεί η κίνηση προτύπων σε μία περιοχή. Μία τέτοια διαδικασία είναι ένα μοντέλο σειράς χρόνου, που αποτελείται από αθροίσματα συστατικών αυτοπαλίνδρομων και κινούμενου μέσου, όπου μία αυτοπαλίνδρομη διαδικασία, μπορεί να περιγραφθεί ως το συζυγισμένο άθροισμα των προηγούμενων τιμών της και ένα σφάλμα λευκού θορύβου.

Συνοψίζοντας, οι περισσότεροι αλγόριθμοι ανίχνευσης για σταθερή κάμερα, χρησιμοποιούν μεθόδους αφαίρεσης παρασκηνίου για να ανιχνεύσουν περιοχές ενδιαφέροντος. Αυτό οφείλεται στο γεγονός, ότι σύγχρονες μέθοδοι αφαίρεσης δύνανται να μοντελοποιήσουν αλλαγές φωτισμού, θόρυβο και περιοδικές κινήσεις περιοχών του παρασκηνίου, και συνεπώς μπορούν με ακρίβεια να ανιχνεύσουν αντικείμενα σε ποικίλες καταστάσεις. Επιπροσθέτως, αυτοί οι αλγόριθμοι είναι υπολογιστικά αποδοτικοί. Στην πράξη, η αφαίρεση παρασκηνίου παρέχει ατελείς περιοχές αντικειμένου, σε πολλές περιπτώσεις. Πιο συγκεκριμένα, τα αντικείμενα μπορεί να χωρίζονται στην παράστασή τους σε πολλαπλές περιοχές. Επίσης μπορεί να έχουν κενά στο εσωτερικό τους, δεδομένου του ότι τα χαρακτηριστικά του μοντελοποιημένου αντικειμένου δεν αποκλείεται να συμπίπτουν με τα χαρακτηριστικά του παρασκηνίου. Ο σημαντικότερος περιορισμός, αυτός της ακίνητης κάμερας, δημιουργεί απαιτήσεις για την υλοποίηση αφαίρεσης παρασκηνίου σε κινούμενη εικόνα [76], οι οποίες σχετίζονται με την υπολογιστική δύναμη. Στόχος είναι να αποκατασταθεί η παραμόρφωση του μοντέλου αλλά και ο επανυπολογισμός του σε μικρά χρονικά παράθυρα, είτε εκ νέου, όπως πρότειναν οι Lucas, Kanade [39][77] με την προσέγγιση οπτικής ροής (optical flow), είτε αντισταθμίζοντας την κίνηση του αισθητήρα δημιουργώντας μωσαϊκά παρασκηνίου όπως πρότειναν οι Rowe και Blake το 1996 [78], και οι Irani και Anandan το 1998 [79]. Ωστόσο, και οι δύο αυτές λύσεις προϋποθέτουν επίπεδες σκηνές και μικρή μόνο κίνηση σε διαδοχικές εικόνες.

Συγκεκριμένα, η μέθοδος των Lucas και Kanade (KLT) είναι από τις πιο διαδεδομένες στην Μηχανική Όραση. Είναι μία διαφορική μέθοδος για την εκτίμηση οπτικής ροής. Θεωρεί ότι η ροή είναι ουσιωδώς σταθερή στην άμεση γειτνίαση του υπό αξιολόγηση pixel, και λύνει τις βασικές εξισώσεις οπτικής ροής για όλα τα pixel της περιοχής, με την χρήση κριτηρίου ελαχίστων τετραγώνων. Συνδυάζοντας πληροφορίες από πολλαπλά γειτονικά pixel, η μέθοδος αυτή μπορεί να επιλύσει την εγγενή ασάφεια της εξίσωσης οπτικής ροής. Επίσης είναι λιγότερο ευαίσθητη σε θόρυβο εικόνας, από ότι οι μέθοδοι σημείων ενδιαφέροντος. Ωστόσο, επειδή είναι αμιγώς τοπική μέθοδος, δεν δύναται να παρέχει πληροφορίες για την ροή στο εσωτερικό ομοιόμορφων περιοχών της εικόνας.

Η μέθοδος Lucas-Kanade θεωρεί ότι η μετατόπιση του περιεχομένου της εικόνας μεταξύ δύο παραπλήσιων χρονικών στιγμών, δηλαδή διαδοχικών καρέ, είναι μικρή και προσεγγιστικά συνεχής εντός του πλαισίου της περιοχής του σημείου p υπό εξέταση. Ως εκ τούτου, η εξίσωση οπτικής ροής, μπορεί να θεωρηθεί ότι ικανοποιεί όλα τα pixel εντός ενός παραθύρου με κέντρο το σημείο p. Τυπικά, το διάνυσμα τοπικής ροής εικόνας, δηλαδή η ταχύτητα (V_x, V_y) πρέπει να ικανοποιεί τις ακόλουθες σχέσεις για κάθε pixel q_i :

$$\left.\begin{array}{l} I_x(q_1)V_x + I_y(q_1)V_y = -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y = -I_t(q_2) \\ \vdots \\ I_x(q_n)V_x + I_y(q_n)V_y = -I_t(q_n) \end{array}\right\} \text{όπου} \begin{cases} I_x = \dfrac{\partial I}{\partial x}(q_i) : \chi\ συντεταγμένη \\ I_y = \dfrac{\partial I}{\partial y}(q_i) : y\ συντεταγμένη \\ I_t = \dfrac{\partial I}{\partial t}(q_i) : t\ χρόνος \end{cases}$$

Ή σε μορφή πίνακα $Au = b$ όπου:

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, u = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, και\ b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

Το σύστημα έχει περισσότερες εξισώσεις από αγνώστους και συνεπώς είναι συχνά υπερορισμένο. Η μέθοδος Lucas-Kanade παρέχει μία λύση συμβιβασμού μέσω της αρχής των ελαχίστων τετραγώνων. Ονομαστικά, επιλύει το σύστημα 2×2 που ονομάζεται και τανυστής δομής ή πίνακας δεύτερης ροπής της εικόνας στο σημείο p.

$$A^T Au = A^T b \text{ ή } V = (A^T A)^{-1} A^T b$$

Όπου A^T είναι ο μετασχηματισμός του πίνακα A. Υπολογίζει δηλαδή τα εξής:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} I_x(q_i)^2 & \sum_{i=1}^{n} I_x(q_i) I_y(q_i) \\ \sum_{i=1}^{n} I_y(q_i) I_x(q_i) & \sum_{i=1}^{n} I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} I_x(q_i) I_t(q_i) \\ -\sum_{i=1}^{n} I_y(q_i) I_t(q_i) \end{bmatrix}$$

1-xvii Οπτική Ροή Lukas-Kanade από τα παραδείγματα του OpenCV .

1.6 Αλγόριθμοι Κατάτμησης

Ο στόχος των αλγορίθμων κατάτμησης είναι να διαχωρίσει την εικόνα σε διαρκώς ίδιες περιοχές. Κάθε αλγόριθμος κατάτμησης απευθύνεται σε δύο προβλήματα, τα κριτήρια για έναν καλό διαχωρισμό και την μέθοδο για να επιτευχθεί αποδοτικός διαχωρισμός. Ακολούθως παρουσιάζονται ορισμένες από τις σημαντικότερες και πιο σύγχρονες τεχνικές κατάτμησης, οι οποίες είναι σχετικές με την αναγνώριση και την παρακολούθηση αντικειμένου.

1.6.1 Ομαδοποίηση μέσης μετατόπισης (Mean-Shift Clustering)

Για το πρόβλημα της κατάτμησης, οι Comaniciu και Meer το 2002 [51] πρότειναν αυτήν την προσέγγιση, για να βρουν ομάδες στο αρθρωτό πεδίο $[l, u, v, x, y]$ χρώματος και χώρου, όπου $[l, u, v]$ παριστούν τις συνιστώσες του χρώματος και $[x, y]$ παριστά την θέση στον χώρο. Δεδομένης μίας εικόνας, ο αλγόριθμος αρχικοποιείται με ένα μεγάλο πλήθος υποθετικών συστάδων κέντρων, τυχαία επιλεγμένων από τα δεδομένα. Τότε, κάθε συστάδα κέντρων μετακινείται στο μέσο των δεδομένων, που κείτονται στο εσωτερικό του πολυδιάστατου ελλειψοειδούς με κέντρο το ίδιο με την συστάδα. Το διάνυσμα που ορίζεται από τα παλαιά και τα νέα κέντρα συστάδας, ονομάζεται διάνυσμα μέσης μετατόπισης. Υπολογίζεται αυξητικά μέχρις ότου τα κέντρα συστάδων δεν μεταβάλλουν την θέση τους. Σημειωτέον ότι κατά τις προσαυξήσεις, κάποιες συστάδες δύνανται να συγχωνευθούν. Η μέθοδος αυτή είναι επίσης επεκτάσιμη σε ποικίλες άλλες εφαρμογές, όπως ανίχνευση ακμών, συστηματοποίηση εικόνας και παρακολούθηση.

Η κατάτμηση βάσει μέσης μετατόπισης απαιτεί λεπτή ρύθμιση σε διάφορες παραμέτρους για να παράγει καλύτερα αποτελέσματα. Για παράδειγμα, η επιλογή του χρώματος και το εύρος ζώνης του χωρικού κορμού, καθώς και το κατώφλι ελάχιστου μεγέθους της περιοχής μπορούν να επηρεάσουν την κατάτμηση που προκύπτει.

1.6.2 Κατάτμηση Εικόνας μέσω Περικομμένων Γραφημάτων (Graph-Cuts)

Η κατάτμηση εικόνας μπορεί επίσης να διαμορφωθεί ως πρόβλημα διαμέρισης γραφημάτων, όπου τα διανύσματα των pixel $V = \{u, v, \dots\}$, ενός γραφήματος εικόνας G διαχωρίζονται σε N ασυνεχή υπογραφήματα περιοχών, κλαδεύοντας τις σταθμισμένες ακμές του γραφήματος.

$$A_i, \bigcup_{i=1}^{N} A_i = V, A_i \cap A_j = \emptyset, i \neq j$$

Η συνολική στάθμη που κλαδεύτηκε μεταξύ δύο υπογραφημάτων ονομάζεται Κοπή. Η στάθμη συνήθως υπολογίζεται μέσω χρώματος, φωτεινότητας, ή ομοιότητας υφής μεταξύ των κόμβων. Οι Wu και Leahy το 1993 [80] χρησιμοποίησαν το κριτήριο ελάχιστης Κοπής, όπου ο στόχος είναι να βρεθούν διαμερίσματα που ελαχιστοποιούν την κοπή αυτή. Στην προσέγγισή τους, οι στάθμες ορίζονται βάσει χρώματος και ομοιότητας. Ένας περιορισμός αυτής της προσέγγισης είναι η τάση υπερβολικής κατάτμησης της εικόνας. Αυτό συμβαίνει επειδή αυξάνεται ο φόρτος μίας κοπής, αναλογικά με τον αριθμό ακμών που τέμνουν δύο διαδοχικά διαμερισμένα τμήματα.

Οι Shi και Malik το 2000 πρότειναν την κανονικοποιημένη κοπή [52] για να αντιμετωπίσουν το πρόβλημα της υπερβολικής κατάτμησης. Στην προσέγγισή τους, η κοπή δεν εξαρτάται μόνο στο σύνολο των σταθμών ακμών της, αλλά και στην αναλογία σταθμών ακμών της συνολικής σύνδεσης σε κάθε διαμέρισμα προς όλους τους κόμβους του γραφήματος. Για κατάτμηση βάσει εικόνας, οι στάθμες μεταξύ των κόμβων ορίζονται από το προϊόν της ομοιότητας χρώματος και της χωρικής εγγύτητας. Εφόσον οι στάθμες μεταξύ κάθε ζεύγους κόμβων υπολογισθούν, κατασκευάζεται ένας πίνακας σταθμών Wκαι ένας διαγώνιος πίνακας D, όπου ισχύει:

$$D_{i,i} = \sum_{j=1}^{N} W_{i,j}$$

Η κατάτμηση πραγματοποιείται αρχικά, υπολογίζοντας τα ιδιοδιανύσματα και τις ιδιοτιμές του γενικευμένου ιδιοσυστήματος $(D-W)y=\lambda Dy$, και στην συνέχεια χρησιμοποιείται το δεύτερο μικρότερο ιδιοδιάνυσμα για να διαιρέσει την εικόνα σε δύο τμήματα. Για κάθε νέο τμήμα, αυτή η διαδικασία επαναλαμβάνεται αναδρομικά μέχρι να επιτευχθεί ένα κατώφλι. Σε κατάτμηση βάσει κανονικοποιημένης κοπής, η λύση στο γενικευμένο ιδιοσύστημα για μεγάλες εικόνες, πιθανώς να είναι δαπανηρή σε υπολογιστική ισχύ και μνήμη. Ωστόσο αυτή η μέθοδος χρήζει λιγότερων χειροκίνητα επιλεγμένων παραμέτρων, εν συγκρίσει με την κατάτμηση μέσης μετατόπισης. Επίσης έχει χρησιμοποιηθεί και στο πλαίσιο παρακολούθησης περιγραμμάτων αντικειμένου.

Η κατάτμηση πραγματοποιείται αρχικά, υπολογίζοντας τα ιδιοδιανύσματα και τις ιδιοτιμές του γενικευμένου ιδιοσυστήματος $(D-W)y=\lambda Dy$, και στην συνέχεια χρησιμοποιείται το δεύτερο μικρότερο ιδιοδιάνυσμα για να διαιρέσει την εικόνα σε δύο τμήματα. Για κάθε νέο τμήμα, αυτή η διαδικασία επαναλαμβάνεται αναδρομικά μέχρι να επιτευχθεί ένα κατώφλι. Σε κατάτμηση βάσει κανονικοποιημένης κοπής, η λύση στο γενικευμένο ιδιοσύστημα για μεγάλες εικόνες, πιθανώς να είναι δαπανηρή σε υπολογιστική ισχύ και μνήμη. Ωστόσο αυτή η μέθοδος χρήζει λιγότερων χειροκίνητα επιλεγμένων παραμέτρων, εν συγκρίσει με την κατάτμηση μέσης μετατόπισης. Επίσης έχει χρησιμοποιηθεί και στο πλαίσιο παρακολούθησης περιγραμμάτων αντικειμένου.

1.6.3 Ενεργά περιγράμματα (Active Contours)

Σε ένα πλαίσιο ενεργών περιγραμμάτων, η κατάτμηση αντικειμένων επιτυγχάνεται εξελίσσοντας ένα μορφολογικά κλειστό περίγραμμα σε όριο του αντικειμένου, με τρόπο που το περίγραμμα εμπερικλείει ερμητικά την περιοχή του αντικειμένου. Η εξέλιξη αυτή διέπεται από μία συνάρτηση ενέργειας E, η οποία καθορίζει την καταλληλότητα του περιγράμματος για την υπόθεση υποψήφιας περιοχής αντικειμένου.

$$E(\Gamma)=\int_0^1 E_{int}(v)+E_{im}(v)+E_{ext}(v)\,ds$$

Όπου s είναι το μήκος τόξου του περιγράμματος Γ, η E_{im} περιλαμβάνει ενέργεια βάσει εμφάνισης, και η E_{ext} καθορίζει επιπρόσθετους περιορισμούς. Η E_{int} περιλαμβάνει περιορισμούς κανονικοποίησης, καθώς και όρους καμπυλότητας, όρους συνέχειας πρώτης τάξης (∇v) ή δεύτερης τάξης ($\nabla^2 v$) για να ανακαλύψει το συντομότερο περίγραμμα. Η ενέργεια βάσει εικόνας μπορεί να υπολογιστεί τοπικά η καθολικά. Τοπικά συνήθως υπολογίζεται, με την μορφή τελεστών κλίσης (gradient) εικόνας, και αξιολογείται περιμετρικά του περιγράμματος. Αντίθετα, τα καθολικά χαρακτηριστικά υπολογίζονται εντός και εκτός της περιοχής αντικειμένου και περιλαμβάνουν το χρώμα και την υφή [81].

Διάφοροι ερευνητές έχουν χρησιμοποιήσει διαφορετικούς όρους στην προαναφερθείσα εξίσωση της ενέργειας. Το 1995 οι Casseles et al. [53] αποκλείουν την E_{ext} και χρησιμοποιούν μόνο την κλίση της εικόνας ως ενέργεια εικόνας.

$$E_{im}=g(|\nabla I|),\text{όπου } g\text{: σιγμοειδής συνάρτηση}$$

Εν συγκρίσει με την κλίση , η συνάρτηση αυτή καθορίζει το περίγραμμα του αντικειμένου ως γεωδαισιακή καμπύλη στον χώρο Riemann. Ωστόσο, οι κλίσεις της εικόνας παρέχουν πολύ τοπικές πληροφορίες και είναι ευαίσθητες σε τοπικά ελάχιστα. Προκειμένου να υπερκερασθεί αυτό το πρόβλημα, κάποιοι ερευνητές εισήγαγαν όρους ενέργειας εικόνας βάσει περιοχής έναντι της κλίσης. Εντούτοις, η χρήση τοπικών όρων στην συνάρτηση ενέργειας, δεν καταλήγει σε καλό εντοπισμό του περιγράμματος του αντικειμένου. Αργότερα, απέκτησαν δημοτικότητα μέθοδοι που συνδυάζουν ενέργεια εικόνας βάσει τοπικότητας και βάσει κλίσης. Συγκεκριμένα οι Παραγιός και Derichce το 2002 [82] πρότειναν την χρήση συνδυασμών ενέργειας κυρτών με κλίση και τοπικότητα. Μοντελοποιούν την εμφάνιση στην E_{region} μέσω μίξης συναρτήσεων Gauss. Η εξέλιξη των περιγραμμάτων, αρχικά πραγματοποιείται καθολικά, και στην συνέχεια τοπικά με διακύμανση από 0 σε 1 σε κάθε προσάυξηση.

$$E_{image}=\lambda E_{boundary}+(1-\lambda)E_{region}$$

Ένα σημαντικό θέμα στις μεθόδους βάσει περιγράμματος είναι η αρχικοποίηση του περιγράμματος. Σε προσεγγίσεις κλίσεως εικόνας, ένα περίγραμμα τυπικά τοποθετείται εξωτερικά της περιοχής αντικειμένου και σμικρύνεται σταδιακά, μέχρις ότου συναντήσει το όριο του αντικειμένου, θυμίζοντας έτσι θηλιά που σφίγγει. Αυτός ο περιορισμός είναι πιο χαλαρός στις μεθόδους βάσει περιοχής, έτσι ώστε το περίγραμμα να μπορεί να αρχικοποιηθεί είτε εντός είτε εκτός του αντικειμένου, προκειμένου το περίγραμμα να επεκταθεί η να συρρικνωθεί αντίστοιχα για να ταιριάξει με το όριο του αντικειμένου. Ωστόσο, αυτές οι προσεγγίσεις απαιτούν προηγούμενη γνώση του παρασκηνίου ή του αντικειμένου. Η αρχικοποίηση μπορεί να πραγματοποιηθεί χωρίς κατασκευή περιοχών εκ των προτέρων, με χρήση πολλαπλών καρέ ή εικόνας αναφοράς.

Πλην της επιλογής της συνάρτησης ενέργειας και την αρχικοποίηση, ένα ακόμα σημαντικό ζήτημα είναι η επιλογή της σωστής παράστασης περιγραμμάτων. Το περίγραμμα αντικειμένου Γ, μπορεί να παρασταθεί είτε άμεσα με σημεία ελέγχου (v), είτε έμμεσα με σύνολα επιπέδων. Στην άμεση παράσταση, η σχέση μεταξύ των σημείων ελέγχου, καθορίζεται με εξισώσεις παρεμβολής καμπύλης (spline equations). Στην παράσταση με σύνολα επιπέδων, το όριο παρίσταται σε ένα χωρικό πλέγμα, το οποίο κωδικοποιεί τις σεσημασμένες αποστάσεις των γραμμών του πλέγματος από το περίγραμμα με αντίθετα πρόσημα για το αντικείμενο και τις περιοχές παρασκηνίου. Το περίγραμμα έμμεσα ορίζεται, ως τα σημεία μηδενικής τομής στο πλέγμα συνόλων επιπέδου. Η εξέλιξη του περιγράμματος αυτού διέπεται από τις μεταβολές στις τιμές του πλέγματος, σύμφωνα με την ενέργεια που υπολογίζεται στην αρχική εξίσωση ενέργειας, και αξιολογείται σε κάθε θέση του πλέγματος. Οι αλλαγές στις τιμές του πλέγματος έχουν ως αποτέλεσμα νέα σημεία μηδενικής τομής και ως εκ τούτου, νέες θέσεις περιγράμματος. Το σημαντικότερο πλεονέκτημα της έμμεσης παράστασης έναντι της άμεσης, είναι η ελαστικότητα στις τοπολογικές αλλαγές διαμέλισης και συγχώνευσης.

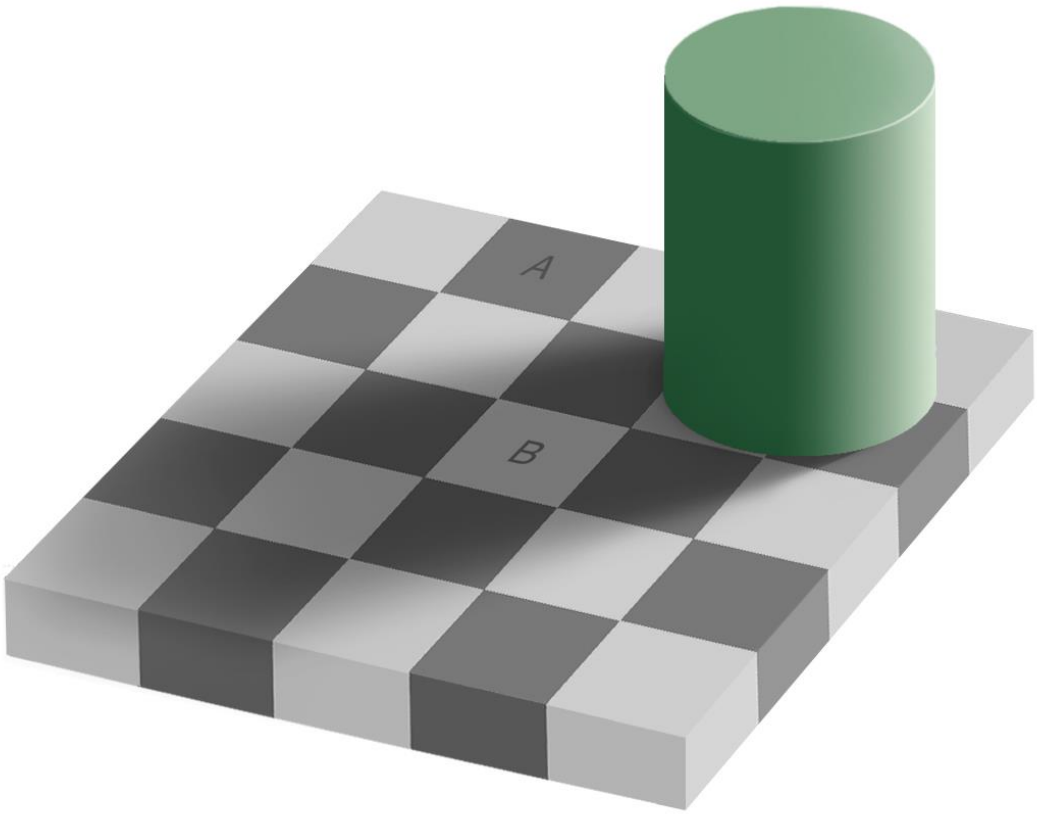
1.7 Εποπτευμένη Μάθηση (Μηχανική Μάθηση)

Η εποπτευόμενη μάθηση είναι το έργο της μηχανικής μάθησης, που αφορά την συναγωγή μίας συνάρτησης από σεσημασμένα δεδομένα εκπαίδευσης. Τα δεδομένα εκπαίδευσης αποτελούνται από σύνολα εκπαιδευτικών παραδειγμάτων. Στην εποπτευόμενη μάθηση, κάθε τέτοιο παράδειγμα, αποτελείται από ένα ζεύγος διανυσμάτων (αντικειμένων) εισόδου και εξόδου, που συχνά αποκαλείται σήμα εποπτείας. Ένας αλγόριθμος εποπτευόμενης μάθησης αναλύει τα εκπαιδευτικά δεδομένα και παράγει μία τεκμαιρόμενη λειτουργία ή συνάρτηση, η οποία μπορεί να χρησιμοποιηθεί για την αντιστοίχιση νέων παραδειγμάτων. Ένα ιδεατό σενάριο επιτρέπει στον αλγόριθμο να προσδιορίσει ορθά τις σημάνσεις κατηγοριοποίησης για αφανείς περιπτώσεις. Αυτό απαιτεί την ικανότητα του αλγορίθμου να γενικεύει τα εκπαιδευτικά δεδομένα σε μοντέλα ικανά να εφαρμοστούν σε αφανείς καταστάσεις με τρόπο λογικό. Η λογική ενός μοντέλου τέτοιου είναι αξιολογήσιμη επιστημονικά με τον ίδιο τρόπο που στην ζωολογία ή την ανθρωπολογία, ένας ευφυής ζωντανός οργανισμός, δύναται να μάθει νέες πληροφορίες σχετικά με το περιβάλλον του. Η ουσία είναι δηλαδή στην ικανότητα των μοντέλων να προβλέψουν μία μελλοντική κατάσταση βάσει μίας παρελθοντικής. Εν γένει η δυνατότητα αυτή είναι και ο βασικός διαχωρισμός των παλαιών παραδοσιακών προσεγγίσεων προς την Τεχνητή Νοημοσύνη από τις σύγχρονες και πολλά υποσχόμενες υποκατηγορίες της Βιο-εμπνευσμένης Πληροφορικής.

Πιο συγκεκριμένα, και ως παράδειγμα που σχετίζεται με το αντικείμενο της εργασίας αυτής μπορούμε να παρατηρήσουμε την λειτουργία της ανθρώπινης όρασης. Ένα κλασικό θέμα είναι η ανίχνευση προσώπου. Στον άνθρωπο πραγματοποιείται ακαριαία και εύκολα στις περισσότερες συνθήκες που επαρκούν οι συνθήκες περιβάλλοντος για την χρήση των αισθητηρίων τους (μάτια). Αυτό συμβαίνει, επειδή οι διαδικασίες της αναγνώρισης προτύπων και της ταξινόμησης, έχουν εξελιχθεί γενετικά βάσει της φυσικής επιλογής και κατά συνέπεια έχουν τελειοποιηθεί στο πέρασμα χιλιάδων γενιών και ποικίλων μεταλλάξεων. Η ικανότητα ενός οργανισμού να αναγνωρίσει πρότυπα κάποιες χιλιάδες χρόνια πριν, παρείχε πλεονέκτημα επιβίωσης σε ένα περιβάλλον, όπου οι απειλές καμουφλάρονται ομοιογενώς με αυτό και συνεπώς οι επόμενες γενιές κληρονομούν ολοένα και καλύτερα μοντέλα πρόβλεψης μέχρι το σημείο που σήμερα μπορούμε βάσει της ευφυΐας μας να κατανοήσουμε σταδιακά την λειτουργία τους και ακόμη και να την μιμηθούμε στα κατασκευάσματά μας. Σε μεγάλη ανάλυση της λειτουργίας της αναγνώρισης προτύπων και αντικειμένων στον άνθρωπο, θα δούμε ότι δεν είναι κάτι που εξολοκλήρου κληρονομείται από το «υλισμικό» μας. Ένας νεογέννητος οργανισμός κατέχει μόνο θεμελιώδεις αντανακλαστικές λειτουργίες σχετικές με όραση. Το υλισμικό του ανθρώπινου εγκεφάλου παρέχει όμως την δυνατότητα να εκπαιδευτεί με τεράστιο όγκο πληροφορίας, κάτι το οποίο πραγματοποιείται κατά την διάρκεια όλης μας της ζωής. Η ικανότητα μας να αναγνωρίζουμε αντικείμενα οπτικά, είναι το αποτέλεσμα χρόνων απορρόφησης δεδομένων εισόδου στα νευρωνικά μας δίκτυα, υλοποιώντας έτσι ένα νευρολογικό «λογισμικό». Το λογισμικό αυτό παρά την περιπλοκότητά του, είναι ικανό να λειτουργήσει ακόμη και περιφερειακά χωρίς να χρειάζεται μεγάλη προσπάθεια από τις ενσυνείδητες διαδικασίες της σκέψης μας. Θα μπορούσε κανείς να παρομοιώσει την λειτουργία αυτή, με την επιτάχυνση της επεξεργασίας δεδομένων που επιτυγχάνεται σε ένα υπολογιστικό σύστημα, όταν εισάγουμε σε αυτό μνήμες Cache και pipelines εντολών. Στην μηχανική των υπολογιστών όμως έχουμε κάποια πλεονεκτήματα το οποία δεν κατέχει ο ανθρώπινος εγκέφαλος:

- Ταχύτητα επεξεργασίας. Ακόμα και αν φαινομενικά το ανθρώπινο μάτι μεταδίδει και αποκωδικοποιεί πληροφορία πολύ γρήγορα, η ικανότητα αυτή ωχριά μπροστά στην ικανότητα ταχύτατης αριθμητικής επεξεργασίας του υπολογιστή, κάτι που θα παρέχει στον υπολογιστή προφανές πλεονέκτημα στην εξαγωγή πληροφορίας, δοθέντος ενός εξίσου αξιόπιστου μοντέλου με αυτό του ανθρώπου.
- Παράλληλη επεξεργασία. Οι σύγχρονοι υπολογιστές μπορούν να πραγματοποιήσουν ολόκληρες διεργασίες ή μικρά παραλληλίσιμα τμήματά τους σε απόλυτη ταυτόχρονη εκτέλεση, κάτι που ο ανθρώπινος εγκέφαλος δεν δύναται να κάνει παρά μόνο με πολύ απλά αντανακλαστικά σε ερεθίσματα. Στον άνθρωπο, η ανίχνευση, ταξινόμηση, αναγνώριση και μνημόνευση προσώπων βασίζεται σε μεγάλο βαθμό στο ψυχολογικό αποτύπωμα βάσει συνυπολογισμού άλλων ερεθισμάτων, αλλά και την αποκωδικοποίηση εκφράσεων και κινητικών ιδιομορφιών κάτι που θα ήταν σχεδόν αδύνατο σε στατικές εικόνες.
- Λογιστικοί περιορισμοί. Ο ανθρώπινος εγκέφαλος έχει συγκεκριμένο μέγεθος, απαιτήσεις ενέργειας, αντοχές αλλά και ημερομηνία φθοράς και λήξης. Ένα υπολογιστικό σύστημα μπορεί να αποτελείται από συστάδες επιμέρους μηχανών, προκειμένου να επιταχύνει την επίλυση πρακτικά οποιουδήποτε προβλήματος και με μεγάλη αποδοτικότητα. Η δυνατότητα συγχρονισμού των επιμερισμένων διαδικασιών ενός προβλήματος σε ένα σύστημα υπολογιστικών μηχανών, είναι σαφώς πλεονεκτική σε σχέση με ένα θεωρητικό αντίστοιχο σύστημα ανθρωπίνων εγκεφάλων, το οποίο στην τρέχουσα κοινωνική μας δομή αντιπροσωπεύεται από μία ομάδα εργασίας. Η επικοινωνία μεγάλου όγκου πληροφορίας μεταξύ ανθρώπων, είναι προφανώς αργότερη από την δικτυακή διασύνδεση υπολογιστών ακόμα και με ξεπερασμένα πρωτόκολλα επικοινωνίας.

Κάποια από τα αντιλαμβανόμενα ως μειονεκτήματα του ανθρώπινου εγκεφάλου, σε κάποιες περιπτώσεις, λειτουργούν ως αφανή εργαλεία των προαναφερθέντων διαδικασιών και θα πρέπει να μελετηθούν και να μοντελοποιηθούν, προκειμένου να καταστούν αξιοποιήσιμα και από τα επιστημονικά πεδία της Πληροφορικής. Για παράδειγμα το φαινόμενο της χρωματικής ομοιογένειας (color constancy) [83], καθιστά τον άνθρωπο ανίκανο να αναγνωρίσει μεγάλες διαφορές σε χρωματικές αποχρώσεις, δεδομένης μίας ψευδούς αντίληψης που οφείλεται σε προσλαμβανόμενη πληροφορία σκίασης και φωτισμού. Σε ένα δεδομένο πρόβλημα όμως, αυτή η ψευδαίσθηση δύναται να επικουρήσει την αναγνώριση αντικειμένου.



1-xviii Οφθαλμαπάτη Χρωματικής ομοιογένειας.

Τα χρώματα των τετραγώνων A και B είναι ίδια.

Πηγή: ("Grey square optical illusion" by Original by Edward H. Adelson, this file by Gustavb - File created by Adrian Pingstone, based on the original created by Edward H. Adelson. Licensed under Copyrighted free)

Η ανίχνευση ενός αντικειμένου μπορεί να πραγματοποιηθεί αυτόματα εκπαιδεύοντας διαφορετικές οπτικές γωνίες του αντικειμένου, από ένα σύνολο παραδειγμάτων μέσω ενός μηχανισμού εποπτευόμενης μάθησης. Η εκπαίδευση των διαφορετικών προοπτικών παραιτείται από την απαίτηση για αποθήκευση πλήρων συνόλων προτύπων. Δεδομένου ενός συνόλου παραδειγμάτων εκπαίδευσης, οι μέθοδοι εποπτευόμενης μάθησης παράγουν μια συνάρτηση που αντιστοιχίζει / χαρτογραφεί εισόδους σε επιθυμητές εξόδους. Μία καθιερωμένη διατύπωση της εποπτευόμενης μάθησης, είναι η ταξινόμηση του προβλήματος ούτως ώστε το εκπαιδευόμενο σύστημα να προσεγγίζει την συμπεριφορά της συνάρτησης, δημιουργώντας μία έξοδο με μορφή είτε συνεχούς τιμής, που ονομάζεται οπισθοδρόμηση, ή με μία σήμανση κλάσεως που ονομάζεται ταξινόμηση. Στο πλαίσιο της ανίχνευσης αντικειμένου, τα εκπαιδευτικά παραδείγματα αποτελούνται από ζεύγη χαρακτηριστικών του αντικειμένου, μαζί με το συσχετιζόμενο αναγνωριστικό κλάσης /κατηγορίας, όπου και οι δύο ποσότητες αυτές προσδιορίζονται χειροκίνητα.

Η επιλογή των χαρακτηριστικών, παίζει σημαντικό ρόλο στην απόδοση της ταξινόμησης, ως εκ τούτου, είναι σημαντικό να χρησιμοποιηθεί ένα σύνολο χαρακτηριστικών τα οποία διακρίνουν την μία κλάση από τις υπόλοιπες. Επιπροσθέτως των χαρακτηριστικών που αναλύθηκαν σε προηγούμενο τμήμα της εργασίας, είναι θεμιτό να χρησιμοποιηθούν άλλα χαρακτηριστικά όπως:

- Εμβαδό αντικειμένου
- Προσανατολισμός του αντικειμένου
- Εμφάνιση αντικειμένου με μορφή συνάρτησης πυκνότητας, όπως για παράδειγμα τα ιστογράμματα.

Άπαξ και επιλεγούν τα χαρακτηριστικά, διάφορες εμφανίσεις του αντικειμένου μπορούν να εκπαιδευτούν, επιλέγοντας μια προσέγγιση της εποπτευόμενης μάθησης. Τέτοιες προσεγγίσεις περιλαμβάνουν μεταξύ άλλων τις ακόλουθες:

- Νευρωνικά δίκτυα [59].
- Προσαρμοστική Ενίσχυση /Adaptive Boosting- Adaboost [60].
- Δένδρα αποφάσεων [84].
- Μηχανές Διανυσμάτων Υποστήριξης /Support Vector Machines –SVM [58]

Οι μέθοδοι εποπτευόμενης εκπαίδευσης απαιτούν συνήθως τεράστια συλλογή δειγμάτων από κάθε κατηγορία αντικειμένου. Επιπρόσθετα, αυτή η συλλογή πρέπει να σημανθεί χειροκίνητα. Μια πιθανή προσέγγιση για να μειωθεί το μέγεθος της σεσημασμένης πληροφορίας, είναι να συνοδευτεί με εποπτευόμενη εκπαίδευση σεσημασμένων και μη δεδομένων [85]. Η βασική ιδέα πίσω από αυτή την μέθοδο αυτή είναι να εκπαιδευτούν δύο ταξινομείς, χρησιμοποιώντας ένα μικρό σύνολο σεσημασμένων δεδομένων όπου τα χρησιμοποιούμενα χαρακτηριστικά κάθε ταξινομέα είναι ανεξάρτητα. Εφόσον επιτευχθεί η εκπαίδευση, κάθε ταξινομέας χρησιμοποιείται για να αναθέσει μη σεσημασμένα δεδομένα στο εκπαιδευτικό σύνολο του άλλου. Έχει δειχθεί, ότι ξεκινώντας από ένα μικρό σύνολο σεσημασμένων δεδομένων με δύο σύνολα στατιστικά ανεξάρτητων χαρακτηριστικών, η δεδομένη μέθοδος δύναται να παρέχει έναν αρκετά ακριβή κανόνα ταξινόμησης. Επιτυγχάνει να μειώσει τον όγκο της χειροκίνητης αλληλεπίδρασης που χρειάζεται κατά την εκπαίδευση, στο πλαίσιο των Adaboost και SVM [86] όπως θα αναλυθεί ακολούθως.

1.7.1 Προσαρμοστική Ενίσχυση /Adaptive Boosting- Adaboost

Η Ενίσχυση είναι μια επαναλαμβανόμενη μέθοδος εύρεσης πολύ ακριβών ταξινομέων, συνδυάζοντας πολλαπλούς βασικούς ταξινομητές, καθένας από τους οποίους πιθανώς να είναι μετρίως ακριβής. Στην φάση της εκπαίδευσης [87] του αλγορίθμου Adaboost, το πρώτο βήμα είναι η κατασκευή μίας αρχικής κατανομής φορτίων επί του εκπαιδευτικού συνόλου. Ο μηχανισμός ενίσχυσης στην συνέχεια επιλέγει έναν βασικό ταξινομέα, ο οποίος αποδίδει το μικρότερο σφάλμα, όπου αυτό ορίζεται ως ανάλογο του φορτίου των εσφαλμένως ταξινομημένων δεδομένων.

$$E(F_{t-1}(x_i)\ \forall\ δείγμα$$

Ακολούθως, τα φορτία που σχετίζονται με τα δεδομένα που ταξινομήθηκαν λάθος από τον βασικό ταξινομέα, αυξάνονται. Ως εκ τούτου ο αλγόριθμος ενθαρύνει την επιλογή άλλων ταξινομέων που θα παρουσιάσουν καλύτερη απόδοση στην επόμενη επανάληψη. Στο πλαίσιο της ανίχνευσης αντικειμένου, οι αδύναμοι ταξινομείς συνήθως είναι απλοί τελεστές, όπως ένα σύνολο από κατώφλια, τα οποία εφαρμόζονται στα χαρακτηριστικά των αντικειμένων που έχουν εξαχθεί από την εικόνα. Ένας τέτοιος ενισχυμένος ταξινομέας έχει την ακόλουθη μορφή.

$$F_T(x) = \sum_{t=1}^{T} f_t(x)$$

Όπου κάθε f_t είναι ένας αδύναμος εκπαιδευόμενος ταξινομέας, που παίρνει ως είσοδο ένα αντικείμενο x και επιστρέφει μία τιμή επαλήθευσης η οποία υποδεικνύει την κατηγορία του αντικειμένου. Το πρόσημο του αδύναμου εκπαιδευόμενου, προσδιορίζει την προβλεπόμενη κλάση αντικειμένου και η απόλυτη τιμή αποτελεί δείκτη βεβαιότητας για την ορθότητα της ταξινόμησης. Ομοίως ο ταξινομέας επιπέδου T, είναι θετικός εφόσον το δείγμα πιθανώς είναι στην θετική κλάση και αρνητικός σε αντίθετη περίπτωση. Κάθε τέτοιος ταξινομέας παράγει μία υπόθεση εξόδου $h(x_i)$, για κάθε δείγμα του εκπαιδευτικού συνόλου. Σε κάθε επανάληψη t επιλέγεται ένας ταξινομέας και ανατείθεται ένας συντελεστής a_t, ούτως ώστε να ελαχιστοποιηθεί το συνολικό σφάλμα εκπαίδευσης του παραγόμενου ενισχυμένου ταξινομέα σταδίου t.

$$E_t = \sum_i E[F_{t-1}(x_i) + a_t h(x_i)]$$

Όπου $F_{t-1}(x)$ είναι ο ενισχυμένος ταξινομέας που έχει αναβαθμιστεί από το προηγούμενο στάδιο της εκπαίδευσης. Επίσης $E(F)$ είναι συνάρτηση σφάλματος και $f_t(x) = a_t h(x)$ είναι ο αδύναμος εκπαιδευόμενος, που βρίσκεται υπό διερεύνηση για προσθήκη στον τελικό ταξινομέα.

Το 2003 οι Viola, Jones et al. χρησιμοποίησαν το πλαίσιο του Adaboost για την ανίχνευση πεζών. Στην προσέγγισή τους χρησιμοποιήθηκαν μηχανές Rosenblatt Perceptron, ως αδύναμοι ταξινομείς οι οποίοι εκπαιδεύτηκαν με χαρακτηριστικά εικόνων, που είχαν εξαχθεί από συνδυασμό χωρικών και χρονικών τελεστών. Οι τελεστές για εξαγωγή χαρακτηριστικών είχαν μορφή απλών παραλληλόγραμμων φίλτρων. Οι δε τελεστές του χρονικού τομέα είχαν μορφή διαφορισμού διαδοχικών εικόνων /καρέ. Αυτή η μέθοδος ελαχιστοποιεί τις εσφαλμένες ανιχνεύσεις, εξαναγκάζοντας την ανίχνευση αντικειμένου μόνο στην περιοχή που συνέβη το γεγονός της κίνησης [88]. Αξίζει να σημειωθεί ότι ο αλγόριθμος Viola-Jones είναι από τους πιο ευρέως χρησιμοποιούμενους για αναγνώριση προσώπου [6], λόγω της απλότητάς του και του μικρού υπολογιστικού φορτίου του, γνωρίσματα που τον κατέστησαν ιδανικό για φορητότητα σε ενσωματωμένα υπολογιστικά συστήματα, με χαρακτηριστικό παράδειγμα τις ψηφιακές φωτογραφικές μηχανές και άλλες κάμερες, που τον χρησιμοποιούν για εντοπισμό προσώπων στην αυτόματη εστίαση και την αφαίρεση κόκκινου ματιού.

1.7.2 Μηχανές Διανυσμάτων Υποστήριξης – SVM

Ως ταξινομέας, τα SVM χρησιμοποιούνται για την δημιουργία συστάδων δεδομένων σε δύο κλάσεις, βρίσκοντας το μέγιστο περιθωριακό υπερεπίπεδο που διαχωρίζει την μία κατηγορία από την άλλη. Το περιθώριο του υπερεπιπέδου, το οποίο μεγιστοποιείται, προσδιορίζεται από την απόσταση μεταξύ των υπερεπιπέδων και των κοντινότερων σημείων δεδομένων. Τα σημεία δεδομένων που κείτονται στο σύνορο του περιθωρίου του υπερεπιπέδου, ονομάζονται Διανύσματα Υποστήριξης (Support Vectors). Στο πλαίσιο της ανίχνευσης αντικειμένου, αυτές οι κλάσεις αντιστοιχούν στις κατηγορίες αντικειμένου (θετικά δείγματα) και τις κατηγορίες εκτός αντικειμένου (αρνητικά δείγματα). Από τα χειροκίνητα καθορισμένα εκπαιδευτικά παραδείγματα που σημάνθηκαν ως αντικείμενα και μη, ο υπολογισμός του υπερεπιπέδου αναμεταξύ άπειρων πιθανών υπερεπιπέδων, πραγματοποιείται μέσω τετραγωνικού προγραμματισμού. Τα διανύσματα που ορίζουν τα υπερεπίπεδα, μπορούν να επιλεγούν ως γραμμικοί συνδυασμοί με παραμέτρους a_i εικόνων των οποίων τα διανύσματα χαρακτηριστικών βρίσκονται στην βάση μοντέλου. Το σημειακό προϊόν χαρτογράφησης χαρακτηριστικού στο πεδίο των SVM, προσδιορίζεται ως συνάρτηση πυρήνα $k(x, y)$. Με την επιλογή των υπερεπιπέδων, τα σημεία x του χώρου χαρακτηριστικών που αντιστοιχίζονται στο υπερεπίπεδο, προσδιορίζονται από την ακόλουθη σχέση.

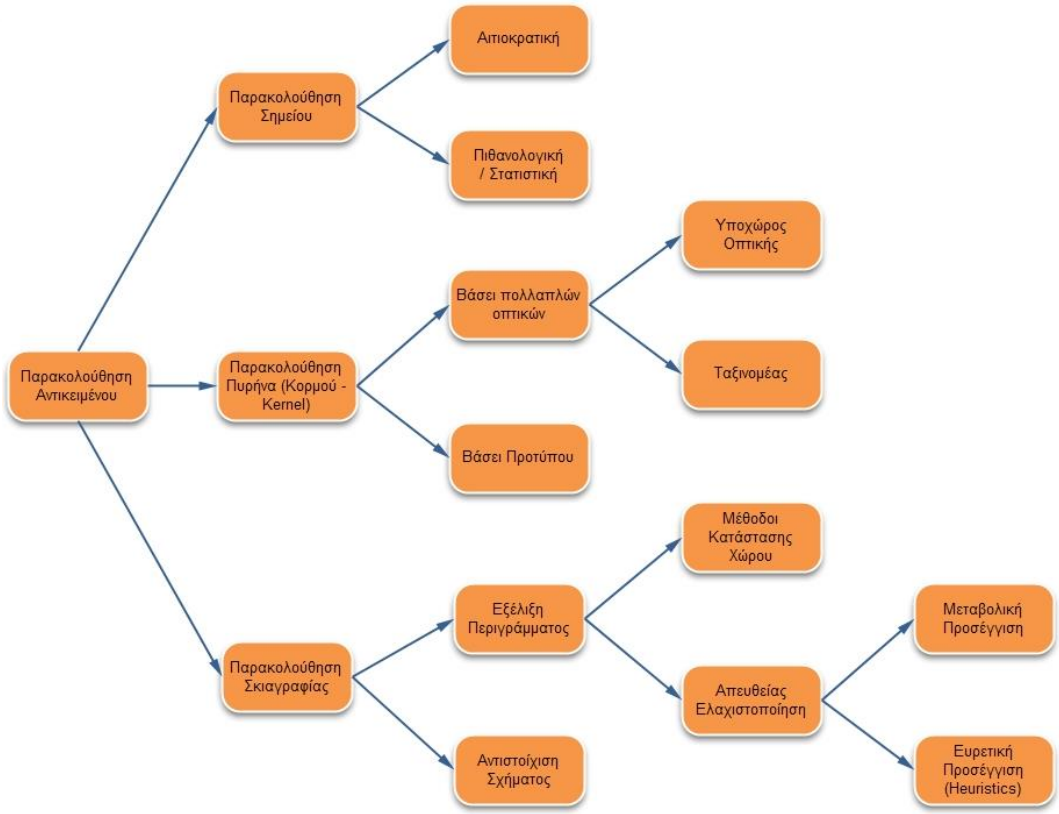
$$\sum_i a_i k(x_i, x) = \sigma\tau\alpha\theta\epsilon\rho\acute{\alpha}$$

Παρόλο που είναι γραμμικός ταξινομέας, τα SVM μπορούν να χρησιμοποιηθούν και ως μη γραμμικός ταξινομέας, χειραγωγώντας την εφαρμογή του κόλπου του πυρήνα στο διάνυσμα χαρακτηριστικών εισόδου. Η εφαρμογή του τεχνάσματος αυτού σε ένα σύνολο που δεν είναι γραμμικά διαχωρίσιμα, μετασχηματίζει τα δεδομένα σε χώρο υψηλότερης διάστασης, όπου τα δεδομένα πιθανώς να είναι γραμμικά διαχωρίσιμα. Οι πυρήνες που χρησιμοποιούνται για το τέχνασμα αυτό είναι πολυωνυμικοί ή ακτινικές συναρτήσεις βάσης, για παράδειγμα κορμός Gauss και Perceptron 2 στιβάδων, ή σιγμοειδής συνάρτηση. Ωστόσο η επιλογή του σωστού πυρήνα για το κάθε δεδομένο πρόβλημα είναι περίπλοκη. Για κάθε πιθανή επιλογή πυρήνα, είθισται η δοκιμή της απόδοσής του για ένα σύνολο παραμέτρων, που πιθανώς να μην αποδώσουν το ίδιο καλά όσο όταν εισαχθούν νέες παρατηρήσεις στο σύνολο δειγμάτων. Το 1998 οι Παπαγεωργίου et al. έκαναν χρήση των SVM για την ανίχνευση πεζών και προσώπων σε εικόνες. Τα χαρακτηριστικά που χρησιμοποιήθηκαν για τον διαχωρισμό μεταξύ των κλάσεων, αποσπάστηκαν με εφαρμογή κυματίων Haar σε σύνολα θετικών και αρνητικών εκπαιδευτικών παραδειγμάτων. Προκειμένου να μειωθεί ο χώρος αναζήτησης, χρησιμοποιήθηκε χρονική πληροφορία, υπολογίζοντας το πεδίο οπτικής ροής στην εικόνα. Συγκεκριμένα, οι ασυνέχειες στο πεδίο οπτικής ροής, αξιοποιείται για να εκκινηθεί η αναζήτηση των πιθανών αντικειμένων, με αποτέλεσμα την μείωση του πλήθους εσφαλμένων θετικών αποτελεσμάτων.

1.8 Παρακολούθηση Αντικειμένου

Ο στόχος ενός ιχνηλάτη, δηλαδή ενός συστήματος παρακολούθησης, είναι να παράγει την τροχιά του στοχευόμενου αντικειμένου με την πάροδο του χρόνου εντοπίζοντας την θέση του σε κάθε νέο καρέ. Ο ιχνηλάτης μπορεί επίσης να παρέχει την πληροφορία προσδιορισμού της περιοχής στην εικόνα, που απασχολείται από το δοθέν αντικείμενο σε κάθε δεδομένη στιγμή. Το καθήκον της ανίχνευσης του αντικειμένου και την εγκαθίδρυση αντιστοιχίας μεταξύ των στιγμιότυπων του αντικειμένου μεταξύ διαφορετικών καρέ, μπορεί να πραγματοποιηθεί είτε ξεχωριστά, είτε ταυτόχρονα με την ιχνηλασία του. Στην πρώτη περίπτωση, πιθανές περιοχές αντικειμένου σε κάθε εικόνα, αποκτώνται από τον αλγόριθμο ανίχνευσης και στην συνέχεια ο ιχνηλάτης αντιστοιχίζει τα αντικείμενα μεταξύ διαφορετικών εικόνων. Στην δεύτερη περίπτωση, η περιοχή του αντικειμένου και η αντιστοίχιση εκτιμώνται από κοινού ανανεώνοντας την θέση του αντικειμένου επαναληπτικά σε σχέση με την θέση που είχε σε προηγούμενες εικόνες. Σε οποιαδήποτε προσέγγιση ιχνηλασίας, τα αντικείμενα αναπαριστώνται με χρήση του σχήματος, ή και των μοντέλων εμφάνισης που έχουν εξηγηθεί σε προηγούμενο τμήμα της εργασίας. Το μοντέλο που επιλέγεται για την παράσταση του σχήματος του αντικειμένου, περιορίζει τους τύπους παραμόρφωσης λόγω κίνησης που δύναται να υποστεί. Αν για παράδειγμα ένα αντικείμενο παρασταθεί ως σημείο, μπορεί να χρησιμοποιηθεί μόνο ένα μεταφραστικό μοντέλο. Στην περίπτωση που χρησιμοποιηθεί παράσταση μέσω γεωμετρικού σχήματος, όπως για παράδειγμα μία έλλειψη, καταλληλότερα είναι μοντέλα όπως ο συσχετισμένος ή ο προβολικός μετασχηματισμός. Αυτές οι αναπαραστάσεις μπορούν να προσεγγίσουν την κίνηση συμπαγών αντικειμένων στην σκηνή. Για ένα μη συμπαγές αντικείμενο, την κίνηση μπορούν να την χαρακτηρίσουν επιτυχώς σκιαγραφίες ή περιγράμματα με παραμετρικά και μη παραμετρικά μοντέλα περιγραφής. Ορισμένες από τις κυριότερες μεθόδους παρακολούθησης ή ιχνηλασίας παρατίθενται ακολούθως.

- Παρακολούθηση Σημείου:
  - Αιτιοκρατικές μέθοδοι
  - Στατιστικές μέθοδοι
- Παρακολούθηση Πυρήνα:
  - Μοντέλα εμφάνισης βάσει Προτύπου και Πυκνότητας
  - Μοντέλα εμφάνισης πολλαπλών οπτικών
- Παρακολούθηση Σκιαγραφίας:
  - Εξέλιξη Περιγράμματος
  - Αντιστοίχιση Σχημάτων



1-xix Γενεαλογία αλγορίθμων Object Recognition

1.8.1 Παρακολούθηση Σημείου

Τα αντικείμενα ανιχνεύονται σε διαδοχικές εικόνες και αναπαριστώνται από σημεία [89]. Η συσχέτιση των σημείων βασίζεται σε προηγούμενη κατάσταση του αντικειμένου, που δύναται να περιέχει περιγραφή της θέσης και της κίνησής του. Αυτή η προσέγγιση απαιτεί έναν εξωτερικό μηχανισμό για την ανίχνευση αντικειμένου σε κάθε καρέ [90].

1.8.2 Παρακολούθηση Πυρήνα / Κορμού

Ο πυρήνας αναφέρεται ως σχήμα και εμφάνιση του αντικειμένου. Για παράδειγμα, ο πυρήνας μπορεί να είναι ένα πρότυπο παραλληλόγραμμο ή ελλειπτικό σχήμα με ένα συσχετιζόμενο ιστόγραμμα. Τα αντικείμενα παρακολουθούνται υπολογίζοντας την κίνηση του πυρήνα σε διαδοχικά καρέ. Αυτή η κίνηση είναι συνήθως σε μορφή παραμετρικού μετασχηματισμού όπως μετάφραση, περιστροφή και συσχέτιση.

1.8.3 Παρακολούθηση Σκιαγραφίας

Η παρακολούθηση πραγματοποιείται, αξιολογώντας την περιοχή του αντικειμένου σε κάθε καρέ. Οι μέθοδοι ιχνηλασίας Σκιαγραφίας χρησιμοποιούν την πληροφορία που είναι κωδικοποιημένη στο εσωτερικό της περιοχής του αντικειμένου. Αυτή μπορεί να έχει μορφή πυκνότητας εμφάνισης και μοντέλων σχήματος, που συνήθως παράγεται από χαρτογράφηση ακμών. Δεδομένων των μοντέλων αντικειμένου, οι σκιαγραφίες παρακολουθούνται, είτε με αντιστοίχιση σχήματος ή με εξέλιξη περιγραμμάτων. Και οι δύο αυτές μέθοδοι δύνανται να χαρακτηριστούν ως μέθοδοι κατάτμησης αντικειμένου, που εφαρμόζεται στο χρονικό πεδίο χρησιμοποιώντας τα προηγούμενα καρέ.

2. ΕΠΙΤΑΧΥΝΣΗ ΕΠΕΞΕΡΓΑΣΙΑΣ

Τα τελευταία χρόνια η ραγδαία ανάπτυξη της τεχνολογίας της μικροηλεκτρονικής έχει αλλάξει κατεύθυνση. Παραδοσιακά, ένα σύστημα γινόταν πιο γρήγορο με την εισαγωγή της νέας γενιάς επεξεργαστών και κυκλωμάτων. Οι επεξεργαστές είχαν μία ανοδική τάση ως προς την απόδοσή τους μέσω ολοένα και αυξανόμενης συχνότητας ρολογιού. Αυτό απαιτούσε συνεχώς μεγαλύτερο βαθμό ολοκλήρωσης και άλλες τεχνικές βελτίωσης της διαδικασίας παρασκευής ημιαγωγών. Αυξάνοντας την ταχύτητα εκτέλεσης, εμμέσως λόγω της αναβάθμισης της συχνότητας ρολογιού, επιτυγχάνονταν βελτιώσεις στην απόδοση του λογισμικού δίχως να χρειάζεται να αναβαθμιστούν ή να τροποποιηθούν σημαντικά οι ίδιες εφαρμογές ή οι βιβλιοθήκες πάνω στις οποίες βασίστηκαν. Όσο αυξανόταν όμως η πυκνότητα των τρανζίστορ, συνολικά τα συστήματα είχαν περισσότερη ροή και απώλεια ρεύματος. Αυτό είχε ως συνέπεια με την σειρά του, την αύξηση της ενεργειακής κατανάλωσης.

Πρόσφατα η βιομηχανία έχει στραφεί στην εξοικονόμηση ενέργειας. Αυτό μεταξύ άλλων εξυπηρετεί τις εκάστοτε πράσινες πολιτικές κρατών, εταιριών και διεθνών οργανισμών, αλλά και τις ανάγκες της αγοράς, δεδομένης της ραγδαία αυξανόμενης ζήτησης για φορητότητα των συσκευών και των εφαρμογών τους. Συνυπολογίζοντας βεβαίως και το τέλμα στις μέχρι σήμερα υπαρκτές τεχνολογίες ολοκλήρωσης, το οποίο προκύπτει από την συσχέτιση μεταξύ μεγαλύτερου βαθμού ολοκλήρωσης και κόστους ανάπτυξής της, αντιλαμβάνεται κανείς τον λόγο που η βιομηχανία και η έρευνα προσανατολίστηκε πλέον σε εναλλακτικούς τρόπους επίτευξης του στόχου της επιτάχυνσης επεξεργασίας. Η επιτάχυνση πλέον, στο επίπεδο του υλικού επιτυγχάνεται με δύο προσεγγίσεις:

- Εξειδίκευση.
- Παραλληλοποίηση.

2.1 Εξειδίκευση

Η εξειδίκευση αφορά την κατασκευή επιταχυντών υλικού με συγκεκριμένο πεδίο εφαρμογής, οι οποίοι μπορούν να πραγματοποιήσουν μία ειδική κατηγορία λειτουργιών αποτελεσματικά. Αυτή η τάση παραδοσιακά, είναι πιο επικερδής και συνεπώς βιώσιμη στην βιομηχανία, όταν ο στόχος είναι η κατασκευή μεγάλου αριθμού μονάδων που θα έχουν πολύ περιορισμένη λειτουργικότητα, αναβαθμισιμότητα και πεδίο εφαρμογής (ASIC). Φυσικά, αυτό σημαίνει ότι ο προβλεπόμενος χρόνος ζωής τους, είναι αυστηρά περιορισμένος από την εξέλιξη των προϊόντων και του ανταγωνισμού, αλλά και την μελλοντική συνάφεια της συγκεκριμένης εφαρμογής με τις ανάγκες της αγοράς. Ο συμβιβασμός αυτής της προσέγγισης με προβλήματα μικρότερου πεδίου εφαρμογής και την δυνατότητα αναβάθμισης, έρχεται με την χρήση των FPGA και παλαιότερα άλλων συστημάτων προγραμματιζόμενης λογικής όπως τα CPLD και PLA. Ομολογουμένως, ειδικά στην έρευνα, αλλά και την μικρής κλίμακας βιομηχανία / βιοτεχνία, τα συστήματα αυτά παρέχουν τεράστιες δυνατότητες ευελιξίας και σε μεγάλο βαθμό συνδυάζονται με προσεγγίσεις παραλληλοποίησης, συχνά εξαιρετικά αποδοτικά. Όμως αλλά ένα μειονέκτημά τους που τα καθιστά δύσχρηστα, είναι η περιπλοκότητα τόσο της υποστήριξής τους από περιφερειακό υλικό, όσο και των εργαλείων ανάπτυξης του λογισμικού τους. Σε συνδυασμό μάλιστα με το αυξημένο κόστος της μονάδας τους, σε σύγκριση με την ευρεία κατανάλωση των εξειδικευμένων και συμβατικών συστημάτων τα καθιστούν απαγορευτικά για πολλές εφαρμογές, στις οποίες το χαμηλό κόστος είναι σημαντικός παράγοντας.

2.2 Παραλληλοποίηση

Στο πρόσφατο παρελθόν, οι κυρίαρχες προσεγγίσεις προς την κατεύθυνση της παραλληλίας ήταν σε δύο τομείς.

- Στον τομέα της αρχιτεκτονικής υπολογιστών [91], υιοθετήθηκαν τεχνικές όπως η σωλήνωση εντολών (pipelining) και πολυνημάτωσης (multithreading), που ανήκουν στον τομέα της Παραλληλίας Επιπέδου Εντολής (ILP) και Παραλληλίας Επιπέδου Νήματος (TLP) αντίστοιχα. Με αυτές κερδίζεται χρόνος σε επίπεδο εκτέλεσης, έχοντας διενεργήσει τα βήματα της ανάγνωσης επόμενων εντολών και της πρώιμης μεταφοράς δεδομένων της επόμενης προς εκτέλεση εντολής, ενώ ακόμα ο επεξεργαστής εκτελεί την τρέχουσα. Φυσικά, αυτό προϋποθέτει ότι οι επόμενες εντολές, δεν θα φέρουν εξαρτήσεις δεδομένων ή θέσεων μνήμης με την τρέχουσα. Ωστόσο, σε καμία περίπτωση αυτή η μέθοδος δεν συνιστά πραγματική παράλληλη εκτέλεση δύο ή περισσότερων εντολών ταυτόχρονα.

Μία ακόμη τεχνική από τον τομέα της Παραλληλίας Επιπέδου Μνήμης (MLP), υπήρξε αυτή των Κρυφών Μνημών (Cache). Προκειμένου να μειωθεί ο χρόνος μεταφοράς των δεδομένων από την κύρια μνήμη στους καταχωρητές του επεξεργαστή, ο οποίος ομολογουμένως είναι μεγάλος σε σύγκριση με τους χρόνους εκτέλεσης που επιτυγχάνονται στους σύγχρονους επεξεργαστές, υιοθετήθηκε η διαμεσολάβηση ενδιάμεσου συστήματος μνήμης και διαχείρισής της εντός του ολοκληρωμένου κυκλώματος του επεξεργαστή, προκειμένου να μπορούν ελεγχόμενα και προστατευμένα από ηλεκτρομαγνητικές παρεμβολές, να μεταφερθούν τα δεδομένα με ταχύτητες που αξιοποιούν αποδοτικότερα τις συχνότητες ρολογιού του επεξεργαστή. Η μνήμη αυτή, που συνήθως αφορά τόσο εντολές όσο και δεδομένα, φέρει τις τελευταίες ή πιο συχνά χρησιμοποιούμενες εντολές ή δεδομένα, προβλέποντας έτσι και προλαμβάνοντας την αποδοτικότερη επανάληψή τους. Δεδομένου του ότι βρίσκεται εντός του ολοκληρωμένου κυκλώματος του επεξεργαστή και ότι το μέγεθός της περιορίζεται από το γεγονός ότι ο χρόνος προσπέλασής της αυξάνεται, όσο μεγαλύτερη είναι η χωρητικότητά της, στις περιπτώσεις αστοχίας καταλήγει να επιβαρύνει την εκτέλεση του προγράμματος. Αυτό αντιμετωπίστηκε σχεδιαστικά, με την προσθήκη περισσότερων και ευρύτερων επιπέδων μνήμης, τα οποία πυραμιδικά δρομολογούν τα δεδομένα και τις εντολές προς τα ανώτερά τους επίπεδα, αυξάνοντας έτσι το συνολικότερο ποσοστό ευστοχίας τους (Cache hit), αλλά δυσχεραίνοντας σημαντικά τον χρόνο εκτέλεσης σε περίπτωση αστοχίας (Cache miss).

- Στον τομέα του λογισμικού οι μεταγλωττιστές και τα υπόλοιπα εργαλεία ανάπτυξης κώδικα, υιοθέτησαν μεθόδους βελτιστοποίησης, με διάφορα επίπεδα παρεμβατικότητας στον πρωτότυπα συγγεγραμμένο κώδικα (compiler optimization levels). Στις πιο κοινές περιπτώσεις, ο μεταγλωττιστής, πέρα από τις διαδικασίες αναδίπλωσης και αναδιάταξης του κώδικα, επιβαρύνεται με την κατασκευή δένδρων εξαρτήσεως, προκειμένου να διαπιστώσει και να δρομολογήσει ανάλογα τα αντίστοιχα τμήματα κώδικα που δύνανται να εκτελεστούν παράλληλα ή με τρόπο που θα αξιοποιείται καλύτερα το διαθέσιμο υλισμικό όπως προαναφέρθηκε πιο πάνω.

Αυτές οι προσεγγίσεις φάνηκαν ιδιαίτερα αποδοτικές εν γένει, και γι αυτό εξακολουθούν να συμπεριλαμβάνονται στις υλοποιήσεις των νέων συστημάτων, συνυπάρχοντας με τις πιο νέες και εξελιγμένες μεθόδους παραλληλίας. Βέβαια, η υλοποίησή τους είναι συχνά περισσότερο περίπλοκη, ογκώδης και ενεργοβόρα από τις δυνατότητες των νέων εργαλείων, που προσανατολίζονται στην απλότητα της πολλαπλά χρησιμοποιούμενης μονάδας και τις απαιτήσεις ενός περιβάλλοντος πολυεπεξεργασίας.

Η σύγχρονη γενική προσέγγιση, αφορά την χρήση πολλαπλών ομοίων επεξεργαστικών μονάδων για την υλοποίηση ενός έργου, αντί για την δημιουργία μίας μεμονωμένης πολύ ισχυρότερης μονάδας. Η συνδυασμένη χρήση των δύο προσεγγίσεων, δηλαδή η χρήση ισχυρών επεξεργαστών ταυτόχρονα με διάφορους επιταχυντές υλικού, ονομάζεται «Ετερογενής Παράλληλη Επεξεργασία». Η παράλληλη επεξεργασία είναι μία εναλλακτική λύση προβλημάτων, τα οποία χρειάζονται μεγάλους χρόνους επεξεργασίας, ή διαχειρίζονται μεγάλους όγκους πληροφορίας. Με την προσέγγιση αυτή, ένα πρόγραμμα είναι σε θέση να δημιουργήσει πολλαπλές διεργασίες, που δουλεύουν μαζί για την λύση του προβλήματος. Η κύρια ιδέα είναι η διαίρεση του προγράμματος σε απλές επιμέρους εργασίες και η ταυτόχρονη επίλυσή τους, με τρόπο τέτοιο που ο συνολικός χρόνος μπορεί να διαμοιραστεί μεταξύ του συνόλου των εργασιών.

Σε ένα υπολογιστικό σύστημα με πολλαπλούς επεξεργαστές, ένα πρόγραμμα μπορεί να χωριστεί σε επιμέρους διεργασίες ή νήματα (threads). Κάθε διεργασία είναι δυνατόν να διεκπεραιώνεται από διαφορετικό επεξεργαστή, σε παραλληλία με τις άλλες, βελτιώνοντας την ταχύτητα εκτέλεσης του προγράμματος. Αυτός ο τρόπος εκτέλεσης καλείται «Παραλληλία Νημάτων» (Thread Level Parallelism – TLP). Σε ένα σύστημα με έναν επεξεργαστή, η επεξεργασία πολλαπλών διεργασιών πραγματοποιείται με την διενέργεια διακοπής σε προκαθορισμένα χρονικά διαστήματα (Time Sliced Multithreading). Σε ένα σύστημα συμμετρικής πολυεπεξεργασίας (SMP - Symmetric Multi - proccessing) με πολλαπλούς επεξεργαστές, οι διεργασίες τοποθετούνται στην μνήμη και οι διευθύνσεις αρχής στον πίνακα με τα διανύσματα διακοπής. Με την δημιουργία διακοπής, το διάνυσμα διακοπής μεταβιβάζεται στον επεξεργαστή, που θα εκτελέσει την συγκεκριμένη διεργασία.

Σε έναν σύγχρονο επεξεργαστή, ο οποίος διαθέτει πολλές μονάδες εκτέλεσης εντολών, μπορούν να εκτελεστούν εντολές παράλληλα, αλλά πάντα κάποια από τις μονάδες εκτέλεσης παραμένει άεργη. Η τεχνική της Υπερνημάτωσης (Hyper Threading) προσπαθεί να εκμεταλλεύεται ταυτόχρονα όλες τις μονάδες επεξεργασίας, εκτελώντας δύο διαφορετικές εργασίες. Η ανεξάρτητη και ταυτόχρονη εκτέλεση δύο διαφορετικών προγραμμάτων, προϋποθέτει δύο ανεξάρτητες ομάδες καταχωρητών. Η μονάδα ανάκλησης εντολών παρέχει στον πυρήνα εντολές, πότε από το ένα και πότε από το άλλο πρόγραμμα. Φυσικά όλη αυτή η διαδικασία προϋποθέτει την συμβατότητα και υποστήριξη της υπερνημάτωσης από το λειτουργικό σύστημα. Ο επεξεργαστής που διαθέτει τέτοια δυνατότητα, ονομάζεται «φυσικός επεξεργαστής», ενώ οι δύο εσωτερικές μονάδες που πραγματοποιούν την παραλληλία, ονομάζονται «λογικοί επεξεργαστές». Αρχικά κάθε φυσικός επεξεργαστής διέθετε δύο λογικούς επεξεργαστές. Το υπολογιστικό σύστημα δύναται να είναι εφοδιασμένο με περισσότερους από έναν φυσικούς επεξεργαστές, σε ένα περιβάλλον πολυεπεξεργασίας, οι οποίοι εκμεταλλεύονται τον ίδιο διάδρομο FSB. Κάθε ομάδα φυσικών επεξεργαστών, καλείται «συστοιχία» (Cluster) και ένα σύστημα πολυεπεξεργασίας μπορεί να διαθέτει περισσότερες από μία συστοιχίες. Κάθε λογικός επεξεργαστής είτε διαθέτει δικούς του πόρους, είτε μοιράζεται κοινούς πόρους, ή εκμεταλλεύεται κοινούς πόρους που διαμοιράζονται ισομερώς σε κάθε λογικό επεξεργαστή.

Η παράλληλη επεξεργασία δεν δύναται να εφαρμοστεί σε όλα τα προβλήματα, δηλαδή δεν είναι όλα τα προβλήματα ικανά να κωδικοποιηθούν σε παράλληλη μορφή. Ένα παράλληλο πρόγραμμα, πρέπει να έχει ορισμένα γνωρίσματα για την ορθή και αποδοτική του λειτουργία. Σε διαφορετική περίπτωση είναι πιθανόν ότι ο χρόνος εκτέλεσης ή η λειτουργία, δεν θα έχει την αναμενόμενη επίδοση. Τα γνωρίσματα αυτά περιλαμβάνουν τα ακόλουθα:

- Διακριτότητα (Granularity). Αυτή αφορά το πλήθος των βασικών μονάδων και ταξινομείται ως εξής:
  - Χονδρού Κόκκου παραλληλίας, δηλαδή λίγες εργασίες και μεγαλύτερος υπολογιστικός φόρτος ανά εργασία
  - Λεπτού Κόκκου παραλληλίας, δηλαδή μεγάλο πλήθος μικρών εξαρτημάτων και μικρότερος υπολογιστικός φόρτος ανά εργασία.
- Τύπος παράλληλης επεξεργασίας:
  - Άμεση: ο αλγόριθμος περιέχει οδηγίες που καθορίζουν ποιες διεργασίες μεταγλωττίζονται και εκτελούνται με τρόπο παράλληλο.
  - Έμμεση: Ο μεταγλωττιστής φέρει το έργο της παρεμβολής των απαραίτητων εκείνων οδηγιών για να εκτελεστεί το πρόγραμμα παράλληλα.
- Συγχρονισμός. Ο τρόπος με τον οποίο αποφεύγεται η επικάλυψη δύο ή περισσότερων διεργασιών.
- Καθυστέρηση. Αυτή είναι ο χρόνος που παρέρχεται για την μετάβαση της πληροφορίας από την αίτηση στην παραλαβή.
- Επεκτασιμότητα. Ορίζεται ως η ικανότητα ενός αλγορίθμου να διατηρήσει την αποδοτικότητα του αυξάνοντας το πλήθος των επεξεργαστών ενώ το μέγεθος του προβλήματος διατηρεί την ίδια αναλογία
- Επιτάχυνση και αποδοτικότητα. Αυτά ορίζονται ως μετρικά μεγέθη για να αξιολογήσουν την ποιότητα μίας παράλληλης υλοποίησης.

Το μέγεθος που περιγράφει την αποδοτικότητα μίας παράλληλης υλοποίησης ονομάζεται «παράγοντας επιτάχυνσης» [92]. Δίνει την αύξηση στην ταχύτητα υπολογισμού και ορίζεται ως:

$$S(p) = \frac{t_s}{t_p}$$

Όπου t_s είναι ο χρόνος εκτέλεσης χρησιμοποιώντας έναν μόνο επεξεργαστή (ο καλύτερος ακολουθιακός αλγόριθμος) και t_p είναι ο χρόνος εκτέλεσης σε ένα πολυεπεξεργαστή ή με p επεξεργαστές. Για έναν μονό επεξεργαστή, χρησιμοποιείται ο καλύτερος ακολουθιακός αλγόριθμος, ενώ στην παράλληλη υλοποίηση ο συνήθως σημειώνονται σημαντικές διαφορές.

Ένα πλήθος p επεξεργαστών δύναται να πετύχει μέγιστη επιτάχυνση η οποία είναι ως επί το πλείστον p γραμμική. Σε ορισμένες περιπτώσεις, είναι δυνατόν να επιτευχθεί επιτάχυνση μεγαλύτερη από p, που λέγεται υπεργραμμική (superlinear), και συνήθως είναι αποτέλεσμα της επιπλέον μνήμης του πολυεπεξεργαστικού συστήματος, καθώς στην περίπτωση χρήσης κρυφών μνημών, όπου γίνεται πιο αποδοτική χρήση τους στα πολυεπεξεργαστικά συστήματα.

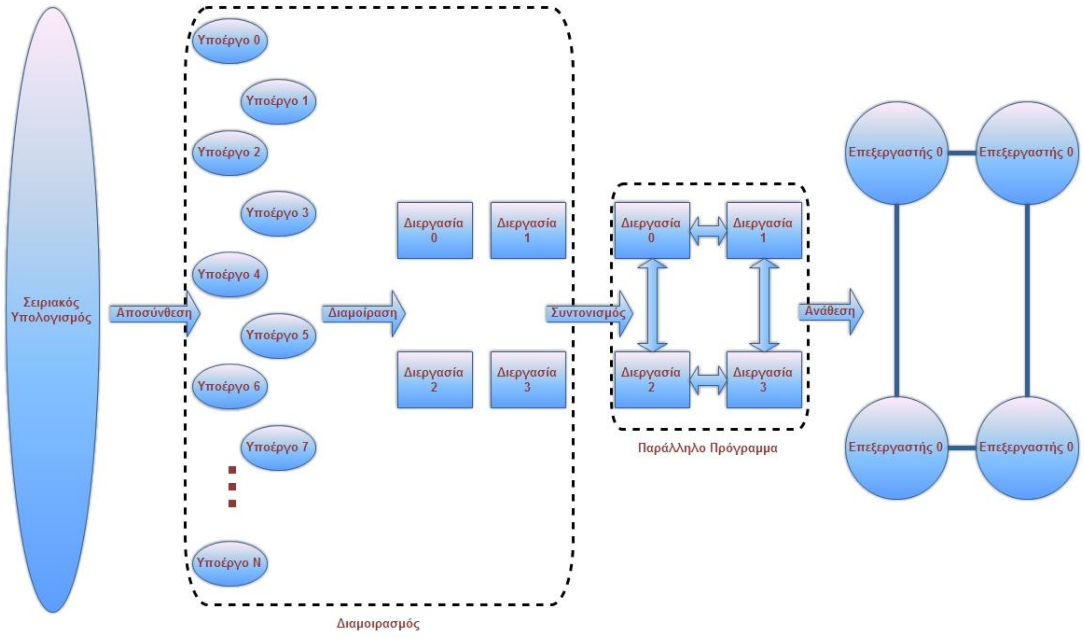
Οι περισσότεροι αλγόριθμοι υπολογισμού περιέχουν τμήματα που είναι αποκλειστικά ακολουθιακά και άλλα τα οποία δύνανται να διαμοιραστούν ως επιμέρους υποέργα. Ορίζοντας t_s τον χρόνο εκτέλεσης ολόκληρου του αλγορίθμου ακολουθιακά, ft_s τον χρόνο εκτέλεσης του ακολουθιακού τμήματος και $(1-f)t_s$ τον χρόνο εκτέλεσης του παραλληλίσιμου τμήματος, σε ένα πολυεπεξεργαστικό σύστημα με p επεξεργαστές, το ακολουθιακό επεξεργαστικό φορτίο διατηρεί χρόνο εκτέλεσης ft_s , ενώ το παράλληλο επεξεργαστικό φορτίο εκτελείται σε χρόνο $\frac{(1-f)t_s}{p}$. Η επιτάχυνση αυτού του υπολογισμού εκφράζεται ως ακολούθως.

$$S(p) = \frac{t_s}{ft_s + \frac{(1-f)t_s}{p}} = \frac{p}{1+(p-1)f}$$

Αυτή η σχέση ονομάζεται «*Νόμος του Amdahl*» και μας δείχνει μεταξύ άλλων ότι ακόμα και με άπειρο πλήθος επεξεργαστών, η μέγιστη επιτάχυνση είναι το πολύ $\frac{1}{f}$.

Γενικά, ορίζουμε ως παράλληλο πρόγραμμα αυτό που αποτελείται από δύο ή περισσότερες διεργασίες, που συνεργάζονται ή και συγχρονίζονται για την επίλυση του ίδιου προβλήματος. Κάθε μία διεργασία αποτελεί από μόνη της ένα ακολουθιακό πρόγραμμα. Οι διάφορες διεργασίες ενός παράλληλου προγράμματος, πετυχαίνουν την απαραίτητη επικοινωνία μεταξύ τους χρησιμοποιώντας κοινές ή διαμοιραζόμενες μεταβλητές (shared variables), ή μέσω αποστολής και παραλαβής μηνυμάτων (message passing).

Η δημιουργία ενός παράλληλου προγράμματος, υλοποιείται σε στάδια. Αρχικά ο ακολουθιακός υπολογισμός επιμερίζεται σε υποέργα. Ακολούθως τα υποέργα διανέμονται σε ένα σύνολο διεργασιών. Συχνά ένα υποέργο παρουσιάζει εξάρτηση δεδομένων με κάποιο άλλο, κάτι που καθιστά απαραίτητο τον συντονισμό και την επικοινωνία μεταξύ των διαφορετικών παράλληλων διεργασιών. Τελικά οι διεργασίες αντιτίθενται σε επεξεργαστές του πολυεπεξεργαστικού συστήματος και οι διαφορετικοί επεξεργαστές εκτελούν τις αντίστοιχες διεργασίες. Ακολουθεί σχηματική παράσταση της δημιουργίας παράλληλου προγράμματος.



2-i Παράλληλοποίηση Έργου και Ανάθεση σε Επεξεργαστές.

Η παράλληλης εκτέλεσης των εργασιών, έχει ως απώτερο σκοπό την περάτωση όλων των υποέργων στον συντομότερο δυνατό χρόνο. Το πρόβλημα της παραλληλίας από την οπτική της θεωρίας υπολογιστικής πολυπλοκότητας, είναι τύπου NP-complete (non-deterministic polynomial), δηλαδή «μη αιτιοκρατικού πολυωνυμικού χρόνου».

Ένα πλήθος επεξεργαστών οι οποίοι έχουν πρόσβαση στην ίδια κοινή μνήμη συναποτελούν έναν πολυεπεξεργαστή. Οι πολυεπεξεργαστές χωρίζονται σε δύο κατηγορίες:

- Πολυεπεξεργαστές Ομοιόμορφης Προσπέλασης Μνήμης (Uniform Memory Access –UMA) ονομάζονται αυτοί οι οποίοι χρησιμοποιούν κεντρική κοινή μνήμη. Η μνήμη αυτή προσπελάζεται μέσω κεντρικού μηχανισμού και οι επεξεργαστές πετυχαίνουν την μεταξύ τους επικοινωνία μέσω συμβατικών εντολών πρόσβασης μνήμης. Η εικονικής μνήμη των διεργασιών εν μέρει αντιστοιχεί σε χώρο της κοινής μνήμης.
- Πολυεπεξεργαστές Μη Ομοιόμορφης Προσπέλασης Μνήμης (Non-Uniform Memory Access – NUMA) ονομάζονται αυτοί των οποίων η κοινή μνήμη είναι κατανεμημένη στους επιμέρους επεξεργαστές. Δίδεται δηλαδή η ψευδαίσθηση ότι υπάρχει κοινή διαμοιραζόμενη μνήμη, επειδή ο κάθε επεξεργαστής διαθέτει την δική του μνήμη. Ωστόσο ο χρόνος για την ολοκλήρωση μίας εγγραφής ή ανάγνωσης στην μνήμη αυτή διαφέρει σημαντικά, αναλόγως του αν η λειτουργία αυτή βρίσκεται στην τοπική μνήμη ή σε μνήμη άλλου επεξεργαστή.

Ο προγραμματισμός σε συστήματα πολυεπεξεργαστών διαμοιραζόμενης μνήμης, γίνεται με τους ακόλουθους τρόπους:

- Νήματα (Threads).

  Σε αυτήν την προσέγγιση, το πρόγραμμα χωρίζεται από τον προγραμματιστή σε επιμέρους παράλληλες ακολουθίες, που ονομάζονται «νήματα». Γίνεται χρήση σφαιρικών-καθολικών (global) μεταβλητών που δηλώνονται έξω από κάθε νήμα αλλά προσπελάζονται από όλα τα νήματα. Χαρακτηριστικά παραδείγματα αυτής της κατηγορίας είναι τα νήματα POSIX (pthreads), τα νήματα Java, και τα νήματα της γλώσσας CUDA (NVIDIA).
- Οδηγίες Προεπεξεργασίας.

  Οι κλασικές γλώσσες προγραμματισμού καθίστανται συμβατές με την χρήση παράλληλων διεργασιών και νημάτων, με την εισαγωγή εντολών προεπεξεργασίας (directives) προς τον μεταγλωττιστή, οι οποίες μεταξύ άλλων, διεκπεραιώνουν την δήλωση κοινών μεταβλητών και προσδιορίζουν τον παραλληλισμό. Αυτό επιτυγχάνεται συνήθως με χρήση εξωτερικής βιβλιοθήκης όπως είναι η OpenMP.
- Αυτόματη υλοποίηση Μεταγλωττιστή.

  Σε αυτήν την προσέγγιση, ο προγραμματιστής συγγράφει το πρόγραμμα σε κάποια κλασική γλώσσα προγραμματισμού και εν συνεχεία, καλείται ο μεταγλωττιστής να μετατρέψει το πρόγραμμα σε παράλληλο, κάτι που με τις παρούσες μεθόδους και τεχνολογίες, δεν επιτυγχάνεται συχνά, εύκολα και αποδοτικά.

Ορισμένα προβλήματα δεν παρουσιάζουν εξαρτήσεις δεδομένων, επικοινωνίας, ή χρονικές μεταξύ επιμέρους υποέργων τους. Σε αυτή την περίπτωση, μπορούμε να πετύχουμε ιδανική επιτάχυνση και τα προβλήματα αυτά ονομάζονται «Επαίσχυντα Παραλληλήσιμα» (Embarrassingly Parallelizable) ή «Ευχάριστα Παραλληλήσιμα» (Pleasingly Parallelizable). Η προσπάθεια που απαιτείται για να διαχωριστεί το πρόβλημα σε παράλληλα έργα είναι ελάχιστη. Η επικοινωνία που απαιτείται μεταξύ των διεργασιών σε τέτοια προβλήματα είναι αισθητά μικρή ή ανύπαρκτη. Συχνά σε τέτοιου είδους υπολογισμούς υιοθετείται η τεχνική Κύριου-Υπόδουλου (Master-Slave). Αρχικά η κύρια διεργασία μοιράζει τα δεδομένα εισόδου, σε ένα πλήθος από δευτερεύουσες και στην συνέχεια, η δευτερεύουσα υλοποιεί ένα τμήμα του υπολογισμού χωρίς να επικοινωνεί με τις υπόλοιπες. Μετά την περάτωση του υπολογισμού που της έχει ανατεθεί, κάθε δευτερεύουσα διεργασία στέλνει το τοπικό της αποτέλεσμα στην κύρια διεργασία, η οποία με την σειρά της συνυπολογίζει το σύνολο των αποτελεσμάτων όλων των δευτερευουσών διεργασιών και επιστρέφει το τελικό αποτέλεσμα.

Υπάρχει μια πληθώρα προβλημάτων που έχουν επισημανθεί ως ευκόλως παραλληλήσιμα. Μεταξύ αυτών βρίσκονται και τα ακόλουθα:

- Σύνολο Mandelbrot.
- Μέθοδοι Μόντε Κάρλο.
- Κατανεμημένες σχεσιακές αναζητήσεις σε Βάση Δεδομένων με επεξεργασία κατανεμημένων συνόλων.
- Διανομή στατικών αρχείων σε πολλαπλούς χρήστες ενός διακομιστή.

- Ερμηνεία πολυμέσων (rendering).
- Αναζητήσεις Ωμής Δύναμης στην κρυπτογραφία, με χαρακτηριστικό παράδειγμα την κατανεμημένη χρήση στην παραγωγή ψηφιακών κρυπτονομισμάτων (bitcoin).
- Γενετικοί Αλγόριθμοι και άλλες ευρετικές μέθοδοι εξελικτικού υπολογισμού.
- Εξομοίωση και ανακατασκευή συμβάντων στην Σωματιδιακή Φυσική.
- Επεξεργασία Εικόνας και Μηχανική Όραση, που αποτελεί και το βασικό πεδίο ενδιαφέροντος της παρούσας εργασίας.

2.2.1 Προγραμματισμός Γενικού Σκοπού Επεξεργαστών Γραφικών (General Purpose Graphics Proccessing Unit – GP GPU Computing)

Ο επεξεργαστής γραφικών είναι ένα ενσωματωμένο υπολογιστικό σύστημα – κύκλωμα, του οποίου ο γενικός σκοπός είναι, να χειρίζεται και να επιταχύνει τις διαδικασίες δημιουργίας εικόνων, που προβάλλονται στην οθόνη του υπολογιστή. Χρησιμοποιείται πλέον σε πληθώρα ψηφιακών συσκευών πέρα από τους συμβατικούς υπολογιστές, όπως: κινητά τηλέφωνα, σταθμοί εργασίας, κονσόλες παιχνιδιών και άλλα ενσωματωμένα υπολογιστικά συστήματα. Όσον αφορά τον υπολογιστή, η GPU μπορεί να συμμετέχει είτε ως ενσωματωμένο περιφερειακό της μητρικής κάρτας, ή ως εξαρτήματος συνδεδεμένης κάρτας γραφικών στον δίαυλο επικοινωνίας PCI-e και παλαιότερα AGP.

Τα παραδοσιακά προβλήματα που καλούνται να λύσουν οι επεξεργαστές γραφικών, είναι αυτά που αφορούν την ερμηνεία και παραγωγή γραφικών με τρόπο ταχύτερο του υπολογιστή και αποδεσμεύοντας τον επεξεργαστή από μεγάλο υπολογιστικό φόρτο, καθώς τα περισσότερα προβλήματα αυτά έχουν μεγάλο δείκτη επαναληψιμότητας και συνήθως είναι ευκόλως παραλληλήσιμα. Αυτό αιτιατά οδήγησε την εξέλιξη των επεξεργαστών γραφικών στην κατεύθυνση του παράλληλου προγραμματισμού και έτσι σταδιακά, οι εταιρείες της βιομηχανίας επεξεργαστών γραφικών υιοθέτησαν διάφορες παράλληλες αρχιτεκτονικές. Από το 2012 περίπου, οι κάρτες γραφικών πλέον είναι συστήματα πολυεπεξεργαστικά, υψηλής παραλληλίας και επιτρέπουν την αποδοτική χειραγώγηση μεγάλων ενοτήτων δεδομένων [93].

Ο Προγραμματισμός Γενικού Σκοπού Επεξεργαστών Γραφικών είναι η χρήση της GPU παράλληλα με την CPU για την επιτάχυνση εφαρμογών επιστημονικών, μηχανικής διαφόρων πεδίων και γενικού σκοπού. Αυτό σημαίνει, ότι πλέον δεν χρησιμοποιούνται μόνο για επεξεργασία και παραγωγή γραφικών. Για την υλοποίηση του προγραμματισμού GPGPU οι εταιρείες της βιομηχανίας αυτής, οδηγούμενες φυσικά και από τον ανταγωνισμό, σταδιακά ανέπτυξαν αναπτυξιακά εργαλεία και προγραμματιστικές διεπαφές, για να διευκολύνουν και να προσελκύσουν το ενδιαφέρον των προγραμματιστών και εταιριών λογισμικού στην χρήση της νέας αυτής τεχνολογίας. Ως εκ τούτου, σταδιακά αναμένεται η ενσωμάτωση του προγραμματισμού GPGPU σε ολοένα και περισσότερες εφαρμογές. Η διαδικασία αναβάθμισης ή εκ νέου συγγραφής του υπάρχοντος λογισμικού, αλλά και η κατάρτιση του προγραμματιστικού ανθρώπινου δυναμικού, είναι παράγοντες που περιορίζουν την άμεση διάδοση και επικράτηση αυτών των νέων τεχνικών.

Πλεονεκτήματα χρήσης GPU.

Ένας λόγος που είναι ωφέλιμο να χρησιμοποιηθεί η GPU, είναι το γεγονός ότι οι αρχιτεκτονικές επεξεργασίας βάσει καρτών γραφικών, επωφελούνται από την επιτόπου άμεση σχεδίαση και υλοποίηση, επειδή τα εξαρτήματα είναι ευρέως διαθέσιμα και σε προσιτές τιμές. Επιπροσθέτως, εφόσον ένας αλγόριθμος όρασης χαρτογραφηθεί στην GPU, θα συνεχίσει να επωφελείται από τις νέες βελτιώσεις του υλισμικού γραφικών αναλόγως και με την συχνότητα που η αγορά υιοθετεί τέτοιες βελτιώσεις.

Οι σύγχρονες GPU έχουν έναν ολοένα αυξανόμενο αριθμό επεξεργαστών, ικανών να διενεργήσουν ταυτόχρονους υπολογισμούς μεταβλητής υποδιαστολής, με υψηλό παράγοντα παραλληλίας, κάτι που επιταχύνει πολλές φορές την απόδοση της μεμονωμένης CPU.

Το αυξημένο εύρος ζώνης των μνημών που παρέχουν συστήματα με GPU, ωφελεί εφαρμογές όπως η αντιστοίχιση προτύπων, όπου ένα σύνολο δεδομένων αναφοράς μπορεί να αποθηκευτεί στην μνήμη κάθε κάρτας γραφικών. Πλέον, η ενσωματωμένη μνήμη μίας κάρτας γραφικών μπορεί να φτάσει σε μέγεθος αρκετών GB. Τυπικά είναι προσβάσιμη από τον επεξεργαστή γραφικών, σημαντικά πιο γρήγορα από ότι η CPU συνήθως επικοινωνεί με την κύρια μνήμη RAM. Οι σύγχρονες κάρτες γραφικών χρησιμοποιούν μνήμες τύπου GDDR5 που πετυχαίνουν ταχύτητες μετάδοσης της τάξεως των εκατοντάδων GB/s, μέγεθος αστρονομικό εν συγκρίσει με τις ταχύτητες μετάδοσης των εξίσου συγχρόνων CPU. Δεδομένου του ότι κάθε GPU είναι σε θέση να προσπελάζει την δική της μνήμη γραφικών, ανεξαρτήτως παρεμβάσεως από την CPU, κάθε επιπλέον κάρτα γραφικών προσθέτει εύρος ζώνης μνήμης στο συνολικό σύστημα.

Συστήματα πολλαπλών καρτών γραφικών.

Η χρήση πολλαπλών καρτών γραφικών για επεξεργασία γραφικών, υλοποιήθηκε αρχικά σε μορφή διεμπλοκής γραμμής σάρωσης. Τα προγράμματα για πολλαπλές GPU, στέλνουν υπολογιστικό φόρτο σε κάθε κάρτα γραφικών ξεχωριστά, θεωρώντας κάθε GPU ως ξεχωριστο υπολογιστικό πόρο, και η επικοινωνία μεταξύ καρτών επιτυγχάνεται μέσω μεταφοράς των αποτελεσμάτων πίσω στον δίαυλο κύριας μνήμης πριν διαμοιρασθούν στις υπόλοιπες GPU. Σε ερευνητικό επίπεδο, έχουν αξιοποιηθεί αρχιτεκτονικές βάσει παράλληλης επεξεργασίας συστημάτων Συστάδων (Clusters) με πολλαπλές κάρτες γραφικών, σε μία πληθώρα επιστημονικών πεδίων, όπως για παράδειγμα την εφαρμογή Folding@Home που αξιοποιεί τελικά Crowdsourcing, για την χρήση της υπολογιστικής ισχύος των μελών της, που συμμετέχουν από όλον τον κόσμο προκειμένου να εξομοιωθεί αναδίπλωση πρωτεϊνών. Σε αντίθεση με την επεξεργασία στην CPU, η GPU απαιτεί την μεταφορά των δεδομένων από και προς την κύρια μνήμη, μέσω του διαύλου του συστήματος. Οι όροι αυτοί έχουν συμβατικά ονομαστεί, ανέβασμα και κατέβασμα ή διάβασμα (upload, download, readback). Ωστόσο, ορισμένοι αλγόριθμοι εφαρμόζουν πολλαπλές προσαυξήσεις ή επαναλαμβανόμενα στάδια επεξεργασίας για την παραγωγή του τελικού αποτελέσματος. Η καθυστέρηση μεταφοράς διαύλου υπονοείται μόνο για την μεταφορά της αρχικής ενότητας δεδομένων, προς επεξεργασία από την κάρτα γραφικών και στην επιστροφή των τελικών αποτελεσμάτων αυτής, προς την κύρια μνήμη και την CPU. Για την ταχύτερη διευθέτηση επικοινωνίας, το υλικό γραφικών βρίσκεται τυπικά πάνω στον ταχύτερο διαθέσιμο δίαυλο, που στην σύγχρονη εκδοχή του είναι ο δίαυλος PCI-express 2.

2.2.2 Εργαλεία ανάπτυξης εφαρμογών GPGPU

Στα πρώιμα στάδια της εξάπλωσης του GPGPU, υπήρχε μόνο ένας τρόπος να προγραμματιστούν οι επεξεργαστές γραφικών. Ο τρόπος αυτός απαιτούσε τον προγραμματισμό της σωλήνωσης γραφικών, με χρήση τεχνολογιών χαμηλού επιπέδου όπως η γλώσσας μηχανής. Τα μοναδικά προγραμματιστικά περιβάλλοντα που αφορούσαν το αντικείμενο αυτό, εκμεταλλεύονταν διεπαφές παραγωγής γραφικών, όπως π.χ. το πρότυπο OpenGL και το Direct3D. Προσφάτως βέβαια, έχουν υλοποιηθεί γλώσσες υψηλού αλλά και μεσαίου επιπέδου από την βιομηχανία του χώρου, αλλά και ερευνητικά ιδρύματα, οι οποίες παρέχουν την απαραίτητη ευελιξία και χρηστικότητα, που είχε ανάγκη το προγραμματιστικό κοινό του GPGPU. Ακολούθως αναφέρονται εν συντομία ορισμένες από τις νέες γλώσσες αυτές, καθώς και τα αναπτυξιακά τους περιβάλλοντα και επισημαίνονται μερικά από τα πλεονεκτήματα, τα μειονεκτήματα και τα ιδιαίτερα γνωρίσματά τους.

Γλώσσες Σκίασης (Shaders).

Αυτά τα συστήματα ανάπτυξης, παρουσιάζουν σημαντική ομοιότητα με τα καθιερωμένα εργαλεία και μεθόδους προγραμματισμού γραφικών. Οι γλώσσες που τα στηρίζουν, δύνανται να αξιοποιηθούν για την παραγωγή εφαρμογών γενικού σκοπού, κάνοντας όμως χρήση της σωστής μοντελοποίησης της εκάστοτε εφαρμογής, με μορφή προβλήματος επεξεργασίας γραφικών. Ακολουθεί συνοπτική αναφορά και περιγραφή των σημαντικότερων από αυτές.

- *C for Graphics (Cg).*

  Η Cg αναπτύχθηκε από την NVIDIA [94] και σε μεγάλο βαθμό έδειξε τον δρόμο στην μετέπειτα ανάπτυξη της CUDA. Έχει σκοπίμως όμοια σύνταξη με την γλώσσα C, και κατ' επέκταση της C++ και της Java, ενσωματώνοντας και ορισμένα γνωρίσματα ορισμένων πιο παλιών γλωσσών σκίασης. Είναι ανεξάρτητη πλατφόρμας λειτουργικού συστήματος λόγω της συμβατότητάς της με το πρότυπο OpenGL, και του ανταγωνιστικού Direct3D. Για να συμβιβάσει τις διάφορες αρχιτεκτονικές επεξεργαστών γραφικών και κατ' αντιστοιχία τις ασυμβατότητες που παρουσιάζουν οι διάφορες προγραμματιστικές διεπαφές γραφικών, εμπερικλείει έναν σύνολο προτύπων, τα οποία εγγυώνται με την τήρησή τους πως οι παραγόμενες εφαρμογές, που θα έχουν συγγραφεί στην γλώσσα σκίασης αυτή, θα εκτελεστούν σε όσες κάρτες γραφικών υποστηρίζουν το επιλεγμένο πρότυπο.

- *High Level Shading language (HLSL).*

  Ταυτόχρονα με την εξάπλωση του DirectX 9, έκανε την εμφάνισή της η HLSL. Αποτελεί γλώσσα σκίασης υψηλού επιπέδου. Η συνεισφορά της ήταν κυρίως η αναβάθμιση από τις πρώιμες γλώσσες σκίασης, οι οποίες ήταν συμβατές με τις προηγούμενες εκδόσεις του DirectX, απαιτούσαν όμως την συγγραφή της εφαρμογής σε γλώσσα μηχανής. Η HLSL παρουσιάζει μεγάλο βαθμό ομοιότητας με την Cg καθώς και οι δύο αποτελούν προϊόν συνεργασίας της NVIDIA με την Microsoft, φέρουν ωστόσο διαφορετική ονομασία για εμπορικούς λόγους και για την ασυμβατότητα της HLSL με το πρότυπο OpenGL.

- *OpenGL Shading Language (GLSlang).*

  Αυτή η γλώσσα [95] είναι μία ακόμη της οποίας η σύνταξη, βασίζεται στην γλώσσα προγραμματισμού C, μολονότι ορισμένα γνωρίσματα της C φέρουν συμφόρηση στην επίδοση της. Παρεμβάλει στην C καινούριες δομές, όπως κάποιους τύπους διανυσμάτων και πινάκων, προκειμένου να διευκολύνει την επεξεργασία γραφικών. Ωστόσο υιοθετεί ορισμένα γνωρίσματα της C++ (π.χ. πολυμορφικές συναρτήσεις). Η πρώτη χρονολογικά συμβατότητά της, υπήρξε η έκδοση 1.4 του OpenGL βάσει της οποίας αναπτύχθηκε σε μεγάλο βαθμό η έκδοση 2.0 του OpenGL.

- *Brook.*

  Το πανεπιστήμιο Stanford, στα πλαίσια ερευνητικού προγράμματος [96] ανέπτυξε το BrookGPU, στοχεύοντας την διευκόλυνση ενός κοινού που δεν εξειδικεύεται στα γραφικά, επιτρέποντας την συγγραφή αποδοτικών εφαρμογών γενικού σκοπού για GPU. Όπως ήταν αναμενόμενο, έτυχε ευρείας αποδοχής από την επιστημονική κοινότητα, καθώς διευκόλυνε την υλοποίηση έργων που χρήζουν υψηλού βαθμού παραλληλίας και μεταθέτοντάς τα στην GPU, επετεύχθη αποτέλεσμα που αλλιώς θα ήταν ιδιαίτερα δαπανηρό σε πόρους οικονομικούς και μη. Συντακτικά, αποτελεί επέκταση της γλώσσας C, με προσθήκες που αξιοποιούν το προγραμματιστικό μοντέλο των ροών (streams). Για το BrookGPU, οι GPU είναι μια πλατφόρμα συνεπεξεργασίας για εκτέλεση παραλληλοποιήσιμου κώδικα. Παρέχει πρόγραμμα μεταγλώττισης, που ονομάζεται brcc. Οι ροές του Brook προσομοιάζουν πίνακες της C, αποθηκεύοντας όμως στην πράξη δεδομένα επεξεργασίας υφών. Κύρια διαφορά είναι η δυνατότητα επεξεργασίας των περιεχομένων τους, ταυτόχρονα από διαφορετικές επεξεργαστικές μονάδες. Οι πυρήνες υπολογισμού του Brook λειτουργούν όμοια με συναρτήσεις της C, με χαρακτηριστική διαφορά την παράλληλη εκτέλεσή τους από πολλαπλούς πυρήνες και αποτελούν τμηματικά προγράμματα. Ουσιαστικά, αυτό που κάνει το Brook, είναι να μετασχηματίζει ένα πραγματικό πρόβλημα στο πεδίο των γραφικών. Στην φάση ολοκλήρωσης, ένα πρόγραμμα Brook, πρακτικά αποσυντίθεται σε κλήσεις κάποιου API γραφικών.

- *Microsoft Accelerator*

  Το Microsoft Accelerator δημιουργήθηκε από την Microsoft, στο πλαίσιο ερευνητικού προγράμματος και είχε ως αποστολή την απλούστευση του προγραμματισμού GPGPU μέσω μιας γλώσσας υψηλού επιπέδου. Ο στόχος ήταν, η καταλληλότητα της γλώσσας αυτής για ανάπτυξη εφαρμογών υψηλής παραλληλισιμότητας δεδομένων, συνεισφέροντας προγραμματιστική βιβλιοθήκη, ικανή να αξιοποιηθεί ακόμη και σε άλλες γλώσσες προγραμματισμού. Ο Accelerator μεταφράζει τις παράλληλες θεωρητικές λειτουργίες, σε αντίστοιχες εντολές γλώσσας σκίασης pixel, επιτυγχάνοντας αισθητή επιτάχυνση σε σχέση με τον αντίστοιχο ακολουθιακό κώδικα πλήρως εκτελούμενου σε CPU.

- *MATLAB*

  Το MATLAB είναι ιδιαίτερα διαδεδομένο στην επιστημονική και ερευνητική κοινότητα, για την διενέργεια μαθηματικών υπολογισμών και ως γλώσσα προγραμματισμού για εφαρμογές υψηλής μαθηματικής πολυπλοκότητας. Ενώ ήδη χρησιμοποιεί βιβλιοθήκες με καλή απόδοση ακολουθιακά, είναι δυνατόν να επιτευχθούν ακόμα υψηλότερες επιδόσεις σε εφαρμογές που τρέχουν με χρήση GPU [97]. Γι αυτό τον σκοπό χρησιμοποιούνται οι επεκτάσεις αρχείου MEX (Matlab EXecutable), που καλούν τμηματικά κώδικα γλώσσας C ή FORTRAN. Επίσης το MATLAB προσφέρει δυνατότητες επιτάχυνσης μέσω των αντίστοιχων προγραμματιστικών βιβλιοθηκών της τεχνολογίας CUDA.

Ωστόσο, οι δύο ευρύτερα διαδεδομένες και επαρκέστερα υποστηριζόμενες τεχνολογίες, είναι το CUDA (Compute Unified Device Architecture) της NVIDIA και η ανοιχτού κώδικα OpenCL (Open Computing Language). Στο πλαίσιο αυτής της εργασίας, γίνεται χρήση της τεχνολογίας CUDA, καθώς παρουσιάζει την καλύτερη συμβατότητα με τα επιλεγμένα εργαλεία ανάπτυξης της εφαρμογής, καθώς και την βαθύτερη υποστήριξη του υλικού και των θεωρητικών εννοιών που πραγματεύεται. Ως επί το πλείστον όμως, οι παρόμοιες αυτές τεχνολογίες μοιράζονται ορισμένα σημεία αρχιτεκτονικής και υιοθετούν το ίδιο πρωτόκολλο υλοποίησης, που αποτελείται από τα ακόλουθα βήματα και μεθόδους:

- Μεταγλώττιση: Σε αυτό το στάδιο παράγεται τμηματικά ο αντίστοιχος κώδικας για την CPU και την GPU από εξειδικευμένους μεταγλωττιστές (compilers) και στην συνέχεια διασυνδέονται σε ένα εκτελέσιμο πρόγραμμα.
- Εκτέλεση: Οι επιμέρους κώδικες εκτελούνται ο καθένας στην αντίστοιχή του συσκευή, διατηρώντας διαφοροποιημένη μνήμη για την GPU και την CPU. Τα δεδομένα προς επεξεργασία από την GPU, πρέπει αρχικά να μεταφερθούν από την CPU, και στην συνέχεια γίνεται κλήση του GPU κώδικα (kernel code). Στο τέλος τα αποτελέσματα επεξεργασίας από την κάρτα γραφικών, μεταφέρονται ξανά στην CPU μέσω του διαύλου του συστήματος και την κύρια μνήμη.

Όπως γενικά στον παράλληλο προγραμματισμό, έτσι και στον προγραμματισμό GPGPU, υιοθετούνται μέθοδοι διαχείρισης των παραλληλισμένων τμημάτων του προγράμματος. Ενδεικτικά αναφέρονται οι σημαντικότερες.

- Map: Σε αυτό το στάδιο αντιστοιχίζεται μία παραλληλίσιμη επαναληπτική διαδικασία ή πράξη, σε όλες τις όμοιες παράλληλες επεξεργαστικές μονάδες. Σε αυτήν την κατηγορία ανήκουν οι αριθμητικές πράξεις, όπως για παράδειγμα, ένα φίλτρο φωτεινότητας, όπου όλα τα στοιχεία που απαρτίζουν τον πίνακα της εικόνας πρέπει να πολλαπλασιαστούν με την ίδια σταθερά. Η χρήση του Map είναι αρκετά φιλική προς τον χρήστη. Ο προγραμματιστής αναλαμβάνει τον διαμερισμό του προβλήματος σε ομάδες pixel και να εφαρμόσει την ίδια πράξη σε όλες αυτές.
- Reduce: Σε αυτό το στάδιο πραγματοποιείται η τελική συγκέντρωση των επεξεργασμένων δεδομένων από τις επιμέρους παράλληλες διεργασίες. Γενικά, το Reduction υλοποιείται με διάφορες ιεραρχίες συλλογής, όπως πυραμιδικά όπου τα επεξεργασμένα δεδομένα του reduction ανώτερου επιπέδου, αποτελούν δεδομένα εισόδου για το επόμενο reduction, καταλήγοντας ιεραρχικά σε ένα συνολικό αποτέλεσμα
- Άλλες κοινές μέθοδοι για τον διαμοιρασμό των υποέργων και την συλογή των αποτελεσμάτων, είναι οι Scatter, Gather, ενώ υποστηρίζονται λειτουργίες Ταξινόμησης, Αναζήτησης και άλλες.

2.3 Περιορισμοί και προβλήματα στην χρήση των GPU

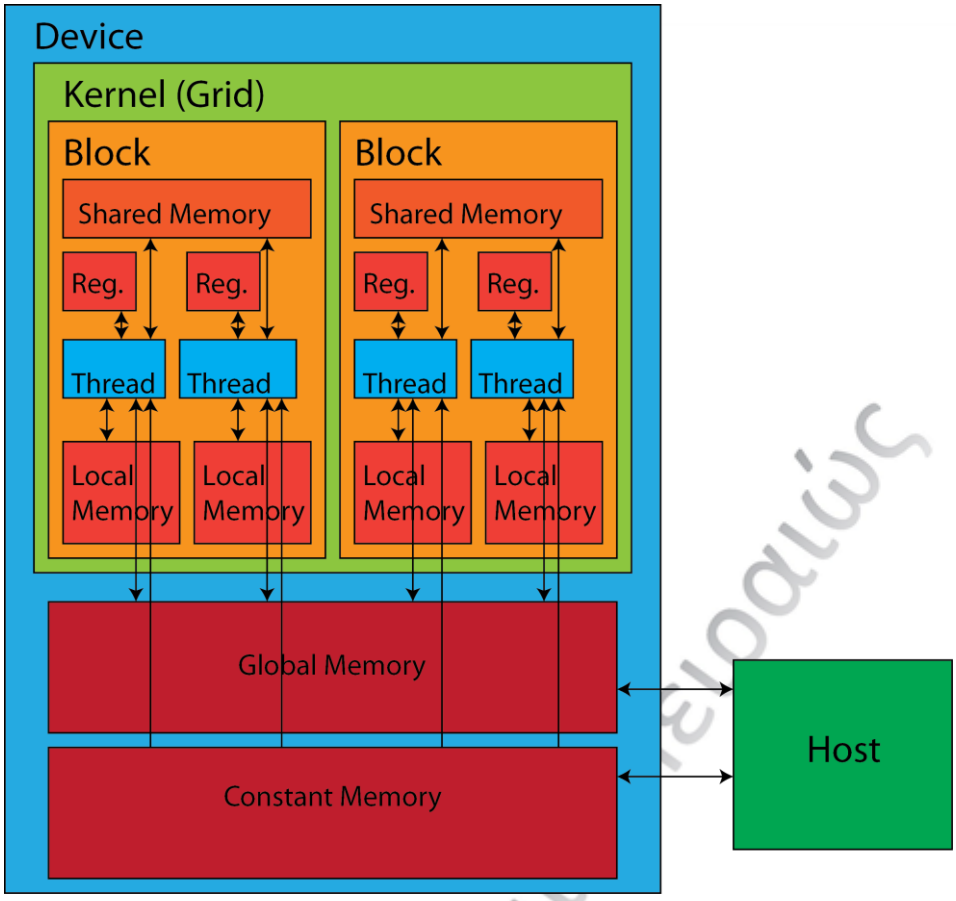
Ο προγραμματισμός GPGPU είναι ακόμα μία αναπτυσσόμενη δυναμικά τεχνολογία. Η «διαστρέβλωση» της προβλεπόμενης χρήσης των GPU, αποκτά περισσότερο νόημα, όσο με την εξέλιξη της τεχνολογίας αυξάνεται ο αριθμός επεξεργαστικών πυρήνων τους σε σύγκριση με αυτόν των CPU, που αιτιατά οδήγησε στην εκρηκτική ανάπτυξη της επεξεργαστικής δύναμης των GPU, ούτως ώστε να μπορούν να επεξεργαστούν κάθε είδος δεδομένων. Παρά την ύπαρξη εκατοντάδων ή χιλιάδων επεξεργαστικών πυρήνων στις GPU, η μεμονωμένη ταχύτητα καθενός από αυτούς, είναι μικρότερη από τους αντίστοιχης τεχνολογίας πυρήνες μιας CPU και παρά την θεωρητική τους πληρότητα κατά το κριτήριο Turing εμφανίζουν σημαντικά λιγότερη προγραμματιστική ευελιξία. Συγκριτικά με τις CPU, αισθητή είναι η απουσία γνωρισμάτων, όπως οι διακοπές (interrupts) και η εικονική μνήμη (virtual memory). Ως εκ τούτου, οι μεμονωμένοι πυρήνες των GPU, παρουσιάζουν μειωμένη απόδοση σε προβλήματα που χρήζουν ακολουθιακού προγραμματισμού. Πολλές φορές, οι ανθρώπινοι πόροι που απαιτούνται για να μεταφραστεί μια υπάρχουσα εφαρμογή στην παράλληλη έκδοσή, της δεν συμφέρουν σε σχέση με την τελική βελτίωση της απόδοσής της. Οι διαδικασίες θεωρητικού επανασχεδιασμού, μοντελοποίησης και αποσφαλμάτωσης των παράλληλων εφαρμογών, προσθέτουν συνήθως επίπονο προγραμματιστικό έργο. Δεδομένου του ότι οι μεγάλες βιομηχανίες καρτών γραφικών, για λόγους ανταγωνισμού και προστασίας των πρωτότυπων τεχνολογιών τους, δεν υιοθετούν πρακτικές ελεύθερου λογισμικού για τις τεχνολογίες των προϊόντων τους, η διαδικασία της βελτιστοποίησης ενός προγράμματος, συχνά περιορίζεται ως προς την συνεισφορά του προγραμματιστή. Αναφορικά η τεχνολογία CUDA της NVIDIA, είναι ιδιόκτητη και κλειστού κώδικα, γεγονός που συχνά φέρει εμπόδια στην πλήρη εκμετάλλευση των δυνατοτήτων της GPU. Η μη ορθολογική και έμπειρη χρήση των GPU, συνιστά σπατάλη υπολογιστικών πόρων και ενίοτε ζημιά για την απόδοση του συστήματος, δεδομένης και της υψηλής κατανάλωσης ισχύος τους.

2.3.1 Η τεχνολογία CUDA

Η CUDA (Compute Unified Device Architecture) [98] είναι μια πλήρης πλατφόρμα παράλληλου προγραμματισμού και ταυτόχρονα ένα αρχιτεκτονικό μοντέλο. Η χρήση του συντελεί στην ενίσχυση της υπολογιστικής απόδοσης, αξιοποιώντας την ισχύ της GPU. Δημιουργήθηκε το 2007 από την εταιρεία NVIDIA και έκτοτε γιγαντώθηκε μέσω πληθώρας εφαρμογών και επιστημονικών μελετών. Εκτιμάται ότι εκατοντάδες εκατομμύρια συσκευές, που περιλαμβάνουν laptops, desktops, μέχρι πανίσχυρα clusters και υπέρ-υπολογιστές χρησιμοποιούν CUDA παρόντως, ενώ χιλιάδες ερευνητές σε όλο τον κόσμο εργάζονται πειραματικά πάνω σε αυτήν.

Το μοντέλο Υπολογισμού

Στο CUDA , ομοίως με τα αντίστοιχα αναπτυξιακά πακέτα για προγραμματισμό GPGPU , η βασική σκοπιμότητα είναι η τμηματική επιτάχυνση συγκεκριμένων υποέργων του προβλήματος από τον επεξεργαστή γραφικών. Τα ενεργοβόρα αυτά υπολογιστικά υποσυστήματα ονομάζονται πυρήνες (kernels). Η ονομασία τους πηγάζει περισσότερο από την Επεξεργασία Σήματος και Εικόνας, παρά από τον κλάδο των Λειτουργικών Συστημάτων. Ένα πρόγραμμα, μια συνάρτηση ή μια προγραμματιστική βιβλιοθήκη, μπορεί να αποτελείται από έναν ή περισσότερους πυρήνες εκτέλεσης στην GPU. Οι πυρήνες αυτοί καλούνται από το πρόγραμμα που εκτελείται στον χώρο της CPU και της κεντρικής μνήμης (host) και η εκτέλεσή τους μετατίθεται στην GPU (device). Η συγγραφή τους γίνεται κυρίως σε μία κοινότυπη γλώσσα, συγκεκριμένα την C , επεκταμένη με επιπλέον δεσμευμένες εντολές, που σηματοδοτούν ευκολότερα την παραλληλία εν συγκρίσει με την παραδοσιακή χρήση επαναληπτικών βρόγχων. Το CUDA ακολουθεί την Αρχιτεκτονική SIMD (Single Instruction Multiple Data), που σημαίνει ότι το ίδιο σύνολο εντολών του πυρήνα εκτελείται πάνω σε διαφορετικά δεδομένα. Αφού οι κώδικες των πυρήνων, περάσουν την διαδικασία της μεταγλώττισης, μοντελοποιούνται πλέον σε πολλαπλά νήματα (threads), τα οποία δηλώνονται προγραμματιστικά κατά την συγγραφή και στην συνέχεια εκτελούν το ίδιο πρόγραμμα σε παραλληλία. Ουσιαστικά, κάθε νήμα αντικαθιστά μια ακολουθιακή επανάληψη του βρόχου. Για παράδειγμα σε έναν αλγόριθμο επεξεργασίας εικόνας, κάθε thread επεξεργάζεται ένα pixel, ενώ συνολικά όλος ο πυρήνας επεξεργάζεται την εικόνα.



2-ii Οργάνωση Μνήμης σε NVIDIA κάρτες ικανές για CUDA.
Πηγή: Cornell University Center for Advanced Computing

Στην συνέχεια, πολλαπλά threads οργανώνονται σε thread blocks, που διευθυνσιοδοτούνται μέχρι και σε τρείς διαστάσεις, και περιέχουν μέχρι έναν συγκεκριμένο αριθμό από threads που υπαγορεύεται από την αρχιτεκτονική κάθε συγκεκριμένου επεξεργαστή. Κάθε thread ενός block, θα τρέξει τελικά σε έναν πολυεπεξεργαστή ροής (Streaming Multiprocessor-SM). Επιτρέπεται στα διαφορετικά thread blocks να συνεργάζονται και να μοιράζονται κοινή μνήμη. Τα blocks χωρίζονται σε ομάδες των 32 threads που ονομάζονται αναδιπλώσεις (warps), και αποτελούν την μονάδα αυτή που θα σταλεί εν τέλει προς εκτέλεση σε έναν SM. Τέλος τα thread blocks συναθροίζονται σε πλέγματα (grids), καθένα εκ των οποίων επιτελεί το τμήμα προγράμματος, που περιγράφεται από έναν μοναδικό πυρήνα (kernel). Οι προαναφερθείσες υπολογιστικές μονάδες φέρουν αναγνωριστικά (identifiers), που προσδιορίζουν τη συσχέτισή τους με τον πυρήνα. Αυτά τα αναγνωριστικά χρησιμοποιούνται από κάθε thread, ως δείκτες ανακατεύθυνσης των δεδομένων εισόδου και εξόδου, ώστε να προσδιορίσουν για τον συγκεκριμένο υπολογισμό που θα πραγματοποιήσει το καθένα.

Εκτέλεση Εφαρμογής

Η κάρτα γραφικών (GTS 450) που χρησιμοποιείται για την παρούσα εργασία, χρησιμοποιεί αρχιτεκτονική Fermi. Κάθε διακριτή χρονική στιγμή, βάσει της αρχιτεκτονικής αυτής, η συσκευή μπορεί να εκτελεί μια εφαρμογή μεμονωμένα, η οποία αποτελείται από πολλούς υπολογιστικούς πυρήνες. Η παράλληλη εκτέλεση πολλαπλών πυρήνων διενεργείται με την ανάθεση κάθε πυρήνα σε έναν ή περισσότερους SM κάθε φορά. Έτσι επιτυγχάνεται η μεγιστοποίηση χρήσης της συσκευής. Η εναλλαγή (switching) μεταξύ εκτελούμενων εφαρμογών, είναι πολύ σύντομη (περίπου 20 μs) και έτσι διατηρείται μεγάλη αξιοποίηση των υπολογιστικών πόρων της κάρτας γραφικών και ταυτόχρονα αξιοποιείται ιδανικά η παραλληλία. Η διαχείριση των εναλλαγών ανατίθεται στο ειδικό τμήμα χρονοδρομολόγησης του ολοκληρωμένου κυκλώματος, που ονομάζεται Giga Thread Scheduler. Το τμήμα αυτό, διαχειρίζεται μέχρι και τον μέγιστο επιτρεπόμενο αριθμό ενεργών νημάτων για κάθε SM, τα οποία κατανέμονται μέχρι και σε 16 πυρήνες.

Γλώσσες Προγραμματισμού με CUDA.

Το CUDA κατά βάσει είναι συμβατό με την γλώσσα C, η οποία είναι και η πιο διαδεδομένη παγκοσμίως (μαζί με την C++) και σε αυτήν είναι γραμμένες και οι περισσότερες εφαρμογές του. Στην CUDA C έχουν εισαχθεί συντακτικές επεκτάσεις, που καθιστούν δυνατή την κατανομή των δεδομένων προς επεξεργασία ενός πυρήνα στις δομές που αναλύθηκαν ανωτέρω. Είναι δυνατόν ωστόσο, να αξιοποιηθεί για την επιτάχυνση προγραμμάτων που έχουν γραφθεί σε FORTRAN, Java, Matlab και Python. Επίσης, εκτός από το δικό της αναπτυξιακό περιβάλλον, υποστηρίζεται από το πρότυπο OpenCL και το Direct Compute API της Microsoft.

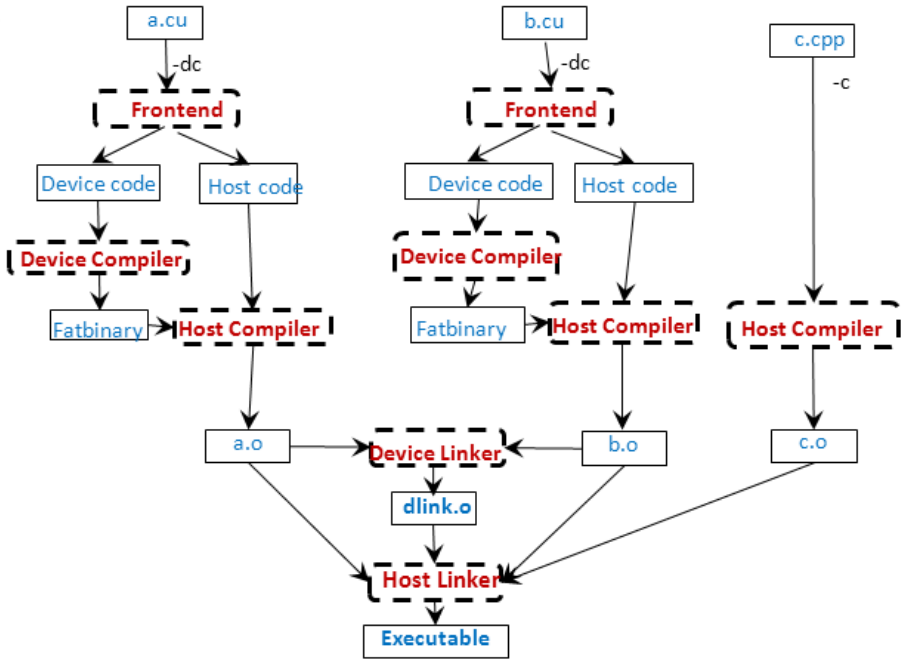
Επίπεδα Υλοποίησης Εφαρμογής

Η στοίβα λογισμικού του CUDA αποτελείται από τα ακόλουθα.

- Το CUDA Driver API που επιτελεί τους χειρισμούς χαμηλού επιπέδου. Είναι κλειστού κώδικα και δυνατότητα επεξεργασίας του έχει μόνο η NVIDIA
- Το CUDA Runtime API δημιουργήθηκε για τον προγραμματισμό των εφαρμογών σε CUDA σε πιο υψηλό επίπεδο από το Driver API και αποτελεί το τελικό εργαλείο συγγραφής του κώδικα του προγραμματιστή.
- Οι CUDA libraries είναι ένα σύνολο συναρτήσεων ευρείας χρήσης για επιστημονικές εφαρμογές, που έχουν υλοποιηθεί και βελτιστοποιηθεί για εκτέλεση στις GPU της NVIDIA. Τέτοιες είναι η CUBLAS για τη Γραμμική Άλγεβρα και η CUFFT για τον Ταχύ Μετασχηματισμό Fourier.
- Το CUDA Toolkit που πέρα από διάφορα χρήσιμα εργαλεία για την ανάπτυξη και αξιολόγηση εφαρμογών CUDA, περιέχει τον Compiler της NVIDIA nvcc, που αποτελεί μίμηση του gcc και σε συνεργασία με αυτόν, ή άλλους μεταγλωττιστές παράγει το εκτελέσιμο αρχείο που εκτελείται τμηματικά στην GPU και την CPU.

Μεταγλώττιση και Εκτέλεση

Οι εφαρμογές γραμμένες σε CUDA αποτελούνται από μίξη συμβατικού κώδικα για CPU, που έχει συγγραφεί σε C, C++, Fortran ή άλλη υποστηριζόμενη γλώσσα υψηλού επιπέδου, κάνοντας χρήση των συναρτήσεων της GPU. Βάσει της διαδικασίας μεταγλώττισης, οι συναρτήσεις προς εκτέλεση στην GPU, διαχωρίζονται και μεταγλωττίζονται από τον ιδιόκτητο μεταγλωττιστή της NVIDIA (nvcc), ενώ το υπόλοιπο πρόγραμμα μεταγλωττίζεται από τους κλασσικούς μεταγλωττιστές, που είναι διαθέσιμοι στο κάθε αναπτυξιακό περιβάλλον (π.χ. gcc, VC, MinGW). Στην συνέχεια οι συναρτήσεις της GPU, ενσωματώνονται σαν εικόνες στο object file που παράχθηκε. Τελικά, διενεργείται η σύνδεση (linking), όπου συγκεκριμένες βιβλιοθήκες χρόνου εκτέλεσης της CUDA, υλοποιούν την υποστήριξη για χειρισμό των GPU. Τέτοιος χειρισμός αφορά την δέσμευση χώρου μνήμης στη συσκευή επιταχυντή και την μεταφορά από τη μνήμη συσκευής στη μνήμη του συστήματος. Ένα σχηματικό παράδειγμα της διαδικασίας μεταγλώττισης και σύνδεσης φαίνεται παρακάτω.



2-iii Παράσταση μεικτής μεταγλώττισης προγραμματιστικού έργου με CUDA

Πηγη:http://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/graphics/nvcc-options-for-separate-compilation.png

Αξίζει να παρατηρηθεί ότι ο κώδικας που θα εκτελεστεί στη συσκευή, αφού διαχωριστεί από τον κώδικα της CPU, μετατρέπεται σε μια ενδιάμεση μορφή που ονομάζεται PTX Assembly (Parallel Thread eXecution), που παρατηρείται σε όλες τις συναφείς αρχιτεκτονικές και στην συνέχεια, αναλόγως της συγκεκριμένης πλατφόρμας που θα εκτελεστεί ο κώδικας, διενεργείται ένα επιπρόσθετο στάδιο μεταγλώττισης, που έχει ως σκοπό την παραγωγή του εκτελέσιμου κώδικα της συσκευής. Ο μεταγλωττιστής nvcc παρέχει την δυνατότητα στον προγραμματιστή, να δημιουργήσει τα *.ptx αρχεία, τα οποία όμως δεν έχουν άλλη χρήση εκτός του CUDA Driver API.

Η εναλλακτική της OpenCL

Η OpenCL (Open Computing Language) είναι μια νέα βιβλιοθήκη [99], που περιέχει σύνολο προτύπων που στοχεύουν τόσο τις GPU, τις CPU και πιθανώς στο μέλλον, άλλους τύπους επιταχυντών υλικού. Για τη διαχείριση του προτύπου, υπεύθυνο είναι το Khronos Group, το οποίο διατηρεί και το γνωστό πρότυπο γραφικών OpenGL. Σε αντίθεση με την CUDA, η OpenCL υλοποιείται μόνο ως βιβλιοθήκη. Πρακτικά, αυτό σημαίνει ότι το έργο της συμπίεσης, αποσυμπίεσης των παραμέτρων των πυρήνων, και άλλες συναφείς εργασίες εναποτίθενται στα χέρια του προγραμματιστή της εκάστοτε εφαρμογής, που την χρησιμοποιεί. Εξαιτίας της προσέγγισης αυτής οι πυρήνες GPU δεν μεταγλωττίζονται μαζί με την υπόλοιπη εφαρμογή, αλλά κατά τον χρόνο εκτέλεσης, από την ίδια την OpenCL. Αυτό πραγματοποιείται, αποστέλλοντας τον πηγαίο κώδικα του πυρήνα ως σύνολο από συμβολοσειρές (strings) στο αντίστοιχο OpenCL API της εκάστοτε αρχιτεκτονικής. Όταν οι πυρήνες μεταγλωττιστούν, η καλούσα εφαρμογή αναλαμβάνει την διαχείριση αυτών των πυρήνων. Πρακτικά, απαιτείται πολύ περισσότερος κώδικας συγκριτικά με μία όμοια εφαρμογή σε CUDA, αν και οι λειτουργίες της είναι αρκετά απλές ως προς την διαχείριση τους. Κατά τα άλλα, οι βασικές προγραμματιστικές αρχές, είναι παρόμοιες με της CUDA.

Στην OpenCL περιλαμβάνεται και η υλοποίηση GPGPU της εταιρείας ATI επ' ονόματι ATIStream, η οποία δεν θα αναλυθεί εκτενώς, καθότι η παρούσα εργασία υλοποιείται με χρήση υλικού και λογισμικού της εταιρείας NVIDIA. Ωστόσο, αξίζει να αναφερθεί ότι η βιβλιοθήκη αυτή, καθώς και αρκετές άλλων εταιριών, σταδιακά έφτασαν να ανταγωνίζονται την ευρεία διάδοση και τις δυνατότητες της CUDA, μάλιστα σε ορισμένες περιπτώσεις, κάνοντας χρήση λογισμικού ανοικτού κώδικα, κάτι που προοιωνίζει την ραγδαία ανάπτυξή τους μέσω της συνεισφοράς της προγραμματιστικής κοινότητας. Επίσης, ήδη γίνονται προσπάθειες να συμπεριληφθούν συναρτήσεις χειρισμού εξωτερικών μονάδων με βάση FPGA κάτι που θα αποτελούσε πραγματικό άλμα στο πεδίο της επιτάχυνσης επεξεργασίας.

2.4 Επεξεργασία Εικόνας και Μηχανική Όραση: Επιτάχυνση με επεξεργαστές γραφικών

Η παραγωγή γραφικών και η Υπολογιστική Όραση είναι προσεγγιστικά αντίστροφοι τομείς της τεχνολογίας [10], ή πιο συγκεκριμένα, η σύνθεση εικόνας ως προς την ανάλυση εικόνας. Τα γραφικά μπορούν να θεωρηθούν, σε υψηλό επίπεδο τουλάχιστον, ως η πράξη της επέκτασης δεδομένων εισόδου. Για παράδειγμα, ένα σύνολο διανυσμάτων και προσδιορισμών φωτεινότητας, επεκτείνονται σε ψηφιδωτά δεδομένων αξίας πολλών megapixel, ενώ αντίστροφα, στην όραση, η κίνηση μία ροή εικόνων (βίντεο) από μία κάμερα δύναται να προσδιοριστεί από ζεύγη εικόνων, που έχουν μέγεθος αντίστοιχο των πολλών megapixel και να εκφράσει το αποτέλεσμα σε πρακτικά πολύ λίγες παραμέτρους. Με την ενσωμάτωση στο υλισμικό γραφικών των εξειδικευμένων επεξεργαστών GPU, υπάρχει πλέον η δυνατότητα, μια αρχιτεκτονική υψηλού επιπέδου παραλληλίας να πραγματοποιεί λειτουργίες παραγωγής γραφικών ιδιαίτερα γρήγορα [11].

Με την εξέλιξη των πολύπειρων αλγορίθμων γραφικών, χρειάστηκε να αντιμετωπιστεί η ανάγκη ανάπτυξης περισσότερο ευέλικτου υλικού και προγραμματιστικών περιβαλλόντων, πράγμα που οδήγησε στην ανάπτυξη υλισμικού προγραμματίσιμου από τον χρήστη. Η νέα αυτή ευελιξία, σε συνδυασμό με την δυνατότητα υπολογισμών κινητής υποδιαστολής και την εξειδικευμένη επίδοση έναντι επεξεργαστών CPU με δυνατότητα ταυτοχρονισμού, οδήγησαν στην άνθιση του προγραμματισμού γραφικών επεξεργαστών γενικού σκοπού (GPGPU). Η τάση αυτή προς μεγαλύτερη προγραμματισιμότητα συνεχίστηκε, με αποτέλεσμα οι νέες αρχιτεκτονικές GPU και τα αντίστοιχά τους προγραμματιστικά περιβάλλοντα να μην βασίζονται στην επεξεργαστική σωλήνωση της παραγωγής γραφικών, αλλά να είναι σε θέση να εκμεταλλευτούν την παράλληλη φύση της GPU για επεξεργασία εικόνας και μηχανική όραση. Για να καταστεί η εκμετάλλευση αυτή αποδοτική, πρέπει να βρεθούν χαρτογραφήσεις και αντιστοιχίσεις των αλγορίθμων στην αρχιτεκτονική της GPU.

Οι πρώτες απόπειρες χρήσης της GPU για επεξεργασία εικόνας, έγιναν με χρήση των σωληνωμένων λειτουργιών για την επιτάχυνση παραγωγής γραφικών σχημάτων ή υφών ,από δεδομένα εικόνας εισόδου από τον υπολογιστή. Εφόσον τα αποτελέσματα της επεξεργασίας αυτής μπορούσαν να μεταφερθούν πίσω στο σύστημα, ήταν λογικό επόμενο να χρησιμοποιηθούν οι ίδιες αυτές λειτουργίες για την διενέργεια βασικών διαδικασιών επεξεργασίας εικόνας. Το ότι κατέστησαν προγραμματίσιμες οι κάρτες γραφικών με την προσθήκη σκιάσεων, ήταν ένα μεγάλο βήμα προόδου. Αυτό επέτρεψε στους προγραμματιστές να συγγράψουν ειδικά προγράμματα, τα οποία εκτελούνται από την GPU, για κάθε τρισδιάστατο σημείο μίας επιφάνειας και όπου κάθε pixel προβαλλόταν στον «καμβά» εξόδου. Αυτό επέκτεινε απέραντα την επεξεργαστική δυνατότητα της GPU και οι δημιουργικότεροι προγραμματιστές του χώρου, ξεκίνησαν να δοκιμάζουν υπολογισμούς γενικού σκοπού στην GPU, τιθασεύοντας τον επιταχυντή γραφικών, για χρήση σε έργα για τα οποία δεν προοριζόταν. Σύντομα, η GPU κατέστη ένα χρήσιμο εργαλείο για την επεξεργασία εικόνας και την μηχανική όραση. Οι σκιάσεις γραφικών βέβαια, δεν παρείχαν πρόσβαση σε πολλές χρήσιμες δυνατότητες του υλικού, όπως ο συγχρονισμός και οι λειτουργίες ατομικής μνήμης. Οι σύγχρονες γλώσσες προγραμματισμού GPU είναι σχεδιασμένες αποκλειστικά για να υποστηρίζουν υπολογισμό γενικού σκοπού στο υλικό. Οι GPU δεν είναι ακόμα το ίδιο ευέλικτες με τις CPU, αλλά πραγματοποιούν την επεξεργασία παράλληλων ροών δεδομένων πολύ πιο αποδοτικά και ένας αυξανόμενος αριθμός εφαρμογών, μη σχετιζόμενων με γραφικά, συγγράφονται εκ νέου κάνοντας χρήση των γλωσσών υπολογισμού σε GPU [100]. Η μηχανική όραση είναι ένας τύπος εφαρμογής, που χαρτογραφείται και αντιστοιχίζεται με φυσικό τρόπο στην GPU. Αυτό δεν αποτελεί σύμπτωση , καθώς η μηχανική όραση στην ουσία λύνει το αντίστροφο πρόβλημα από αυτό των γραφικών όπως αναφέρθηκε πιο πάνω. Οι GPU περιέχουν πολλές παρόμοιες επεξεργαστικές μονάδες και είναι ιδιαίτερα αποτελεσματικές στην εκτέλεση απλών, ομοίων υποέργων, όπως η δημιουργία ή το φιλτράρισμα μεμονωμένων pixel. Τέτοια έργα, όπως έχει ήδη αναφερθεί, ανήκουν στην κατηγορία των επαίσχυντα παραλληλήσιμων. Ωστόσο, πολλά έργα δεν παραλληλίζονται εύκολα, καθώς περιέχουν ακολουθιακά τμήματα όπου τα αποτελέσματα των μετέπειτα σταδίων, εξαρτώνται από τα αποτελέσματα προηγούμενων. Σε αυτήν την κατηγορία ανήκουν αλγόριθμοι αριθμητικής βελτιστοποίησης και δένδρα αναζήτησης βάσει σωρού. Επειδή πολλά έργα υψηλού επιπέδου αποτελούνται από ακολουθιακά και παράλληλα υποέργα, το συνολικό έργο μπορεί να επιταχυνθεί, εκτελώντας ορισμένα από τα συστατικά του στην CPU και τα υπόλοιπα στην GPU. Δυστυχώς, αυτό συνιστά δύο πηγές ανεπάρκειας, που αποτελούν και τις βασικές προκλήσεις στην επιτάχυνση της μηχανικής όρασης σε ένα ετερογενές σύστημα CPU-GPU.

Οι προκλήσεις αυτές είναι:

- Η μία είναι ο συγχρονισμός. Όταν δηλαδή ένα υποέργο εξαρτάται από το αποτέλεσμα ενός άλλου, το μετέπειτα στάδιο πρέπει να περιμένει μέχρι να ολοκληρωθεί το προηγούμενο.
- Η άλλη είναι η καθυστέρηση της μετακίνησης των δεδομένων μεταξύ των μνημών των GPU και CPU, και δεδομένου του ότι τα έργα μηχανικής όρασης χρειάζεται να επεξεργαστούν πολλαπλά pixel, μπορεί να σημάνει ότι χρειάζεται να μετακινηθούν μεγάλοι όγκοι πληροφορίας.

2.4.1 Εργαλεία Επεξεργασίας Εικόνας και Μηχανικής Όρασης

Όσο αναπτύσσεται ο τομέας της Μηχανικής Όρασης και η ζήτηση εφαρμογών του αυξάνεται, αναπόφευκτα θα εμφανίζονται νέες τεχνολογίες και υλοποιήσεις των παραδοσιακών αλλά και καινοτόμων θεωρητικών αρχών που τον αποτελούν. Για την αξιοποίηση αυτών, με διάφορα επίπεδα λεπτομέρειας και κατ' αναλογία πολυπλοκότητας και δυσκολίας έχουν αναπτυχθεί μέχρι σήμερα αρκετά πακέτα λογισμικού, προγραμματιστικές βιβλιοθήκες και περιβάλλοντα ανάπτυξης, που είτε έχουν εξ' ολοκλήρου στόχο, ή απλά προσφέρουν εργαλεία που εξυπηρετούν τις ανάγκες της επεξεργασίας εικόνας και της μηχανικής όρασης [101]. Ορισμένα από αυτά αναφέρονται ακολούθως.

- MATLAB - matrix laboratory
- OpenCV - Open Computer Vision
- libCVD - computer vision library
- IPPL - Intel Performance Primitives Library
- NI Vision (National Instruments)
- GpuCV: GPU-accelerated Computer Vision
- Halcon - Machine Vision
- Open eVision - Image Analysis Software Tools
- Adaptive Vision Library - Machine Vision Library for C++ and .NET
- Common Vision Blox
- VXL - C++ Libraries for Computer Vision Research and Implementation
- The CImg Library - C++ toolkit for image processing
- Camellia - Image Processing & Computer Vision library
- SimpleCV - Computer Vision platform using Python.
- Microsoft Vision SDK

Από τα πακέτα αυτά, αναμφίβολα το πιο ευρέως διαδεδομένο πλέον είναι το OpenCV [102], χρήση του οποίου έγινε και στα πλαίσια της εργασίας αυτής. Σταδιακά δείχνει να καθιερώνεται ως κυρίαρχο πακέτο καθώς ενσωματώνεται σε δημοφιλή λογισμικά όπως το Matlab και υποστηρίζει τεχνολογίες παράλληλης επεξεργασίας όπως CUDA, OpenCL, OpenMP. Σε ορισμένες περιπτώσεις μάλιστα, ανεξάρτητα πακέτα λογισμικού, υιοθετήθηκαν ή και ενσωματώθηκαν στην βιβλιοθήκη του όπως το GpuCV [12] και τα IPPL. Επίσης, είναι σε πολύ μεγάλο βαθμό ανεξάρτητο πλατφόρμας και λειτουργικού συστήματος, και έχει χρησιμοποιηθεί από εταιρείες κολοσσούς της πληροφορικής όπως η Microsoft, η Google και η Intel.

2.4.2 OpenCV

Το OpenCV (Open source Computer Vision library) είναι μία βιβλιοθήκη λογισμικού ανοιχτού κώδικα μηχανικής όρασης και μάθησης. Το OpenCV δημιουργήθηκε για να παρέχει μία κοινή υποδομή για εφαρμογές μηχανικής όρασης και για να επιταχύνει τη χρήση μηχανικής αντίληψης στα εμπορικά προϊόντα. Όντας ένα προϊόν αδειοδοτημένο κατά BSD, το OpenCV καθιστά εύκολο για τις επιχειρήσεις να χρησιμοποιούν και να τροποποιούν το κώδικα.

Ξεκινώντας επίσημα το 1999, το OpenCV έργο ήταν αρχικά μία πρωτοβουλία έρευνας της Intel, για να εξελίξει εφαρμογές μεγάλων απαιτήσεων επεξεργασίας από CPU, όντας μέρος μιας σειράς έργων συμπεριλαμβανομένων της ανίχνευσης ακτίνας (ray tracing) σε πραγματικό χρόνο και τρισδιάστατων τοίχων προβολής (3D display walls). Οι βασικοί συνεργάτες του έργου, περιελάμβαναν έναν αριθμό εμπειρογνωμόνων βελτιστοποίησης της Intel από την Ρωσία, καθώς και την ομάδα της Βιβλιοθήκης Απόδοσης της Intel (Intel's Performance Library).

Στα πρώιμα στάδια του OpenCV, οι στόχοι του έργου είχαν περιγραφεί ως εξής:

- Πρόοδος της έρευνας όρασης, παρέχοντας όχι μόνο ανοιχτό αλλά και βελτιστοποιημένο κώδικα για την βασική υποδομή όρασης. Δεν είναι απαραίτητη δηλαδή η «επανεφεύρεση του τροχού».
- Διάδοση γνώσης της όρασης, παρέχοντας μία κοινή υποδομή πάνω στην οποία μπορούν να βασιστούν οι προγραμματιστές, έτσι ώστε ο κώδικας να μπορεί να είναι πιο εύκολα αναγνώσιμος και μεταβιβάσιμος.
- Πρόοδος των εμπορικών εφαρμογών με βάση την όραση, δημιουργώντας φορητό κώδικα, με βελτιστοποιημένη απόδοση, ο οποίος θα διατίθεται δωρεάν, με τέτοια άδεια που δε θα χρειαζόταν να είναι ανοιχτού κώδικα ή δωρεάν οι ίδιες.

Η πρώτη alpha έκδοση του OpenCV παρουσιάστηκε στο κοινό κατά τη διάρκεια του IEEE Συνεδρίου πάνω στη Μηχανική Όραση και Αναγνώριση Προτύπων το 2000, και πέντε δοκιμαστικές (beta) εκδόσεις κυκλοφόρησαν μεταξύ 2001 και 2005.Η πρώτη 1.0 έκδοση κυκλοφόρησε το 2006. Στα μέσα του 2008, το OpenCV έλαβε εταιρική υποστήριξη από το Willow Garage, και τώρα είναι και πάλι υπό ενεργό ανάπτυξη. Η δοκιμαστική έκδοση 1.1 κυκλοφόρησε τον Οκτώβριο του 2008.

Η δεύτερη σημαντική κυκλοφορία του OpenCV πραγματοποιήθηκε τον Οκτώβριο του 2009.Το OpenCV 2 περιλαμβάνει σημαντικές αλλαγές στη διεπαφή C++, στοχεύοντας σε ευκολότερες, πιο ασφαλούς τύπου (type-safe) διατάξεις, καινούριες λειτουργίες, και καλύτερες εφαρμογές για τις ήδη υπάρχουσες, όσον αφορά την επίδοση (κυρίως σε συστήματα με πολλούς πυρήνες). Επίσημες κυκλοφορίες δημοσιεύονται πλέον κάθε έξι μήνες και η ανάπτυξη πλέον γίνεται από μία ανεξάρτητη Ρωσική ομάδα, που υποστηρίζεται από εμπορικές επιχειρήσεις.

Τον Αύγουστο του 2012,η υποστήριξη του OpenCV αναλήφθηκε από ένα μη κερδοσκοπικό ίδρυμα, το OpenCV.org, το οποίο αποτελείται από έναν προγραμματιστή και μία ιστοσελίδα για τους χρήστες. Η βιβλιοθήκη έχει πάνω από 2500 βελτιστοποιημένους αλγόριθμους, οι οποίοι περιλαμβάνουν ένα πλήρες σύνολο από κλασσικούς, αλλά και τελευταίας τεχνολογίας αλγόριθμους μηχανικής όρασης και μάθησης. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την ανίχνευση και την αναγνώριση προσώπων, τον προσδιορισμό αντικειμένων, τη κατάταξη ανθρώπινων ενεργειών σε βίντεο, τη παρακολούθηση των κινήσεων της κάμερας, τη παρακολούθηση κινούμενων αντικειμένων, την εξαγωγή τρισδιάστατων μοντέλων από αντικείμενα, τη παραγωγή τρισδιάστατων σημειακών νεφελωμάτων (point clouds) από στεροσκοπικές κάμερες, τη συρραφή εικόνων ώστε να παραχθεί μια εικόνα υψηλής ανάλυσης μιας ολόκληρης σκηνής, την ανεύρεση παρόμοιων εικόνων από μία βάση δεδομένων εικόνας, την αφαίρεση κόκκινου ματιού από φωτογραφίες που έχουν ληφθεί με φλας, τη παρακολούθηση των κινήσεων του ματιού, την αναγνώριση τοπίου και τη δημιουργία δεικτών προς επικάλυψη αυτού με προσαυξημένη πραγματικότητα και άλλα. Το OpenCV έχει πάνω από 47 χιλιάδες χρήστες κι ο εκτιμώμενος αριθμός μεταφορτώσεων ξεπερνά τα 7 εκατομμύρια. Η βιβλιοθήκη χρησιμοποιείται εκτενώς σε εταιρείες, ερευνητικές ομάδες και από κρατικούς φορείς. Μαζί με τις καθιερωμένες εταιρείες όπως η Google, το Yahoo, η Microsoft, η Intel, η IBM, η Sony, η Honda, και η Toyota που χρησιμοποιούν τη βιβλιοθήκη, υπάρχουν καινούριες εταιρείες όπως η Applied Minds, η VideoSurf, και η Zeitera, που κάνουν εκτεταμένη χρήση του OpenCV. Οι αναπτυσσόμενες χρήσεις του OpenCV καλύπτουν όλο το φάσμα από τη συρραφή streetview εικόνων, τον εντοπισμό εισβολών σε βίντεο παρακολούθησης στο Ισραήλ, τη παρακολούθηση εξοπλισμού ορυχείων στη Κίνα, βοηθώντας ρομπότ να περιηγηθούν και να συλλέξουν αντικείμενα, στην Willow Garage, τον εντοπισμό ατυχημάτων από πνιγμό σε πισίνες στην Ευρώπη, τη διαδραστική τέχνη σε Ισπανία και Νέα Υόρκη, τον έλεγχο διαδρόμων για συντρίμμια στην Τουρκία, την επιθεώρηση ετικετών προϊόντων σε εργοστάσια σε όλο τον κόσμο μέχρι και την ταχεία ανίχνευση προσώπων στην Ιαπωνία.

Διαθέτει διεπαφές C++, C, Python, Java και MATLAB και υποστηρίζει Windows, Linux, Android και Mac OS. Το OpenCV κλίνει περισσότερο προς τις εφαρμογές της όρασης σε πραγματικό χρόνο και εκμεταλλεύεται τις MMX και SSE οδηγίες όταν είναι διαθέσιμες. Οι πλήρως εξοπλισμένες διεπαφές CUDA και OpenCV αναπτύσσονται ενεργά αυτή τη στιγμή. Υπάρχουν πάνω από 500 αλγόριθμοι και περίπου δεκαπλάσιο πλήθος συναρτήσεων που συνθέτουν και υποστηρίζουν αυτούς τους αλγόριθμους. Το OpenCV είναι γραμμένο εγγενώς (natively) σε C++ και έχει μία πρότυπη διεπαφή που συνεργάζεται άψογα με υποδοχείς STL.

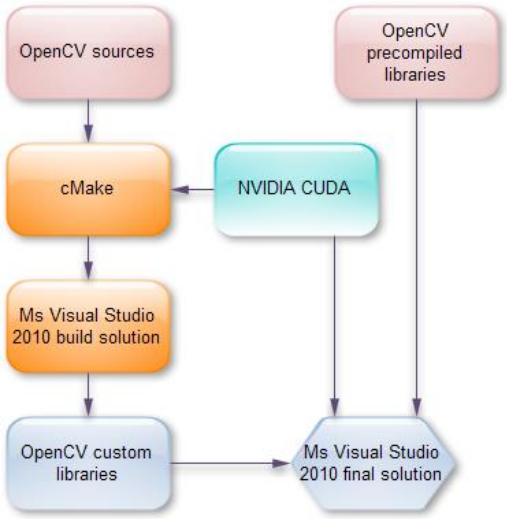
3. ΠΑΡΑΔΕΙΓΜΑ ΕΠΙΤΑΧΥΝΣΗΣ ΑΛΓΟΡΙΘΜΟΥ ΜΗΧΑΝΙΚΗΣ ΟΡΑΣΗΣ

3.1 Χρησιμοποιούμενα Εργαλεία

Για την ανάπτυξη της εφαρμογής αυτής, χρησιμοποιήθηκε ένα σύνολο εργαλείων που κρίθηκε ότι αντιπροσωπεύει την πρακτικότερη υλοποίηση για το δεδομένο πρόβλημα. Ως προγραμματιστικό περιβάλλον κυρίως χρησιμοποιήθηκε το Microsoft Visual Studio Professional 2010, λόγω της ευρείας υποστήριξης που το συνοδεύει σε εκπαιδευτικό υλικό, της αξιοπιστίας του, αλλά και της συμβατότητας που παρουσιάζει προς τα υπόλοιπα εργαλεία. Ως γλώσσα για την ανάπτυξη της εφαρμογής, πέρα από την χρήση της γλώσσας CUDA [103] επιλέχθηκε η C και C++ και μεταγλωττιστής ο παρεχόμενος από την Visual C++ της Microsoft (native, CLI, CLR).

Η εφαρμογή αποτελεί υλοποίηση του αλγορίθμου SIFT με χρήση της γλώσσας CUDA, για την επιτάχυνση των μαθηματικών υπολογισμών, της επεξεργασίας των εικόνων και των λοιπών δεδομένων που διαχειρίζεται. Ο αλγόριθμος βασίστηκε σε framework γραμμένο με την γλώσσα C που υλοποιεί την ακολουθιακή μορφή του αλγορίθμου [104]. Το framework αυτό έχει γραφτεί το 2011 στο πολυτεχνικό πανεπιστήμιο του Eindhoven στην Ολλανδία από τους Zhenyu Ye και Dongrui She.

Το προγραμματιστικό API Επεξεργασίας Εικόνας και Μηχανικής Όρασης που χρησιμοποιήθηκε μερικώς, είναι το OpenCV, έκδοση 2.4.9 και έγινε ανάμικτη χρήση των προ-μεταγλωττισμένων βιβλιοθηκών και του πηγαίου κώδικα του πακέτου, προκειμένου να αντιμετωπιστούν αναπάντεχα bugs που απαντήθηκαν κατά την ανάπτυξη της εφαρμογής. Για την μεταγλώττιση της βιβλιοθήκης, χρησιμοποιήθηκε το λογισμικό CMake και στην συνέχεια το Visual Studio.



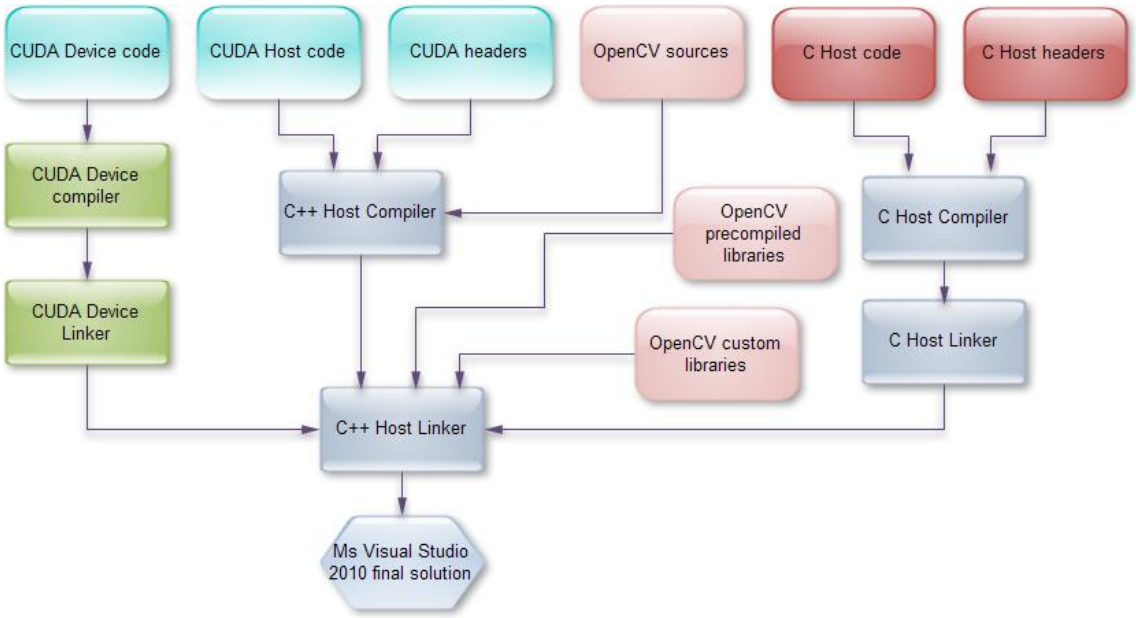
3-i Διαδικασία εγκατάστασης του OpenCV

Για την εξαγωγή γραφημάτων και την διενέργεια θεωρητικών μαθηματικών υπολογισμών, σε διάφορα στάδια της εργασίας, έγινε χρήση των λογισμικών Matlab 11 και Microsoft Excel. Οι εκδόσεις της γλώσσας CUDA μέχρι 6.5 είναι συμβατές με την εφαρμογή αυτή.

Ο υπολογιστής που αναπτύχθηκε και δοκιμάστηκε η εφαρμογή είναι τεχνολογίας 2010. Φέρει μητρική ASUS P6T με 12GB κύρια μνήμη σε διάταξη Triple Channel. Ο επεξεργαστής είναι Intel core i7 έκδοση 930 με ταχύτητα επεξεργαστή CPU 2.8 GHz. Η κάρτα γραφικών που χρησιμοποιήθηκε είναι NVIDIA GTS 450 αρχιτεκτονικής Fermi με 192 CUDA cores, Direct3D API έκδοσης 11. Η ταχύτητα του επεξεργαστή GPU είναι 1566 MHz. Η μνήμη γραφικών είναι τεχνολογίας GDDR5 με μέγεθος 1GB, 128 bit διεπαφή και εύρος ζώνης 57,73GB/s. Είναι εγκατεστημένη σε δίαυλο PCI Express x16 2^{ης} γενιάς.

Σε δοκιμές με λήψη βίντεο πραγματικού χρόνου χρησιμοποιήθηκαν τυπικές webcam επιλεγμένες εμπειρικά για τον μικρό χρόνο καθυστέρησης μετάδοσης της εικόνας στο λειτουργικό σύστημα. Μία εξ αυτών είναι η κάμερα που παρήγαγε η SONY για την πλατφόρμα Play Station 3. Ο λόγος που επιλέχθηκε είναι ο υψηλός ρυθμός μετάδοσης (frame rate: 60 Hz) επαρκούς ανάλυσης (resolution 640*480 pixel) εικόνας με χαμηλή καθυστέρηση. Τα χαρακτηριστικά αυτά ανταποκρίνονται στις εφαρμογές Μηχανικής Όρασης, που η ίδια η εταιρία παράγει και η τιμή της κάμερας είναι προσιτή σε σχέση με επαγγελματικές κάμερες Ρομποτικής ή Μηχανικής Όρασης της αγοράς, που έχουν παραπλήσια χαρακτηριστικά. Για την αξιοποίησή της στην εφαρμογή που αναπτύχθηκε, έγινε χρήση ιδιότυπου προγράμματος οδήγησης, που έχει αναπτυχθεί από την κοινότητα ανοιχτού κώδικα, καθώς η εταιρία δεν παρέχει υποστήριξη για οποιαδήποτε χρήση της εκτός του προβλεπόμενου συστήματος που συνοδεύει. Οι στατικές εικόνες υψηλής ανάλυσης που χρησιμοποιήθηκαν ως δείγματα για την εξαγωγή στατιστικών, είναι σε ορισμένες περιπτώσεις εταιρικά και άλλα λογότυπα υψηλής ανάλυσης, δείγματα λήψεων των φωτογραφικών μηχανών της FUJI που λήφθηκαν από τον επίσημο ιστότοπο της εταιρίας. Παρουσιάζουν χρωματικά και ποιοτικά χαρακτηριστικά με αρκετή ποικιλία προκειμένου να αποδώσουν όσο πιο πιστά την στατιστική πραγματικότητα για τους αλγόριθμους που τις επεξεργάστηκαν. Για την μαζική μετατροπή των φωτογραφιών σε διαφορετικές, μικρότερες αναλύσεις, έγινε χρήση του λογισμικού FastStone Photo Resizer για την πληθώρα αλγορίθμων μετατροπής που παρέχει, καθώς και τις δυνατότητες μαζικής μετονομασίας των παραγόμενων αρχείων. Οι γραφικές αναπαραστάσεις και τα λογικά, διαγράμματα δημιουργήθηκαν με το λογισμικό Edraw Flowchart. Τα στατιστικά δεδομένα προκύπτουν μετά από επεξεργασία των εξαγόμενων δεδομένων από την εκτέλεση της εφαρμογής στο λογισμικό Microsoft Excel.

Η ανάπτυξη του κώδικα έγινε σε γλώσσα C και CUDA C [105] με σκοπό να γίνει πιο χαμηλού επιπέδου διαχείριση της μνήμης, αποφεύγοντας αυτοματοποιημένες δομές της C++ όπως τα vectors και οι κλάσεις. Το σύστημα εισαγωγής και εξαγωγής εικόνας που παρείχε η ακολουθιακή έκδοση αντικαταστάθηκε από ιδιότυπες συναρτήσεις αναδίπλωσης (wrappers) του OpenCV, του οποίου όμως οι συναρτήσεις, αν και παρέχουν συμβατότητα με C, είναι γραμμένες σε gnu C++. Παρομοίως αντικαταστάθηκε το σύστημα χρονομέτρησης με αυτό του OpenCV για την επιπρόσθετη ευελιξία του. Ο κορμός του κώδικα εξακολουθεί να είναι το ακολουθιακό πρόγραμμα, προκειμένου να είναι εφικτή η κλήση της εκτέλεσης του αλγορίθμου εξολοκλήρου ή και τμηματικά, είτε σε ακολουθιακή είτε σε παράλληλη μορφή. Αυτό εξυπηρέτησε στην σύγκριση και αντιπαραβολή των μετρήσεων της κάθε εκτέλεσης και την μαζικοποίηση και αυτοματοποίηση της εξαγωγής των μετρήσεων αυτών, καθώς και των παραγόμενων στατιστικών δεδομένων τους. Η μίξη αυτή είχε βέβαια ως παρενέργεια την αύξηση της προγραμματιστικής πολυπλοκότητας του συνολικού έργου, καθώς για την πλήρη μεταγλώττιση του γίνεται χρήση του μεταγλωττιστή της CUDA ο οποίος τηρεί πρότυπο μεγαλύτερο του C99 για τον κώδικα Host C και C++ των συναρτήσεων που βρίσκονται στα *.cu αρχεία. Ο κορμός όντας γραμμένος σε αρχεία *.c και με αρχεία κεφαλίδων (headers) *.h, καλείται να μεταγλωττιστεί από το Visual Studio, που παρέχει συμβατότητα με πρότυπο C90 αντί του C99 της CUDA. Η διασύνδεση των αντικειμενικών αρχείων *.obj που παρήχθησαν γίνεται από τον Linker του Visual Studio, κατόπιν κλήσης του από τον μεταγλωττιστή της CUDA. Αυτό έχει ως παρενέργεια την αποκλειστική κλήση του Linker για αντικειμενικά αρχεία γραμμένα σε C++. Για την παράκαμψη αυτού του προβλήματος, οι συναρτήσεις στα CUDA αρχεία, οι κεφαλίδες καθώς και οι καθολικές (global) μεταβλητές που εμπεριέχονται σε αυτές, πρέπει να εμπερικλείονται σε προεπεξεργαστικές οδηγίες extern”C”{}.



3-ii Σύνδεση εργαλείων της εφαρμογής

Στο επόμενο κεφάλαιο αναλύεται η λειτουργία του αλγορίθμου και παρουσιάζονται τα ενδιάμεσα αποτελέσματα των βημάτων του, σε δοκίμιο την παραδοσιακά καθιερωμένη εικόνα του κλάδου της Επεξεργασίας Εικόνας, "lena.jpg".

3.2 Ο αλγόριθμος SIFT (Scale Invariant Feature Transform)

Η αναγνώριση αντικειμένου σε πραγματικά τοπία χρειάζεται τοπικά χαρακτηριστικά εικόνας τα οποία δεν επηρεάζονται από την αταξία του περιβάλλοντος και την μερική επικάλυψη. Το 1999 ο David G. Lowe του πανεπιστημίου University of British Columbia παρουσίασε στην σχετική δημοσίευσή του έναν καινοτόμο αλγόριθμο για την αναγνώριση αντικειμένων βασισμένο στην ανίχνευση τοπικών χαρακτηριστικών και αργότερα το 2004 εισήγαγε ορισμένες βελτιώσεις και επεκτάσεις της λειτουργικότητας του. Ο αλγόριθμος αυτός ονομάστηκε SIFT (Scale Invariant Feature Transform) και προστατεύεται με πατέντα ως πνευματική ιδιοκτησία για εμπορική χρήση. Ως αποτέλεσμα αυτών των δύο εργασιών, γεννήθηκε μία ολόκληρη γενιά αλγορίθμων αναγνώρισης αντικειμένου που χρησιμοποιούν τις ίδιες αρχές και φιλοσοφία, προσπαθώντας να βελτιώσουν τμηματικά τα επιμέρους στοιχεία του ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής που εξυπηρετούν. Οι περισσότεροι από αυτούς όμως καταλήγουν να θυσιάζουν μία αρετή του αλγορίθμου για να βελτιώσουν μία άλλη. Δεδομένου του ότι ο αλγόριθμος αυτός είναι μαθηματικά περίπλοκος, φέρει μεγάλο υπολογιστικό φόρτο και συνεπώς η εκτέλεσή του στην θεμελιώδη του μορφή είναι σχετικά αργή. Έτσι οι απόγονοί του [106], συχνά τείνουν να βελτιώνουν παράγοντες που τον καθιστούν αργό θυσιάζοντας είτε μία από τις πτυχές του, ή την γενική αξιοπιστία και σταθερότητά του.

Ο αλγόριθμος SIFT ικανοποιεί τα εξής αμετάβλητα κριτήρια της αναγνώρισης αντικειμένου:

- Μέγεθος (Scale Invariance)
- Περιστροφή (Rotation Invariance)
- Φωτεινότητα (Brightness Invariance)
- Προοπτική (Viewpoint Invariance)
- Μερική Επικάλυψη (Occlusion Invariance)

Μέχρι την εμφάνιση του αλγορίθμου, οι παραδοσιακές μέθοδοι επιχειρούσαν να αναγνωρίσουν αντικείμενα με την χρήση ανιχνευτών γωνιών όπως ο αλγόριθμος του Harris, σχημάτων και ακμών. Ο SIFT βασίζεται σε συνδυασμό ορισμένων από αυτές τις τεχνικές με σκοπό την εξαγωγή σημείων ενδιαφέροντος στην αναλυόμενη εικόνα. Τα σημεία αυτά ονομάζονται "keypoints" και ο συνδυασμός τους είναι ικανός να παράγει ένα αρκετά μοναδικό αναγνωριστικό για το εικονιζόμενο αντικείμενο. Τα σημεία αυτά εξάγονται σε μορφή λίστας που αποτελεί τον «περιγραφέα» της εικόνας και είναι ιδιαίτερα εύχρηστα σε εφαρμογές Μηχανικής Μάθησης λόγω του μικρού όγκου τους και της περιεκτικότητας πληροφορίας που αντιπροσωπεύουν. Αρκεί η αντιστοίχιση ελάχιστων από αυτά με την βάση μοντέλων για να επιβεβαιωθεί με μεγάλη ακρίβεια η ταυτοποίηση ενός από τα αντικείμενα της βάσης, στην πλειονότητα των εφαρμογών μηχανικής όρασης.

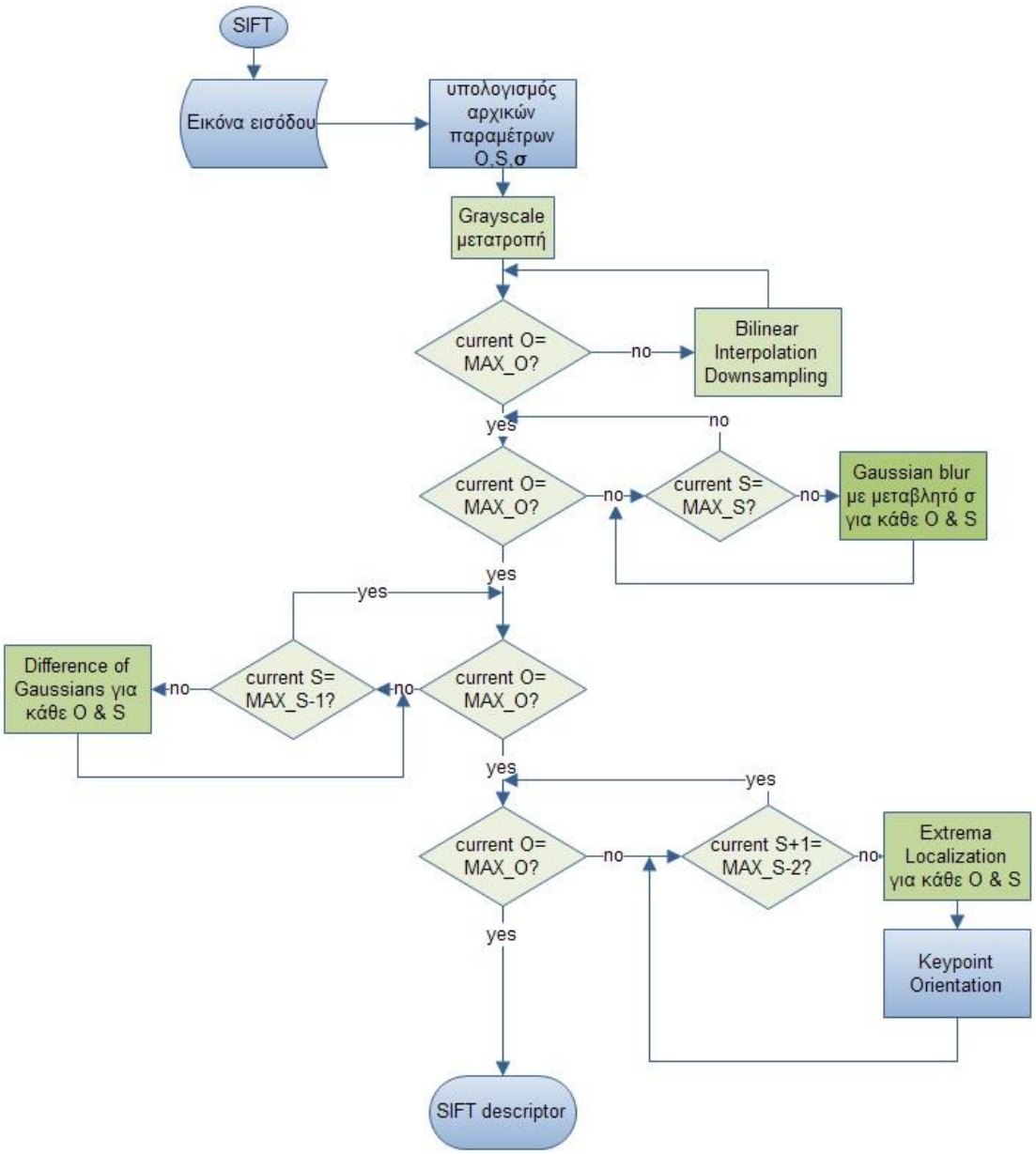
Ο αλγόριθμος έχει μορφή αρθρωτή και μπορεί να περιγραφεί στα εξής βήματα.

- Μετατροπή της εικόνας σε μονοχρωματική στην περίπτωση που δεν είναι ήδη κατά την πρόσληψή της.
- Διπλασιασμός του μεγέθους της αρχικά και θόλωση για να εξομαλυνθεί το αποτέλεσμα του διπλασιασμού.

Τα δύο πρώτα αυτά βήματα είναι προαιρετικά καθώς το πρώτο εξαρτάται από τα δεδομένα εισόδου της εφαρμογής και το δεύτερο αφαιρέθηκε ως σύσταση στην αναθεώρηση της μεθόδου από τον δημιουργό το 2004.

- Δημιουργία πυραμίδας εικόνων διαφορετικού μεγέθους με βάση την αρχική και χρήση αλγορίθμου «διγραμμικής παρεμβολής» (bilinear interpolation). Κάθε νέα εικόνα έχει λόγο πλευράς 1/2 από την προηγούμενη. Γι αυτό τον λόγο οι εικόνες αυτές αποκαλούνται «οκτάβες» και η πυραμίδα εικόνων που παράγεται ονομάζεται «βάση οκτάβων» (octave base). Η διαδικασία αυτή γίνεται προκειμένου το αποτέλεσμα του SIFT να φέρει την παραγόμενη πληροφορία ανεξαρτήτως του μεγέθους του ανιχνευόμενου αντικειμένου. Εφόσον ένα σημείο ενδιαφέροντος είναι ανιχνεύσιμο σε πολλαπλές οκτάβες, φέρει μεγαλύτερή βαρύτητα ως αναγνωριστικό του αντικειμένου.
- Δημιουργία «χώρου κλίμακας» (scale-space) με βάση την εικόνα κάθε οκτάβας και την συνέλιξή της με ένα φίλτρο Gauss αυξανόμενου $\boldsymbol{\sigma}$ σε κάθε επίπεδο θόλωσης. Η λειτουργία αυτή πραγματοποιείται για να αφαιρεθούν λεπτομέρειες και θόρυβος από κάθε επίπεδο της πυραμίδας εικόνων όπως αποδεικνύεται από την μαθηματική θεωρία σχετικά με το χαμηλοπερατό φίλτρο θόλωσης Gauss .
- Δημιουργία πυραμίδων [25] εικόνων με διαφορικές συνελιγμένες εικόνες του προηγουμένου βήματος. Ο σκοπός του βήματος αυτού είναι η προσεγγιστική υλοποίηση πολλαπλών φίλτρων Laplacian Of Gaussian (LoG) οι οποίες όμως θα ήταν χρονικά μη συμφέρουσες λόγω υπολογιστικού φόρτου. Γι αυτό τον λόγο ο Lowe, χρησιμοποιώντας ένα μαθηματικό τέχνασμα, πρότεινε την χρήση διαφορών μεταξύ των επιπέδων της θόλωσης Gauss για το προσεγγιστικά ίδιο αποτέλεσμα με το φίλτρο LoG. Η χρήση των παραγώγων $2^{ου}$ βαθμού Laplace που υλοποιεί το φίλτρο αυτό εξυπηρετεί μεταξύ άλλων και στην αμεταβλητότητα του αποτελέσματος σε σχέση με την φωτεινότητα καθώς ο λόγος των διαφορικών $2^{ου}$ βαθμού δεν επηρεάζεται σημαντικά έως καθόλου από την παράμετρο της φωτεινότητας.
- Εντοπισμός τοπικών ακρότατων. Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, το φίλτρο LoG χρησιμοποιείται για την εξαγωγή περιγραμμάτων και ακμών από εικόνες. Για την αναγνώριση όμως με χρήση SIFT, τα σημεία που εντοπίζονται πάνω σε τέτοια χαρακτηριστικά θεωρούνται αναξιόπιστα και συνεπώς πρέπει να απαλειφούν. Για να επιτευχθεί αυτό επιλέγονται μόνο σημεία που είναι ακρότατα σε μία εντοπισμένη περιοχή της κατακερματισμένης εικόνας και αποκλείονται οι ακμές και τα όρια με χρήση μεθόδου, παρόμοιας με τον ανιχνευτή γωνιών του Harris. Ακολούθως συγκρίνονται με τα αντίστοιχα σημεία των δύο γειτονικών επιπέδων θόλωσης. Σε αυτή την φάση και με χρήση αναπτυγμάτων Taylor απορρίπτονται τα σημεία χαμηλής αντίθεσης, με απλή αριθμητική σύγκρισή τους με μία εμπειρική σταθερά και αν επιβεβαιωθούν ως επικρατούντα, καθιερώνονται ως SIFT keypoints.
- Προσδιορισμός του προσανατολισμού των σημείων ενδιαφέροντος. Αυτό το βήμα εξασφαλίζει την αμεταβλητότητα του αλγορίθμου στην περιστροφή του αντικειμένου και αποτελεί προσθήκη του 2004 στον αρχικό αλγόριθμο του 1999. Τα εναπομείναντα σημεία επανεξετάζονται για να διαπιστωθεί η γωνία της διαβάθμισής τους και η σχετική τοποθεσία τους σε συνδυασμό με τον προσανατολισμό τους τα καθιστά πλέον ικανά SIFT keypoints και εισάγονται στο διάνυσμα του περιγραφέα (descriptor).

Στο ακόλουθο σχήμα φαίνεται το απλοποιημένο διάγραμμα ροής του συνολικού αλγορίθμου SIFT.



3-iii Διάγραμμα ροής: Αλγόριθμος SIFT

3.2.1 Αρχικοί υπολογισμοί και προετοιμασία

Ακολούθως, θα αναλυθούν τα βήματα αυτά με επεξήγηση των μαθηματικών διαδικασιών και των προγραμματιστικών μεθόδων που τις υλοποιούν.

Η διαδικασία της μετατροπής μίας έγχρωμης εικόνας σε μονοχρωματική είναι σχετικά απλή. Η έγχρωμη εικόνα συνήθως ακολουθεί κάποιο τριχρωματικό πρότυπο όπου συνδυάζοντας τιμές φωτεινότητας των τριών θεμελιωδών χρωμάτων ή καναλιών, με τους αντίστοιχους συντελεστές για κάθε pixel προκειμένου να παραχθεί η ψευδαίσθηση της διατήρησης της φωταύγειας. Έτσι για κάθε pixel με συντεταγμένες x, y , r συντελεστή φωταύγειας του κόκκινου, g συντελεστή φωταύγειας του πράσινου, b συντελεστή φωταύγειας του μπλε και τα τρία κανάλια (εικόνες) των χρωμάτων κόκκινο, πράσινο και μπλε, έχουμε:

$$Gray_{(x,y)} = Red_{(x,y)} * r + Green_{(x,y)} * g + Blue_{(x,y)} * b$$

Στην υλοποίηση αυτού του βήματος σε παράλληλη μορφή με γλώσσα CUDA έγινε αξιοποίηση της μεγάλης παραλληλίας των υπολογισμών εκτελώντας την πράξη για όλα τα pixel παράλληλα έχοντας ως είσοδο τις τρεις εικόνες (πίνακες) με τις τιμές των χρωμάτων και έξοδο τον πίνακα της μονοχρωματικής εικόνας.

Για την παραγωγή του scale-space ο Lowe προτείνει έναν θεωρητικά επαρκή αριθμό 4 οκτάβων και 5 επιπέδων θόλωσης ανά οκτάβα. Στην πράξη όμως τα καλύτερα αποτελέσματα επιτυγχάνονται για μεγάλες εικόνες. Αυτό συμβαίνει επειδή εξοικονομείται υπολογιστική ισχύς από την μη εκτέλεση περιττών επαναληπτικών διεργασιών στις μικρές. Πειραματικά αποδεικνύεται ότι είναι καλύτερη η δυναμική επιλογή των αριθμών αυτών βάσει του μεγέθους της εικόνας, επιλέγοντας την μεγαλύτερη διάσταση από τις 2 και υπολογίζοντας το ακέραιο μέρος του λογαρίθμου βάσης 2 για την διάσταση αυτή, προκειμένου να αποφασίσουμε για τον αριθμό των οκτάβων O. Έτσι έχουμε:

$$O = log2[maximum(Width, Height)]$$

Επίσης ένα μέγιστο πλήθος $S = 6$ θολώσεων κρίνεται εμπειρικά επαρκές για μεγαλύτερα μεγέθη και ταυτόχρονα λόγω των σμικρύνσεων δεν επιβαρύνει υπολογιστικά τις μικρές εικόνες σε σημαντικό βαθμό.

Στην συνέχεια υπολογίζεται η σταθερά $\boldsymbol{\sigma}$ συναρτήσει μίας αρχικής τιμής της $\boldsymbol{\sigma_0} = 1.6$ την οποία προτείνει εμπειρικά ο Lowe, ενώ πολλές υλοποιήσεις προτιμούν τιμή $\sqrt{2}$ στην θέση της, για κάθε οκτάβα $o < O$ και επίπεδο θόλωσης $s < S$ του scale-space.

$$\boldsymbol{\sigma}_{(o,s)_{(x,y)}} = \sigma_{0*} 2^{\frac{s}{S-3}}$$

Η σταθερά αυτή πρέπει να είναι για κάθε επίπεδο πολλαπλάσια του προηγουμένου επιπέδου με οποιονδήποτε σταθερό λόγο $k > 0$ και είναι καθοριστική για τον προσδιορισμό της κατανομής Gauss και άλλων μεγεθών που χρησιμοποιούνται στον αλγόριθμο. Συνήθως η σταθερά k τίθεται ίση με $\sqrt{2}$.

$$\boldsymbol{\sigma_n} = \boldsymbol{\sigma_0} + \boldsymbol{\sigma_0} * n * k,$$
$$k = \sqrt{2}$$

3.2.2 Διγραμμική Παρεμβολή (Bilinear Interpolation)

Έχοντας πλέον υπολογίσει τις διάφορες παραμέτρους του αλγορίθμου, το πρώτο ουσιαστικό βήμα είναι η δημιουργία της πυραμίδας εικόνων με βηματική σμίκρυνσή τους σε οκτάβες. Αυτό στην περίπτωση του SIFT συνιστά τον υπολογισμό του μέσου όρου έντασης γειτονικών pixel της μεγαλύτερης εικόνας και την ανάθεσή του στην μικρότερη.

Έτσι για $o < O$ και συντεταγμένες του κάθε pixel x, y έχουμε:

$$octave_{o_{(x,y)}} = \frac{octave_{o-1_{(x,y)}} + octave_{o-1_{(x+1,y)}} + octave_{o-1_{(x,y+1)}} + octave_{o-1_{(x+1,y+1)}}}{4}$$

Στην υλοποίηση αυτού του βήματος σε παράλληλη μορφή με γλώσσα CUDA έγινε αξιοποίηση της μεγάλης παραλληλίας που παρουσιάζει η συνάρτηση αυτή πραγματοποιώντας παράλληλο υπολογισμό για κάθε pixel της παραγόμενης εικόνας με δεδομένα εισόδου την εικόνα του προηγούμενου μεγέθους.

Το αποτέλεσμα της συνάρτησης αυτής, εφαρμοσμένη στην εικόνα "lena.jpg" είναι η παραγωγή των ακόλουθων εικόνων οι οποίες αποτελούν την πυραμίδα οκτάβων.

3-iv Βάση Οκτάβων (Octave Base)

3.3.3 Χώρος Κλίμακας (Scale Space)

Σε αυτό το βήμα, αρχικά υλοποιείται ο υπολογισμός της Gauss κατανομής για τις διάφορες τιμές της παραμέτρου $\boldsymbol{\sigma}$ που υπολογίστηκαν στην αρχή του προγράμματος. Η κατανομή πραγματοποιείται σε μία διάσταση, καθώς είναι υπολογιστικά πιο αποδοτικό να πραγματοποιηθεί η συνέλιξη των δύο διαστάσεων σε δύο σαρώσεις μονοδιάστατων συνελικτικών παραθύρων με κάθετη εφαρμογή σάρωσης μεταξύ τους. Δεδομένου του ότι ο Lowe δεν κάνει αναφορά στο μέγεθος του συνελικτικού παραθύρου, οι διάφορες υλοποιήσεις του αλγορίθμου έχουν την ελευθερία να το προσδιορίσουν ανάλογα με τις απαιτήσεις ποιότητας της θόλωσης Gauss έναντι του υπολογιστικού φόρτου και χρόνου εκτέλεσης που απαιτείται. Συνήθως σε εφαρμογές επεξεργασίας εικόνας, η λειτουργία της θόλωσης Gauss βασίζεται σε κατανομές που επαρκούν για το 99.7% των τιμών μέσα στο πεδίο επίδρασης $\pm 3\boldsymbol{\sigma}$**.** Στην συγκεκριμένη εφαρμογή, θέτουμε την ακτίνα του παραθύρου gR ως εξής:

$$\left.\begin{array}{ll} \forall\sigma < \frac{1}{4}, & gR = 1 \\ \forall\sigma > \frac{1}{4}, & gR = 4\sigma \end{array}\right\} gW = 2gR + 1$$

Και στην συνέχεια υπολογίζεται η κατανομή Gauss για κάθε τιμή i στο πεδίο αυτό.

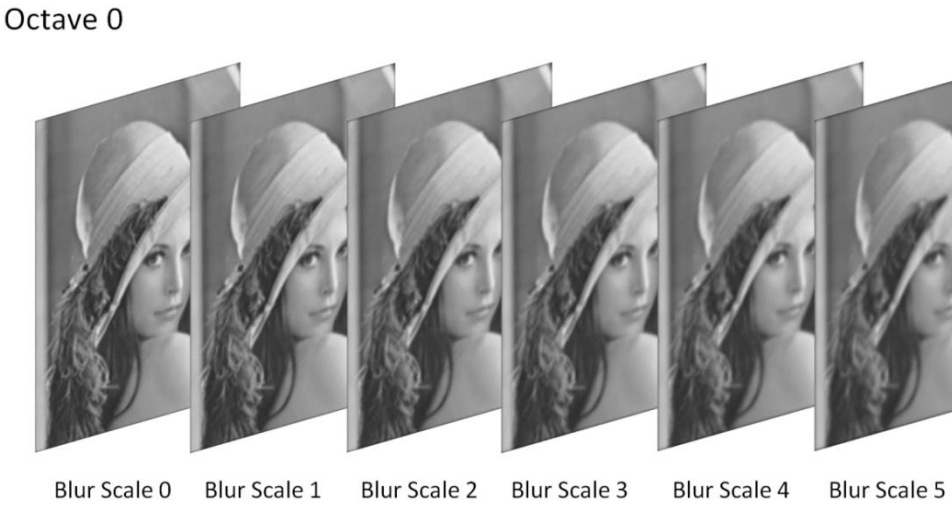
$$G_i = e^{-\frac{(i-gR)^2}{2}}$$

Συνελίσσοντας την εικόνα με δύο περάσματα κάθετα μεταξύ τους του παραθύρου αυτού, λαμβάνουμε την επεξεργασμένη εικόνα του τελεστή θόλωσης Gauss:

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

Στην παράλληλη υλοποίηση του τμήματος αυτού του αλγορίθμου παρουσιάστηκε ο μεγαλύτερος δείκτης επιτάχυνσης με σημαντικά μεγαλύτερες τιμές για τις μεγαλύτερες εικόνες. Ο λόγος που συνέβη αυτό είναι ότι η μεγάλη παραλληλισιμότητα της επεξεργασίας κάθε pixel σε συνδυασμό με το μεγάλο πλήθος μαθηματικών υπολογισμών για το κάθε ένα από αυτά καθόρισε σημαντική εξοικονόμηση χρόνου έναντι της σειριακής εκτέλεσης.

Το αποτέλεσμα της συνάρτησης αυτής, εφαρμοσμένη στην εικόνα "lena.jpg" είναι η παραγωγή των ακόλουθων εικόνων οι οποίες αποτελούν τις πυραμίδες του χώρου κλίμακας (Scale Space).



3-ν Χώρος Κλίμακας Gauss (Gaussian Scale Space)

3.3.4 Προσεγγίσεις Laplacian of Gaussian

Σε αυτό το βήμα, προσομοιώνεται η λειτουργία ενός φίλτρου Laplacian of Gaussian προκειμένου να τονιστούν οι ακμές και οι γωνίες της εικόνας καθώς αποτελούν καλή βάση για την εύρεση σημείων ενδιαφέροντος. Για μία εικόνα προεπεξεργασμένη με Gauss $f_{(x,y)}$ δηλαδή, το επιθυμητό αποτέλεσμα είναι:

$$L_{(x,y)} = \nabla^2 f_{(x,y)} = \frac{\partial^2 f_{(x,y)}}{\partial x^2} + \frac{\partial^2 f_{(x,y)}}{\partial y^2}$$

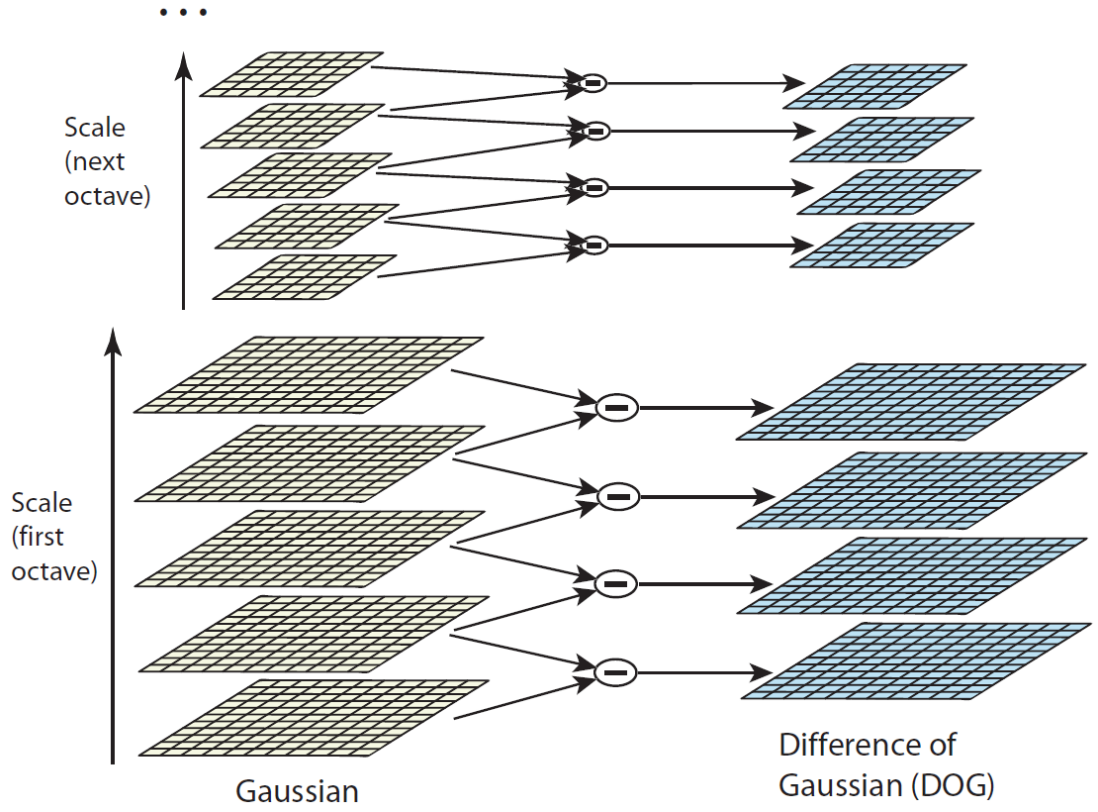
Και συνολικά το φίλτρο LoG είναι:

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

Προκειμένου να έχουμε πραγματική αμεταβλητότητα κλίμακας, πρέπει να απαλειφεί το $\boldsymbol{\sigma^2}$ από τον παρονομαστή καθώς αυτό υπαγορεύει το μέγεθος. Έτσι το LoG που εκφράζεται και ως $\nabla^2 G$ θα πρέπει να μετατραπεί σε αμετάβλητο κλίμακας, δηλαδή με μορφή $\sigma^2\nabla^2 G$. Πραγματοποιώντας «διαφορές των Gaussians» (Difference of Gaussians) (DoG) αντί για LoG εξασφαλίζουμε το αποτέλεσμα αυτό. Ο Lowe επίσης έχει αποδείξει πειραματικά ότι η αμεταβλητότητα μεγέθους/κλίμακας παράγει πιο αξιόπιστα σημεία ενδιαφέροντος. Το μειονέκτημα είναι ότι πέρα από το $\boldsymbol{\sigma^2}$ όμως, το αποτέλεσμα του DoG είναι επίσης πολλαπλασιασμένο με έναν τελεστή k-1 ο οποίος αναφέρθηκε στο προηγούμενο βήμα. Δεδομένου όμως του ότι στα πλαίσια του SIFT μας ενδιαφέρει μόνο η τοποθεσία και όχι η τιμή των ακρότατων (maxima), το μαθηματικό αυτό τέχνασμα είναι χρήσιμο προκειμένου να μειωθεί πολύ ο υπολογιστικός φόρτος και να βελτιωθεί η ταχύτητα εκτέλεσης του αλγορίθμου.

Στην πράξη ο αλγόριθμος πραγματοποιεί διαδοχικές αφαιρέσεις των γειτονικών εικόνων στο πεδίο του χώρου κλίμακας και παράγει πυραμίδες εικόνων πλήθους $O * (S-1)$.

Η υλοποίηση του βήματος αυτού επωφελείται σε μεγάλο βαθμό από την παραλληλία και όλες οι αφαιρέσεις μεταξύ 2 επιπέδων κάθε εικόνας εκτελούνται παράλληλα σε έναν επαναληπτικό βρόγχο.



3-vi Χώρος κλίμακας Gauss και Διαφορές των Gauss (DoG) από την εργασία του D. Lowe

Το αποτέλεσμα της συνάρτησης αυτής, εφαρμοσμένη στην εικόνα “lena.jpg” είναι η παραγωγή των ακόλουθων εικόνων οι οποίες αποτελούν τις πυραμίδες Difference of Gaussians.

Octave 0

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

Octave 1

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

Octave 2

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

Octave 3

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

Octave 4

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

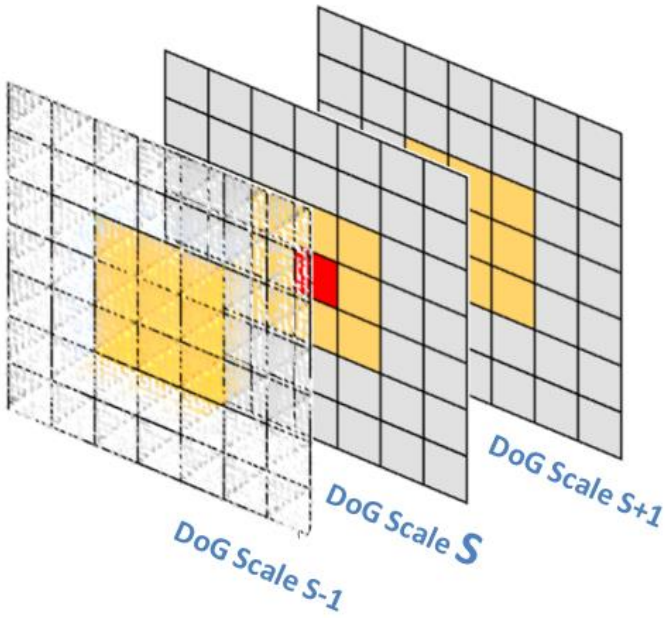
Octave 5

DoG Scale 0 DoG Scale 1 DoG Scale 2 DoG Scale 3 DoG Scale 4

3-vii Πυραμίδες Διαφορών Gauss (Difference of Gaussian Pyramids)

3.3.5 Εντοπισμός Τοπικών Ακρότατων (Extrema Localization)

Τα τοπικά ακρότατα ή extrema points (minima και maxima) στις εικόνες που έχουν παραχθεί από τις διαφορές εικόνων DoG, εντοπίζονται ελέγχοντας κάθε pixel σε σύγκριση με τα οκτώ γειτονικά του στην εικόνα που ανήκει, τα εννέα της αντίστοιχης τοποθεσίας στην υψηλότερη κλίμακα θόλωσης και τα εννέα της χαμηλότερης κλίμακας θόλωσης. Δηλαδή για κάθε pixel εικόνας DoG γίνονται $(9-1)+2*9=26$ έλεγχοι συνολικά. Αν το pixel παρουσιάζει ελάχιστη ή μέγιστη τιμή από τα συγκρινόμενα με αυτό, σημειώνεται ως τοπικό ακρότατο. Προκειμένου να γίνουν αυτές οι συγκρίσεις, αναγκαστικά απορρίπτονται το ανώτατο και το κατώτατο επίπεδο θόλωσης αφού δεν υπάρχουν και τα δύο γειτονικά επίπεδα που απαιτούνται για να συγκριθούν.



3-viii Εντοπισμός ακρότατων από περιοχή γειτονικών pixel και επιπέδων (extrema localization)

Η διαδικασία αυτή παράγει τα προσεγγιστικά ακρότατα του χώρου sub-pixel επειδή τα εξεταζόμενα pixel είναι αποτέλεσμα διαφορών και όχι τα αρχικά της εικόνας που επεξεργάζεται ο SIFT. Προκειμένου να υπολογιστούν τα sub-pixel maxima χρησιμοποιείται το ανάπτυγμα Taylor της εικόνας γύρω από το προσεγγιστικό keypoint. Μαθηματικά αυτό εκφράζεται ως εξής:

$$D_{(\boldsymbol{x})} = D + \left(\frac{\partial D}{\partial \boldsymbol{x}}\right)^T \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T \frac{\partial^2 D}{\partial^2 \boldsymbol{x}} \boldsymbol{x}\,, \qquad \text{όπου } \boldsymbol{x} = (x, y, \sigma)^T$$

Η τοποθεσία του ακρότατου καθορίζεται παραγωγίζοντας την συνάρτηση αυτή ως προς $\boldsymbol{x}$, και θέτοντας την στο μηδέν, δηλαδή:

$$\hat{\boldsymbol{x}} = -\frac{\partial^2 D}{\partial \boldsymbol{x}^2}^{-1} \frac{\partial D}{\partial \boldsymbol{x}}$$

Αφαιρώντας τις δύο παραπάνω εξισώσεις παράγεται μία αξιοποιήσιμη μορφή του ακρότατου $|D(\hat{\boldsymbol{x}})|$ που συγκρινόμενη με κάποια εμπειρική σταθερά που εκφράζει την αντίθεση της εικόνας στο σημείο αυτό μπορεί να απαλείψει τα σημεία χαμηλής αντίθεσης. Ο Lowe έθεσε την σταθερά αυτή αυθαίρετα πειραματικά στην τιμή 0.03. Στην υλοποίηση αυτής της εργασίας όμως έχει εμπειρικά επιλεγεί η τιμή 3 λόγω και του αριθμητικού συστήματος αναφοράς που χρησιμοποιείται. Η τιμή αυτή συγκρίνεται με την προαναφερθείσα αφαίρεση εξισώσεων που έχει την εξής μορφή:

$$D(\hat{\boldsymbol{x}}) = D + \frac{1}{2}\left(\frac{\partial D}{\partial \boldsymbol{x}}\right)^T \hat{\boldsymbol{x}}$$

Προκειμένου να απορριφθούν τα keypoints που συμπίπτουν με ακμές, θεωρείται θεμιτό να υπολογιστεί η Εσσιανή Μήτρα (Hessian Matrix) H για την $D(x, y, \sigma)$ και να θεωρηθεί ότι οι ιδιοτιμές της είναι ανάλογες προς την θεμελιώδη καμπυλότητα της $D(x, y, \sigma)$:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial y \partial x} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix}$$

Οι παράγωγοι υπολογίζονται λαμβάνοντας τις διαφορές των γειτονικών δειγματοληπτούμενων σημείων. Έστω α η ιδιοτιμή με το μεγαλύτερο μέγεθος και β με το μικρότερο. Τότε υπολογίζονται το σύνολο των ιδιοτιμών από το ίχνος της H και το γινόμενο τους από την ορίζουσα:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$Det(H) = D_{xx}D_{yy} - \left(D_{xy}\right)^2 = \alpha\beta.$$

Στην απίθανη περίπτωση όπου η ορίζουσα είναι αρνητική, οι καμπυλότητες έχουν διαφορετικά πρόσημα, το σημείο απορρίπτεται ως ακρότατο. Έστω r ο λόγος μεταξύ της μεγαλύτερης ιδιοτιμής και της μικρότερης, ώστε να ικανοποιεί την σχέση $\alpha = r\beta$, τότε έχουμε την ακόλουθη σχέση που εξαρτάται μόνο στον λόγο των ιδιοτιμών αντί για τις τιμές τους:

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta} = \frac{(r+1)^2}{r}$$

Το $\frac{(r+1)^2}{r}$ είναι ελάχιστο όταν οι δύο ιδιοτιμές είναι ίσες και αυξάνεται με το r. Ως εκ τούτου, για να ελεγχθεί η αναλογία των θεμελιωδών καμπυλοτήτων σε σύγκριση με ένα κατώφλι, αρκεί να ελέγξουμε:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

Ο Lowe προτείνει τιμή $r = 10$ ως κατώφλι για να αποφανθεί ο αλγόριθμος την φύση της ακμής ως μη γωνία.

3.3.6 Προσδιορισμός Προσανατολισμού Σημείων Ενδιαφέροντος (Orientation Assignment)

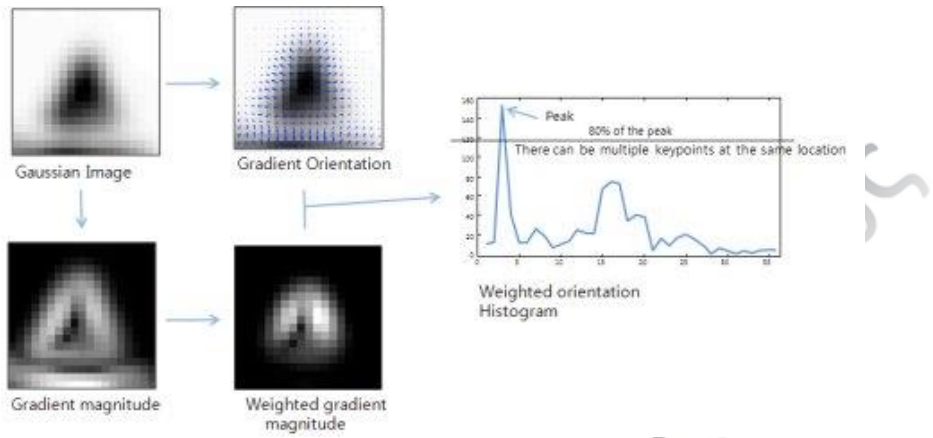
Το μέγεθος της διαβάθμισης αλλά και η κατεύθυνσή της υπολογίζονται στο βήμα αυτό. Μαθηματικά εκφράζονται με τρόπο παρόμοιο με τον τελεστή Sobel που έχει αναφερθεί σε προηγούμενο κεφάλαιο, ως εξής:

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

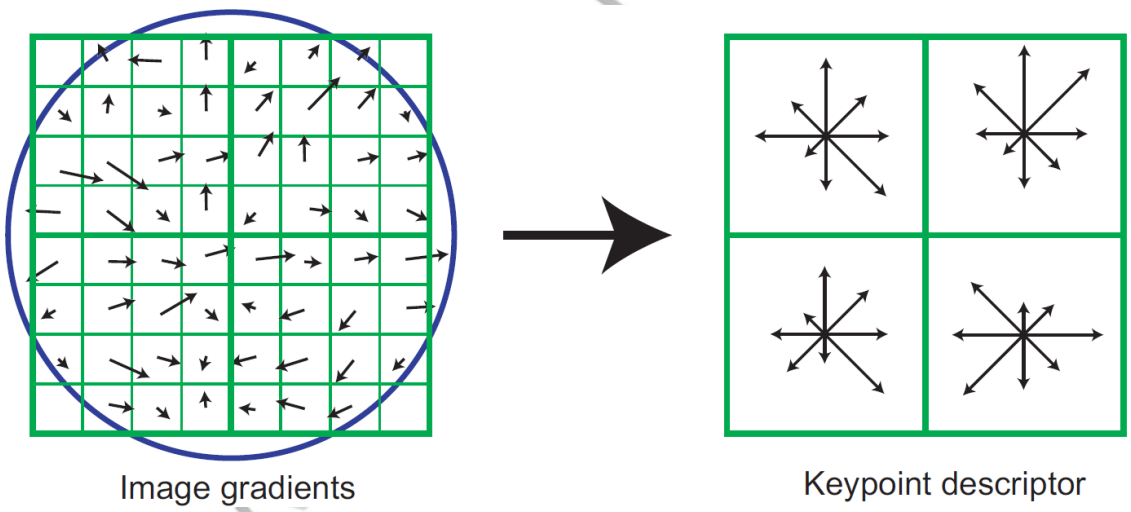
$$\theta(x, y) = \tan^{-1}\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

Το μέγεθος της διαβάθμισης υπολογίζεται με την βαρύτητα συντελεστών από το παράθυρο συνέλιξης Gauss, με σκοπό να μειώσει την επίδραση των διαβαθμίσεων μακριά από το κέντρο του. Ο Lowe πρότεινε την επεξεργασία των σημείων σε σχέση με τετράγωνο τμήμα της εικόνας και πλευράς 16 pixel, ενώ μετέπειτα υλοποιήσεις έχουν πραγματοποιήσει το ίδιο με πλευρά τετραγώνου 8 pixel, για εξοικονόμηση υπολογιστικών πόρων.

Στόχος είναι η δημιουργία ενός ιστογράμματος από τον προσανατολισμό της διαβάθμισης. Το ιστόγραμμα αυτό έχει 36 διακριτές τιμές οι οποίες καλύπτουν το σύνολο των 360^o με βήμα 10^o. Η δεσπόζουσα τιμή του ιστογράμματος (εντός του 80%) του καθολικού μέγιστου, αντιστοιχεί στην δεσπόζουσα κατεύθυνση της τοπικής διαβάθμισης. Η μέθοδος αυτή ονομάζεται Ιστόγραμμα Κλίσεων Διαβαθμίσεων ή Histogram Orientation Gradient (HOG) [4] και χρησιμοποιείται σε πληθώρα αλγορίθμων μηχανικής όρασης, συχνά σε μορφές βελτιστοποιημένες ως προς την ταχύτητα [3] και αναδιαμορφωμένες για τον εκάστοτε αλγόριθμο.



3-ix Ιστόγραμμα Κλίσεων Διαβαθμίσεων (HOG)



3-x Προσανατολισμός "περιγραφέα" SIFT από την εργασία του D.Lowe

Όταν ο τοπικός προσανατολισμός και η κλίμακα του keypoint έχουν προσδιοριστεί, μπορούν να εξαχθούν οι κλιμακωμένες και προσανατολισμένες περιοχές γύρω από το keypoint (16*16) για να σχηματίσουν τον «περιγραφέα» του χαρακτηριστικού (feature descriptor). Για να είναι αμετάβλητος ως προς την ένταση, ο περιγραφέας κανονικοποιείται σε μήκος μονάδας και για να είναι ανθεκτικός προς άλλες φωτομετρικές εναλλαγές, οι τιμές ψαλιδίζονται σε 0.2 και ακολούθως το εξαγόμενο διάνυσμα κανονικοποιείται εκ νέου σε μήκος μονάδας.

Ο περιγραφέας διαμορφώνεται από ένα δίανυσμα που περιέχει τις τιμές όλων των παραμέτρων του ιστογράμματος προσανατολισμού που αντιστοιχούν στα μήκη των βελών που απεικονίζονται στην αριστερή πλευρά της παραπάνω εικόνας. Στα δεξιά απεικονίζεται ένας πίνακας 2*2 με τα ιστογράμματα προσανατολισμού. Ωστόσο ο Lowe προτείνει ως πιο αξιόπιστη την χρήση αντίστοιχου πίνακα 4*4 ιστογραμμάτων με 8 «κάδους» προσανατολισμού έκαστο. Το τελικό αποτέλεσμα είναι ένα διάνυσμα 128 διαστάσεων (4*4*8) και είναι πλέον αξιοποιήσιμο από εφαρμογές ταυτοποίησης (matching) και μηχανικής όρασης. Ο ίδιος ο Lowe προτείνει χρήση γενικευμένου μετασχηματισμού Hough για την ταυτοποίηση των περιγραφέων σε μία μεγάλη βάση μοντέλων.

3.3 Η παράλληλη υλοποίηση σε CUDA

Σε όλα τα στάδια του αλγορίθμου που κρίθηκε σκόπιμο και εφικτό να παραλληλοποιηθούν χρησιμοποιείται η ακόλουθη στρατηγική. Επειδή οι ακολουθιακές συναρτήσεις παρουσίαζαν παραλληλία σε πολλαπλά επίπεδα, κρίθηκε ως βέλτιστη πρακτική να δοθεί προτεραιότητα στην υλοποίηση της παραλληλίας του βαθύτερου επιπέδου, δηλαδή του προγραμματιστικού πυρήνα που με επαναληπτικούς βρόγχους προσπελάζει τα pixel της εικόνας ένα προς ένα [1]. Ο kernel ρυθμίζεται ώστε να αξιοποιεί στο μέγιστο τις δυνατότητες της συγκεκριμένης κάρτας γραφικών, η οποία σε compute capability 2.0 παρέχει μέγιστο μέγεθος block ίσο με 1024 threads μονοδιάστατο, η 32*32 threads δισδιάστατο το οποίο τελικά και χρησιμοποιήθηκε. Το μέγεθος του grid ορίζεται ως δισδιάστατο και υπολογίζεται σε χρόνο εκτέλεσης βάσει των διαστάσεων της εκάστοτε εικόνας που καλείται να επεξεργαστεί. Επίσης για να αποφευχθεί η πιθανή εκτέλεση του kernel με μηδενικό πλήθος blocks, αλλά και να εξασφαλιστεί η πλήρης προσπέλαση της εικόνας ακόμα και αν το μέγεθός της δεν είναι ακέραια διαιρούμενο με την διάσταση του block, οι διαστάσεις του grid ορίζονται ως η αντίστοιχη διάσταση της εικόνας προσαυξημένη κατά την διάσταση του block-1 και ακέραια διαιρεμένη με την διάσταση αυτή.

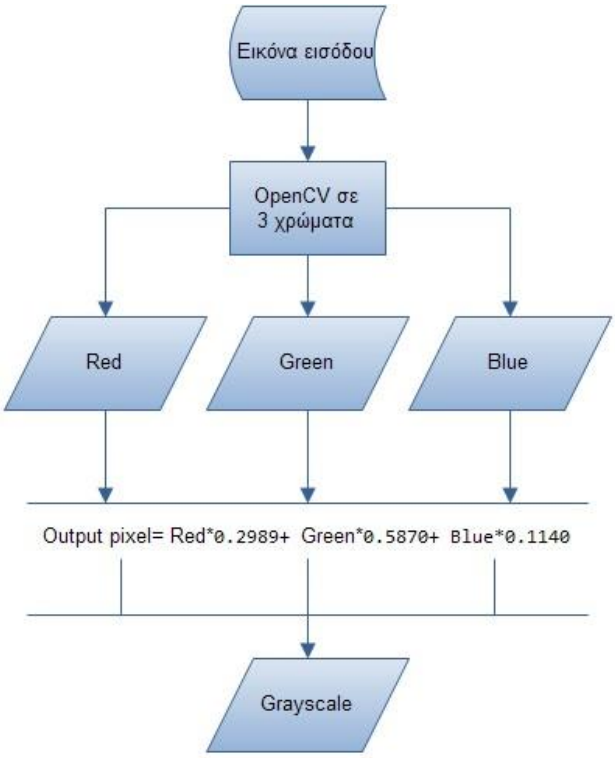
```
const dim3 block(32,32);
const dim3 grid((inW+block.x-1)/block.x,(inH+block.x-1)/block.y);
```

Σημειώνεται οι εικόνες σε επίπεδο κώδικα είναι «ισοπεδωμένες» (flattened). Αυτό σημαίνει ότι η προσπέλαση του κάθε pixel γίνεται με μονοδιάστατη διευθυνσιοδότηση με χρήση δεικτών. Δηλαδή οι συντεταγμένες ενός pixel (x, y) προσπελάζονται ως $(y * Width + x)$. Γι αυτό τον λόγο η πρόσβαση του κάθε thread στο αντίστοιχο pixel που καλείται να επεξεργαστεί ή να διαβάσει πραγματοποιείται με χρήση των δύο διαστάσεων του block και του thread, δηλαδή blockIdx.x, blockIdx.y, threadIdx.x και threadIdx.y ως εξής:

```
const int x=blockIdx.x*blockDim.x+threadIdx.x;
const int y=blockIdx.y*blockDim.y+threadIdx.y;
```

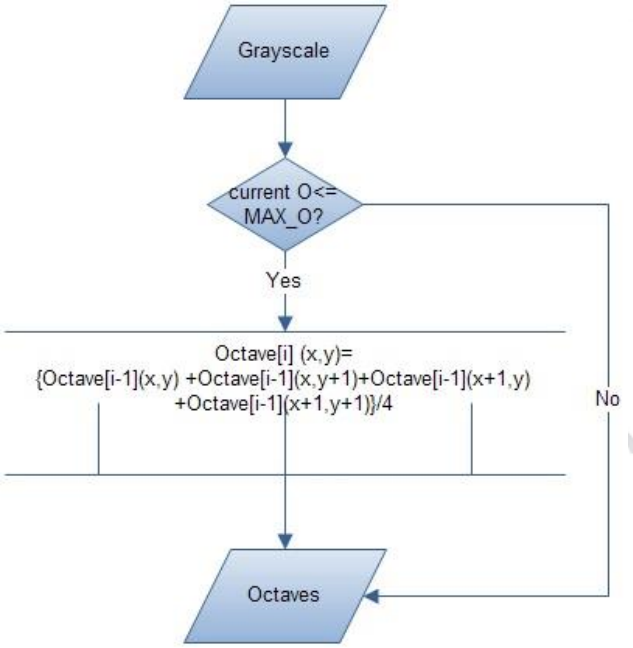
Ορισμένα χαρακτηριστικά των συναρτήσεων σε παράλληλη υλοποίηση έχουν αναφερθεί στην περιγραφή του αλγορίθμου που προηγήθηκε. Ακολούθως θα αναφερθούν περισσότερες και πιο σημαντικές τεχνικές λεπτομέρειες της κάθε υλοποιημένης συνάρτησης.

Στην συνάρτηση Grayscale μετατροπής, δεδομένα εισόδου αποτελούν τα τρία χρωματικά κανάλια της έγχρωμης εικόνας όπως έχουν εξαχθεί από το format που διαχειρίζεται το OpenCV. Τα τρία κανάλια εισάγονται ως ξεχωριστοί πίνακες και συνυπολογίζονται στον πίνακα της εικόνας εξόδου, με τους συντελεστές βαρύτητάς τους, κατά το πρότυπο φωταύγειας που ακολουθήθηκε.



3-xi Διάγραμμα ροής: Μονοχρωματικό φίλτρου με CUDA (CUDA grayscale flowchart)

Στην συνάρτηση της διγραμμικής παρεμβολής (bilinear interpolation-downsize) δεδομένο εισόδου αποτελεί η μονοχρωματική εικόνα και δεδομένα εξόδου οι εικόνες μικρότερων μεγεθών (οκτάβες) το πλήθος των οποίων καθορίζεται από τους υπολογισμούς που εκτελούνται στην αρχή του αλγορίθμου και εξαρτάται από τις διαστάσεις της εικόνας εισόδου. Η ταυτόχρονη παραγωγή όλων των επιπέδων, αν και εφικτή, καθίσταται μη βέλτιστη από το γεγονός ότι μία συνάρτηση bilinear interpolation με μεταβλητό μέγεθος εισόδου και αναδρομική δομή (recursive), που θα ήταν απαραίτητη, για να υπολογίζεται η μέση τιμή έντασης μεγαλύτερου πλήθους από pixel σε κάθε παραλληλισμένη επανάληψη, δεν ήταν βέλτιστη λόγω του εκθετικά υψηλότερου υπολογιστικού κόστους ανά pixel. Έτσι κρίθηκε ως απλούστερη και αποδοτικότερη λύση αυτή της επαναληπτικής εκτέλεσης της παράλληλης συνάρτησης με δεδομένο εισόδου σε κάθε επανάληψη την εικόνα που παράχθηκε από την προηγούμενη.



3-xii Διάγραμμα ροής: Σμίκρυνση με διγραμμική παρεμβολή με CUDA (CUDA bilinear interpolation downsampling flowchart)

Η συνάρτηση της δημιουργίας της Gaussian πυραμίδας δέχεται ως δεδομένα εισόδου τις εικόνες που έχει παράγει η συνάρτηση του bilinear interpolation, δηλαδή την πυραμίδα οκτάβων. Ως δεδομένα εξόδου έχει τις πυραμίδες Gaussian Blur, δηλαδή τον χώρο-κλίμακας (Scale-Space).

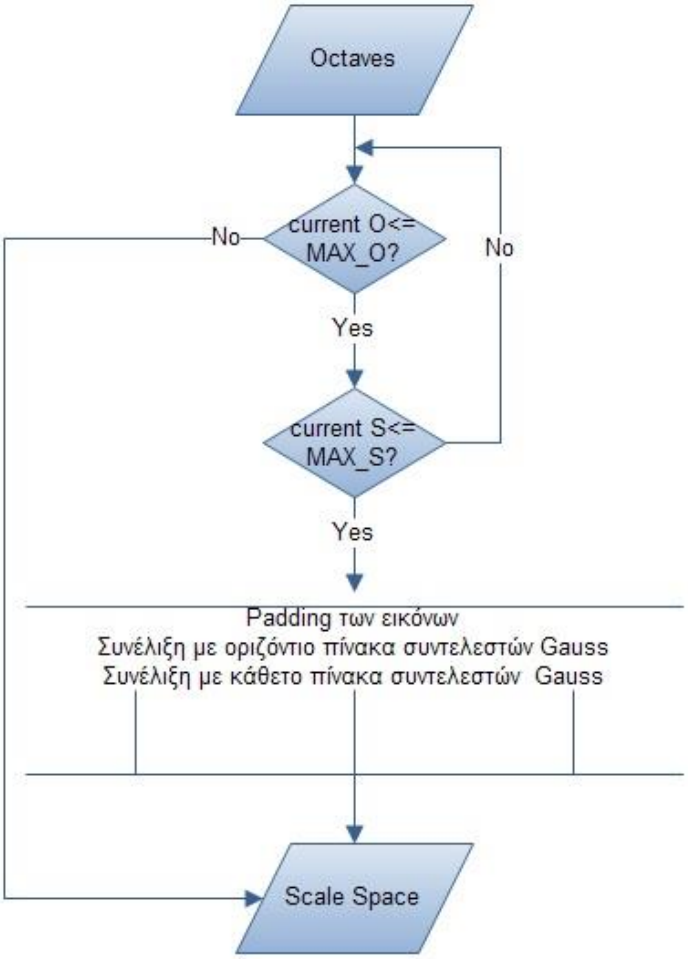
Στην πράξη γίνονται τρείς επαναληπτικές διαδικασίες. Αρχικά δημιουργούνται εικόνες με προσθήκη «κορνίζας» μεγαλώνοντας τις εικόνες περιμετρικά με επιπρόσθετη λωρίδα από pixel φάρδους gR προκειμένου να έχουν επαρκές μέγεθος για την ολίσθιση του συνελικτικού παραθύρου σε ολόκληρη την περιοχή ενδιαφέροντος της εικόνας. Τα pixel αυτά αυθαίρετα τίθενται σε τιμή 0 (μαύρο) και συνεπώς οι τιμές των συνελιγμένων εικόνων κοντά στα όρια τους, σε απόσταση gR, δεν είναι αντιπροσωπευτικές του μαθηματικού ιδανικού φίλτρου, αλλά αυτό είναι ένα πολύ γνωστό πρόβλημα της θόλωσης Gauss και για χάρη εξοικονόμησης υπολογιστικής ισχύος δεν κρίνεται απαραίτητο να δοθεί κάποια πιο περίπλοκη μαθηματική επίλυση. Έτσι και αλλιώς, σε μετέπειτα στάδιο του αλγορίθμου συνυπολογίζεται το σφάλμα αυτό και δεν παράγονται εσφαλμένα σημεία ενδιαφέροντος εξ' αιτίας του. Αυτό εξασφαλίζεται απορρίπτοντας τα σημεία ενδιαφέροντος που βρίσκονται κοντά στις άκρες της εικόνας.

Ενδεχομένως η χρήση κοινής μνήμης στα blocks να παρήγαγε ακόμα μεγαλύτερους δείκτες επιτάχυνσης, αλλά λόγω του χρόνου αντιγραφής των δεδομένων στην μνήμη για κάθε εικόνα και επίπεδο θόλωσης θα αποσβαινόταν σε βαθμό αρκετά πιο μεγάλο από τον επιθυμητό, για να θεωρηθεί βελτίωση, δεδομένου του ότι και η χρήση της μνήμης της κάρτας γραφικών θα ήταν πολλαπλάσια λόγω της απαραίτητης υπερκάλυψης των γειτονικών block που απαιτείται για την ομαλή επεξεργασία ολόκληρης της εικόνας.

Οι υπολογισμοί του παραθύρου συνέλιξης Gauss και των ορίων padding για τις ενδιάμεσες εικόνες βάσει του παραθύρου αυτού πραγματοποιούνται στο Host επειδή το πλήθος και ο βαθμός περιπλοκότητας των υπολογισμών δεν συνιστούν πρόβλημα που χρήζει παραλληλίας. Στην συνέχεια, και εφόσον έχουν μεταφερθεί οι εικόνες των πυραμίδων στην μνήμη της κάρτας γραφικών, όπως επίσης και οι μονοδιάστατοι πίνακες με τους μονοδιάστατους συντελεστές του δισδιάστατου παραθύρου συνέλιξης Gauss, εκτελείται η διαδικασία της συνέλιξης για κάθε εικόνα, κάθε οκτάβας, με το αντίστοιχο παράθυρο Gauss του εκάστοτε επιπέδου. Η διαδικασία αυτή αποτελείται από τρία στάδια, όπως προαναφέρθηκε, τα οποία εκτελούνται στην σειρά με την κλήση τριών διαδοχικών kernel grids. Ο πρώτος αναλαμβάνει να δημιουργήσει και να γομώσει την ενδιάμεση εικόνα μεταξύ των δύο περασμάτων του μονοδιάστατου Gauss πίνακα συντελεστών. Επίσης σχηματίζει την ίδια γόμωση στην αρχική εικόνα αντιγράφοντάς την φυσικά σε νέα, έχοντας προσθέσει έτσι την κορνίζα που αναφέρθηκε σε προηγούμενο κεφάλαιο.

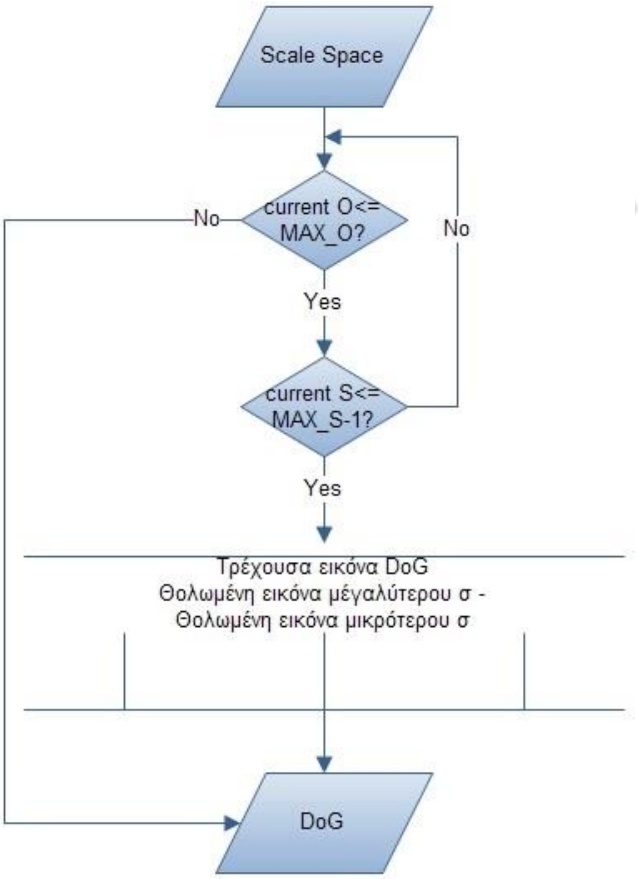
Στην συνέχεια καλούνται οι δύο kernels που πραγματοποιούν την οριζόντια και κάθετη σάρωση του συνελικτικού παραθύρου [107] έχοντας παράγει ως αποτέλεσμα ο πρώτος την ενδιάμεση εικόνα και ο δεύτερος την τελική θολωμένη συνελιγμένη κατά Gauss εικόνα.

Ο βαθύς πυρήνας της παραλληλίας όπως προαναφέρθηκε, βρίσκεται στην επεξεργασία των pixel ένα προς ένα και έτσι η διαδικασία των τριών αυτών συναρτήσεων επαναλαμβάνεται για όλες τις εικόνες της κάθε οκτάβας και κάθε επιπέδου θόλωσης.



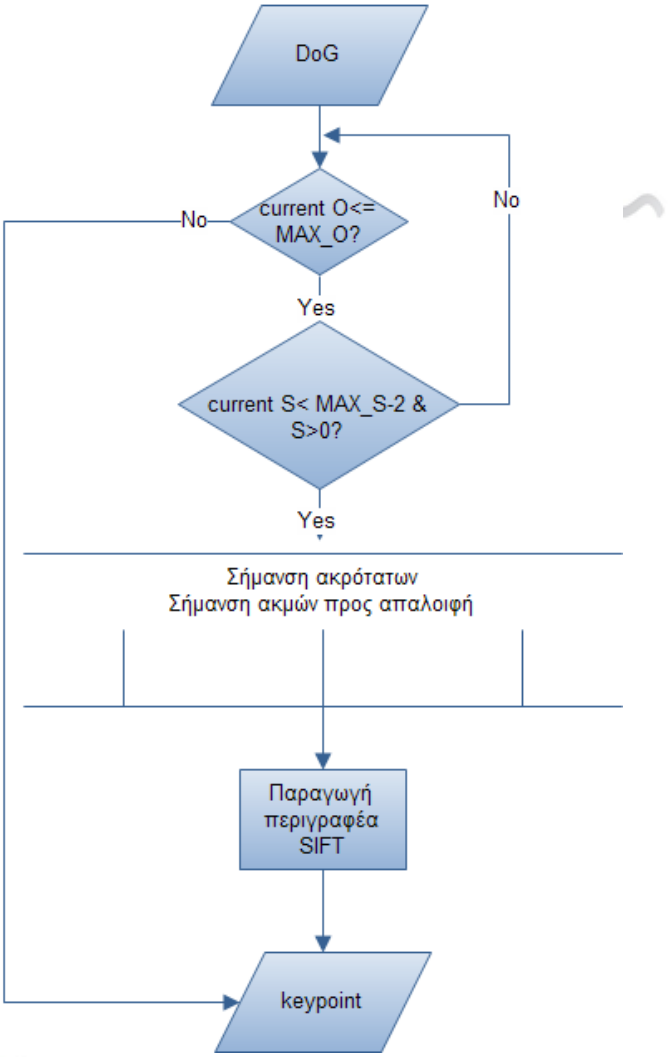
3-xiii Διάγραμμα ροής: Πυραμίδες Χώρου Κλίμακας με CUDA (CUDA Gaussian Scale Space Pyramid Flowchart)

Το στάδιο της παραγωγής των εικόνων Difference of Gaussians είναι μικρότερης υπολογιστικής πολυπλοκότητας από την θόλωση Gauss. Με τους ίδιους όρους παραλληλίας, δηλαδή για κάθε ζεύγος εικόνων των επιπέδων θόλωσης και σε κάθε οκτάβα, πραγματοποιούνται αφαιρέσεις των αντιστοιχούντων pixel και παράγονται οι διαφορικές εικόνες των οποίων το πλήθος είναι μία λιγότερη ανά οκτάβα από το Scale Space (S-1).



3-xiv Διάγραμμα ροής: Πυραμίδες Διαφορών Gauss με CUDA (CUDA Difference of Gaussians Pyramid flowchart)

Το στάδιο της παραγωγής ακρότατων (extrema localization) όπως και αυτό της απαλοιφής αυτών που ανήκουν σε ακμές παρουσιάζουν υψηλό δείκτη παραλληλίας καθώς οι 26 συγκρίσεις που απαιτούνται για κάθε pixel της εκάστοτε εικόνας DoG είναι χρονοβόρα και επαναληπτική διαδικασία. Στα πλαίσια αυτής της εργασίας υλοποιήθηκε η παραλληλοποίηση του σταδίου των ακρότατων και στην συνέχεια χρησιμοποιήθηκε το ακολουθιακό τμήμα του αλγορίθμου για τον προσδιορισμό των ακμών.



3-xv Διάγραμμα ροής εξαγωγής ακρότατων και απαλοιφής ακμών (Extrema localization and Edge point elimination)

Το τελικό στάδιο του αλγορίθμου αν και επαναληπτικό, δεν παρουσιάζει μεγάλη παραλληλησιμότητα. Αυτό οφείλεται στο γεγονός ότι τα keypoints έχουν πλήθος απρόβλεπτο και εξαρτώμενο από την φιλικότητα της εκάστοτε εικόνας προς τον αλγόριθμο SIFT. Επίσης, ο χρόνος εκτέλεσης του τμήματος αυτού σε ακολουθιακή μορφή είναι σχεδόν αμελητέος σε σύγκριση με άλλα τμήματα του αλγορίθμου.

3.5 Πειραματικές μετρήσεις

Για την διεξαγωγή μετρήσεων χρονισμού, χρησιμοποιήθηκαν συναρτήσεις του OpenCV αναδιπλωμένες σε ιδιότυπες συναρτήσεις διαχείρισής τους [108]. Έγινε χρήση μίας καθολικά δηλωμένης δομής η οποία καταχωρεί σε δυναμικό πίνακα την χρονική στιγμή της εκάστοτε μέτρησης μαζί με ένα σύντομο κείμενο που περιέγραφε την στιγμή αυτή. Ο χρόνος μετράται σε παλμούς ρολογιού του συστήματος από την στιγμή της εκκίνησής του. Η συνάρτηση καλείται να αφαιρέσει την τρέχουσα τιμή από την αρχική τιμή που λαμβάνεται κατά την εκκίνηση του προγράμματος και να καταχωρήσει την διαφορά ως παρελθόντα χρόνο εκτέλεσης. Δεν κρίθηκε σωστό οι χρονικές μετρήσεις να καταγράφονται σε αρχείο κατά την στιγμή της καταχώρησής τους επειδή αυτό θα δημιουργούσε καθυστέρηση λόγω της αντίστοιχης λειτουργίας προσπέλασης αρχείου. Η αποθήκευση του συνόλου των καταγραφών γίνεται σε δύο αρχεία μετά το πέρας της εκτέλεσης. Το ένα αρχείο έχει αναλυτικές τιμές των παλμών ρολογιού, του περιγραφικού κειμένου και της μέτρησης, μετατρεμμένη σε μονάδες millisecond που προκύπτουν γνωρίζοντας και την συχνότητα λειτουργίας της CPU η οποία επίσης διαπιστώνεται από το api του OpenCV. Το δεύτερο αρχείο έχει λιγότερες πληροφορίες και είναι σε μορφή λίστας, με τιμές χωρισμένες από διαχωριστικά λίστας (tab), προκειμένου να είναι αξιοποιήσιμα τα δεδομένα του με τρόπο εργονομικό από λογιστικό φύλλο.

Στο λογιστικό φύλλο, και με την βοήθεια των κειμένων σήμανσης που καταχωρήθηκαν, είναι πλέον εύκολη η εξαγωγή χρόνων εκτέλεσης για οποιοδήποτε τμήμα του κώδικα κρίνεται επιθυμητό.

Έτσι εξάγονται διαφορετικά στατιστικά στοιχεία που οδηγούν σε χρήσιμα συμπεράσματα για την βελτιστοποίηση του αλγορίθμου και της παράλληλης υλοποίησής του. Διαπιστώθηκε με αυτόν τον τρόπο πως μεγάλο χρονικό διάστημα δαπανάται για την μεταφορά δεδομένων από και προς την μνήμη της κάρτας γραφικών, κάτι που πρέπει να συνυπολογίζεται στην σχεδίαση και την αξιολόγηση της εκάστοτε παράλληλης υλοποίησης.

Στην προκειμένη εργασία, δοκιμάστηκε ένα σύνολο εικόνων οι οποίες είχαν μετατραπεί σε διάφορα προκαθορισμένα μεγέθη κατά τα εμπορικώς αποδεκτά πρότυπα ανάλυσης εικόνας. Οι μετατροπές αυτές έγιναν με το πρόγραμμα FastStone Photo Resizer και κάνοντας χρήση του αλγορίθμου Lanczos3. Οι αρχικές εικόνες είχαν μέγεθος 1080p, δηλαδή 1920*1080, διαστάσεις που τυπικά αναφέρονται ως Full HD.

Στο τέλος κάθε συγκριτικής εκτέλεσης, υπολογίζεται μία ακόμα εικόνα η οποία αποτελεί τη σημείο προς σημείο διαφορά μεταξύ των αποτελεσμάτων της εκτέλεσης CPU και GPU, κατά τρόπο παρόμοιο με τα disparity maps των εφαρμογών στερεοσκοπικής όρασης. Ο λόγος που γίνεται αυτό είναι να επιβεβαιωθεί η ομοιότητα των αποτελεσμάτων μεταξύ των δύο εκτελέσεων. Όταν τα δύο αυτά αποτελέσματα είναι πανομοιότυπα, η εικόνα αυτή είναι εντελώς μαύρη, αφού οι διαφορές των pixel δίνουν τιμές μηδενικές. Έτσι οποιαδήποτε απόκλιση θα φανεί ως artifact μέσα στην μαύρη εικόνα. Σε ορισμένες περιπτώσεις όντως παρουσιάζονται μερικές διαφορές, οι οποίες αφορούν έναν μικρό αριθμό από SIFT keypoints τα οποία αναγνωρίστηκαν μόνο από μία από τις δύο εκτελέσεις ή έχουν υποστεί διαφορετική στρογγυλοποίηση στις παραμέτρους τους. Αυτό εκτιμάται ότι οφείλεται σε μικρές διαφορές στα αποτελέσματα αριθμητικών πράξεων κινητής υποδιαστολής, δεδομένου του ότι ενδέχεται να υπάρχουν διαφορές στην διασπορά των αριθμών κινητής υποδιαστολής στα πρότυπα αρχιτεκτονικής των επεξεργαστών της NVIDIA και της INTEL. Σε κάθε περίπτωση όμως, οι διαφορές αυτές είναι ελάχιστες και ασήμαντες ως προς την ομαλή και αξιόπιστη λειτουργία του αλγορίθμου και οποιασδήποτε εφαρμογής μηχανικής όρασης πρόκειται να τον χρησιμοποιήσει. Δεδομένου του ότι η ταύτιση ενός αντικειμένου εμπειρικά συνήθως απαιτεί μικρό αριθμό keypoints, σε ορισμένες περιπτώσεις είναι αρκετή η ύπαρξη τριών σημείων, για να αναγνωριστεί αξιόπιστα ένα αντικείμενο. Έτσι λοιπόν δεν κρίνεται σκόπιμο να υλοποιηθεί επιπρόσθετη λειτουργικότητα, η οποία θα αύξανε τον υπολογιστικό φόρτο και θα αναιρούσε μέρος της επιτάχυνσης, προκειμένου να αντιμετωπιστεί αυτός ο παράγοντας.

Πραγματοποιήθηκαν δοκιμές του SIFT σε εκτέλεση GPU και CPU για τα ακόλουθα μεγέθη εικόνας:

Πρότυπο	Διαστάσεις	Πλήθος pixels
240p	427*240	102480
320p	569*320	182080
480p	853*480	409440
576p	1024*576	589824
(Half HD) 720p	1280*720	921600 (1MP)
810p	1440*810	1166400
(Full HD) 1080p	1920*1080	2073600 (2MP)

3-xvi Μεγέθη δοκιμίων

Το βασικό μέγεθος που υπολογίστηκε από τις μετρήσεις των επιμέρους παραλληλισμένων τμημάτων του αλγορίθμου είναι ο παράγοντας επιτάχυνσης:

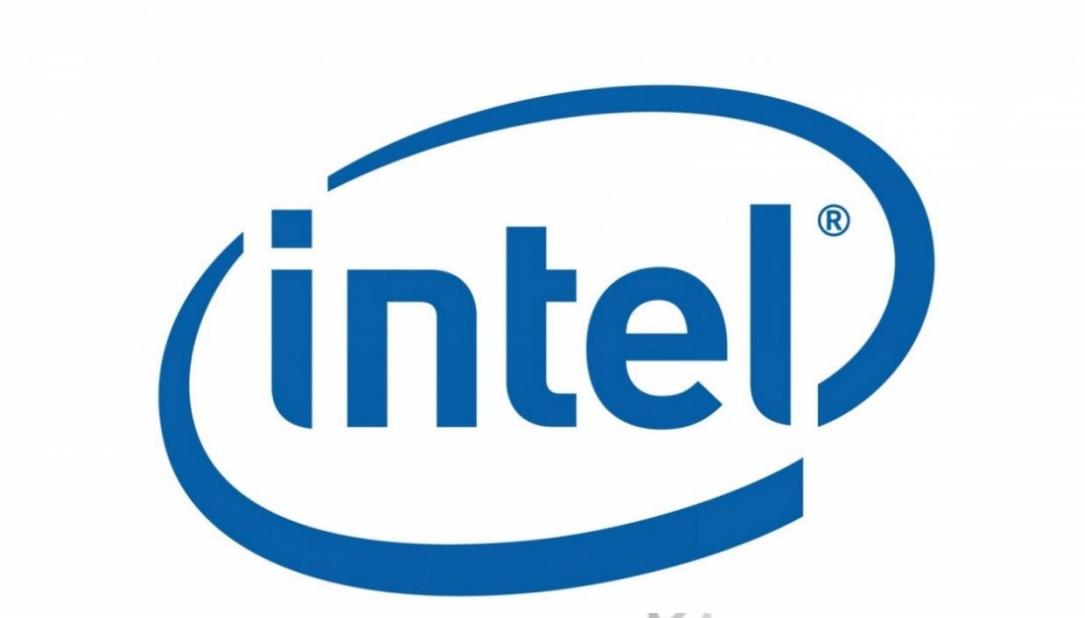
$$S(p) = \frac{t_s}{t_p}$$

Ο υπολογισμός του για κάθε τμήμα συμπεριλαμβάνει την αντιγραφή δεδομένων στην μνήμη της κάρτας γραφικών, εκτός από τις συναρτήσεις που η παράλληλη υλοποίησή τους, σκόπιμα πραγματοποιεί την δέσμευση της μνήμης και την αντιγραφή από και σε αυτήν εκτός του βασικού επαναληπτικού βρόγχου, σε αντίθεση με την σειριακή εκδοχή του αλγορίθμου που πραγματοποιεί το αντίστοιχο στην κύρια μνήμη εντός του επαναληπτικού βρόγχου. Ο παράγοντας αυτός ενδέχεται να μεγάλωνε εκθετικά σε δοκιμές εικόνων μεγαλύτερες από Full HD (1920*1080) αλλά με την παρούσα υλοποίηση αυτό προκαλεί σφάλματα out of memory στην GPU δεδομένου του ότι η κάρτα γραφικών που χρησιμοποιήθηκε έχει 1GB gRAM. Η τμηματική επεξεργασία είναι πιθανώς εφικτή αλλά με κόστος χρόνου λόγω υπολογισμών τεμαχισμού της εικόνας, cudaMalloc και cudaMemcpy.

Ακολούθως θα παρατεθούν ενδεικτικά οι μετρήσεις και οι υπολογισμοί για πέντε εικόνες που εξετάστηκαν στην διενέργεια του πειράματος.

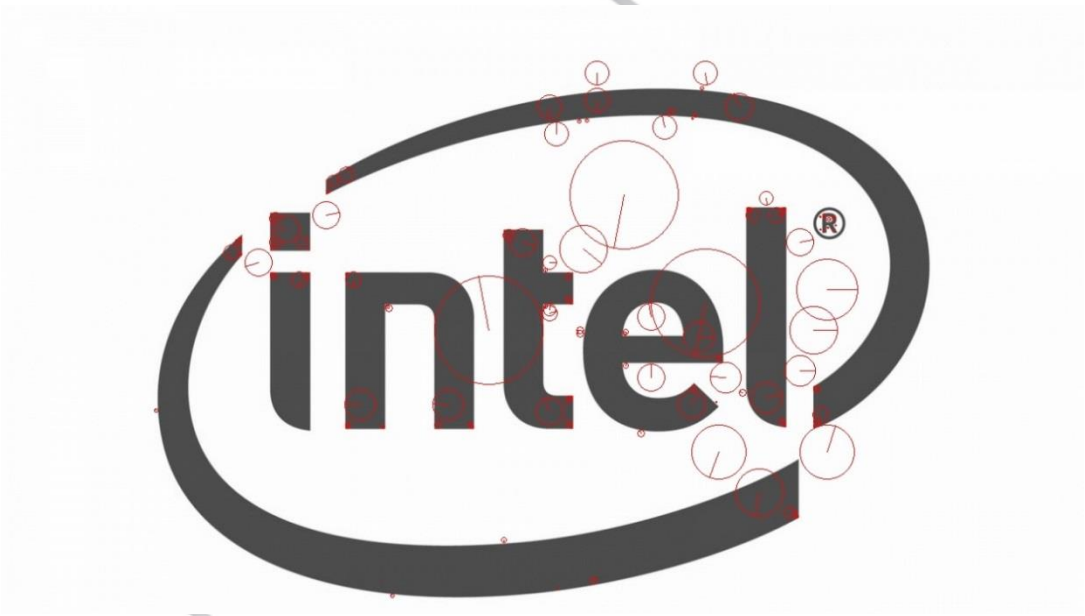
Intel logo

Η αρχική εικόνα είναι:



3-xvii Intel logo (original)

Η εικόνα μετά την εκτέλεση του αλγορίθμου και την αποτύπωση των SIFT keypoints:

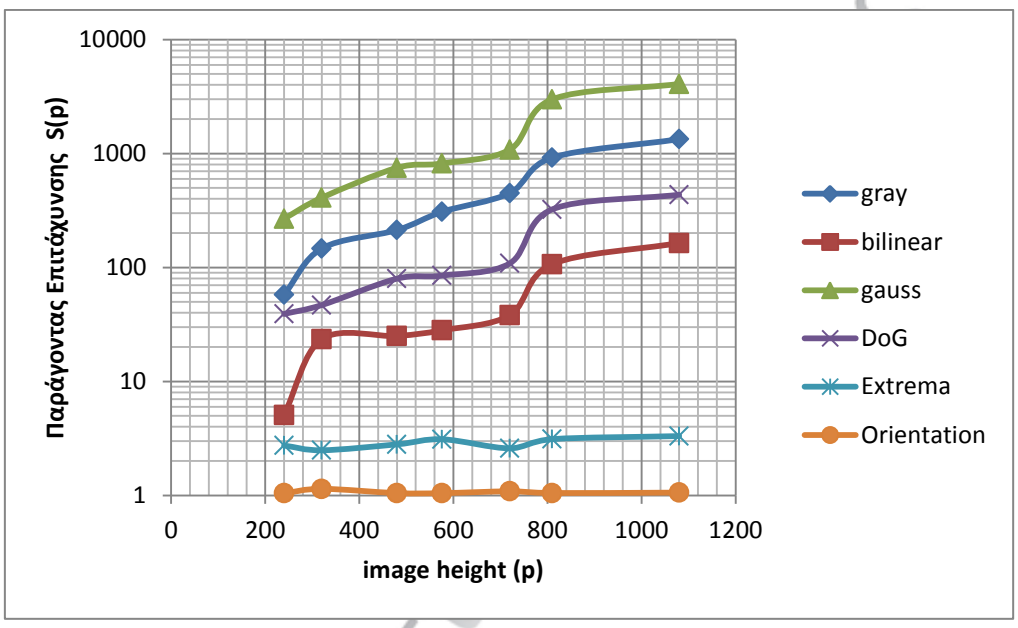


3-xviii Intel logo (SIFT)

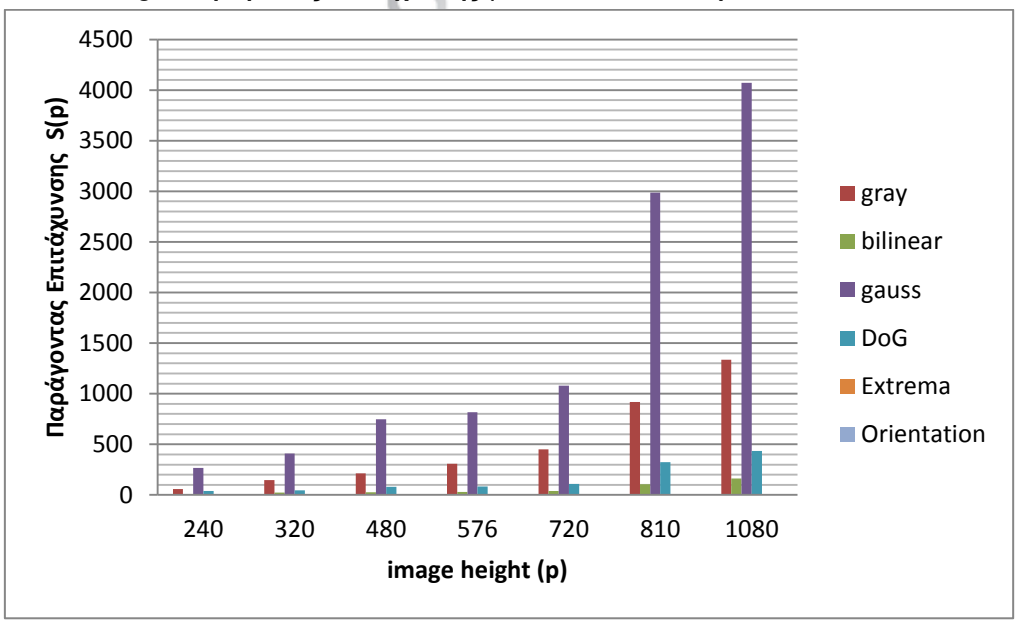
Μετά την εκτέλεση του αλγορίθμου, έχουν εξαχθεί μετρήσεις χρονισμών για τις δύο εκδοχές του αλγορίθμου CPU και GPU. Με τις μετρήσεις αυτές υπολογίζεται ο παράγοντας επιτάχυνσης $S(p)$ για κάθε διάσταση της εικόνας που δοκιμάστηκε στην εκτέλεση αυτή. Ακολουθεί ο πίνακας των επιταχύνσεων ανά τμήμα του αλγόριθμου και οι γραφικές παραστάσεις των δεδομένων. Οι γωνίες των γραμμάτων προσφέρονται ως αξιόπιστα σημεία ενδιαφέροντος

Intel logo	*gray*	*bilinear*	*gauss*	*DoG*	Extrema	Orientation
240p	57.72	5.088235	267.8398	39.09251	2.738544	1.044554
320p	146.6875	23.44681	408.7709	46.75862	2.489364	1.14188
480p	212.8889	25.05556	748.5529	79.71496	2.804043	1.047673
576p	307.8519	28.13235	818.1067	84.87434	3.108363	1.046218
720p	449.8621	38.16667	1080.921	108.6317	2.586526	1.085714
810p	919.6111	106.7222	2985.479	323.2192	3.115661	1.049383
1080p	1335.273	163.8293	4072.002	434.8212	3.314882	1.058657

3-xix Intel logo: Μετρήσεις Παράγοντα Επιτάχυνσης (Acceleration Factor measurements)



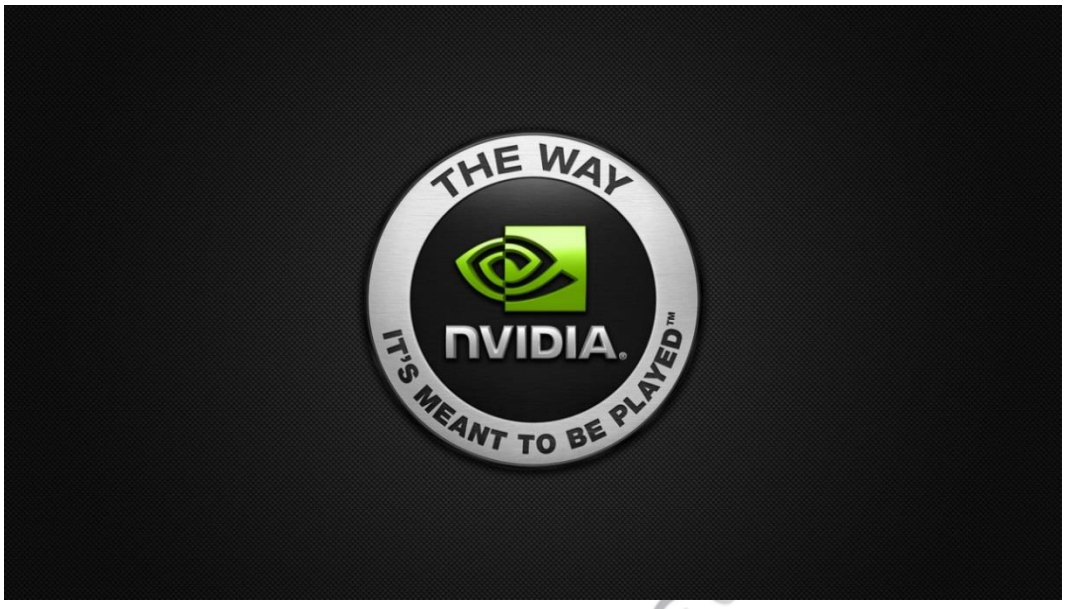
3-xx Intel logo: Παράγοντας Επιτάχυνσης (Acceleration Factor)



3-xxi Intel logo: Παράγοντας Επιτάχυνσης ανά ύψος εικόνας (Acceleration Factor per image size)

NVIDIA logo

Η αρχική εικόνα είναι:



3-xxii NVIDIA logo (original)

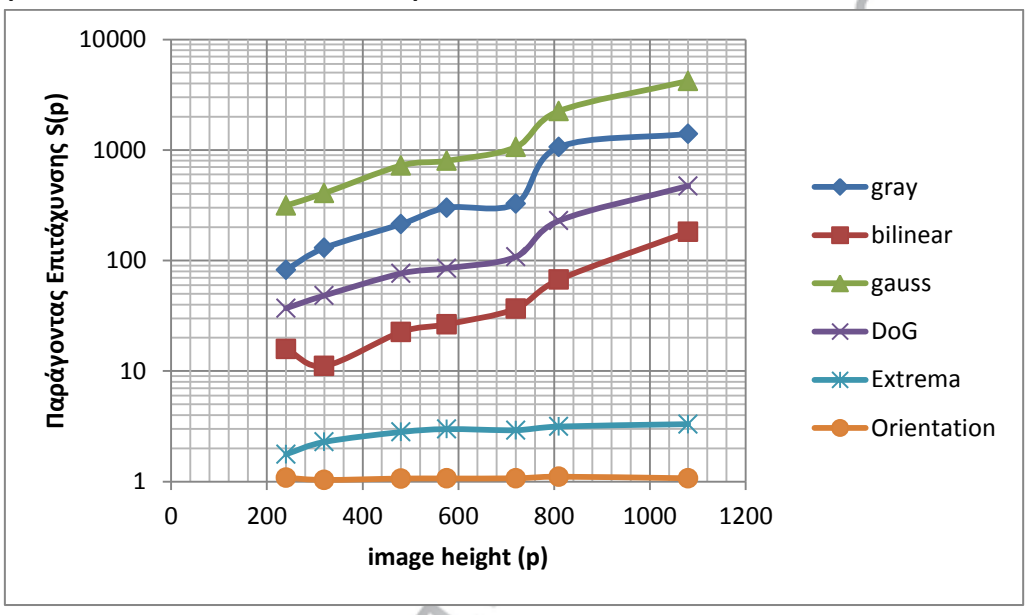
Η εικόνα μετά την εκτέλεση του αλγορίθμου και την αποτύπωση των SIFT keypoints:

3-xxiii NVIDIA logo (SIFT)

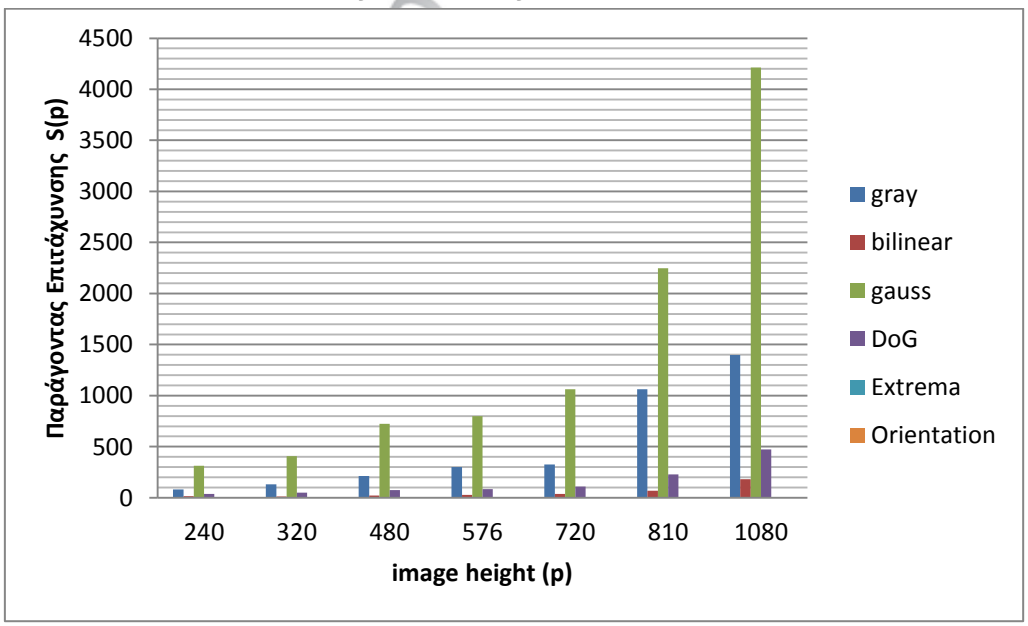
Μετά την εκτέλεση του αλγορίθμου, έχουν εξαχθεί μετρήσεις. Με τις μετρήσεις χρονισμών για τις δύο εκδοχές του αλγορίθμου CPU και GPU, υπολογίζεται ο παράγοντας επιτάχυνσης $S(p)$ για κάθε διάσταση της εικόνας που δοκιμάστηκε στην εκτέλεση αυτή. Ακολουθεί ο πίνακας των επιταχύνσεων ανά τμήμα του αλγόριθμου και οι γραφικές παραστάσεις των δεδομένων. Η μεγάλη μαύρη επιφάνεια προσφέρει ως σημεία ενδιαφέροντος μόνο τις εναλλαγές υψηλής αντίθεσης των γραμμάτων και του εσωτερικού λογότυπου.

NVIDIA logo	*gray*	*bilinear*	*gauss*	*DoG*	Extrema	Orientation
240p	82.4375	15.84615	314.6257	37.04803	1.769338	1.083481
320p	129.8846	11.12963	406.9135	48.24919	2.284259	1.035599
480p	213.3704	22.53333	722.6655	76.29128	2.807928	1.066393
576p	300.4643	26.5	799.6141	85.26203	2.984301	1.067047
720p	326.475	36.65476	1062.395	108.0203	2.91818	1.07183
810p	1061.792	67.12281	2248.769	228.9005	3.150356	1.108108
1080p	1398.857	181.5385	4212.57	473.1497	3.310656	1.072155

3-xxiv NVIDIA logo: Μετρήσεις Παράγοντα Επιτάχυνσης (Acceleration Factor measurements)



3-xxv NVIDIA logo: Παράγοντας Επιτάχυνσης (Acceleration Factor)



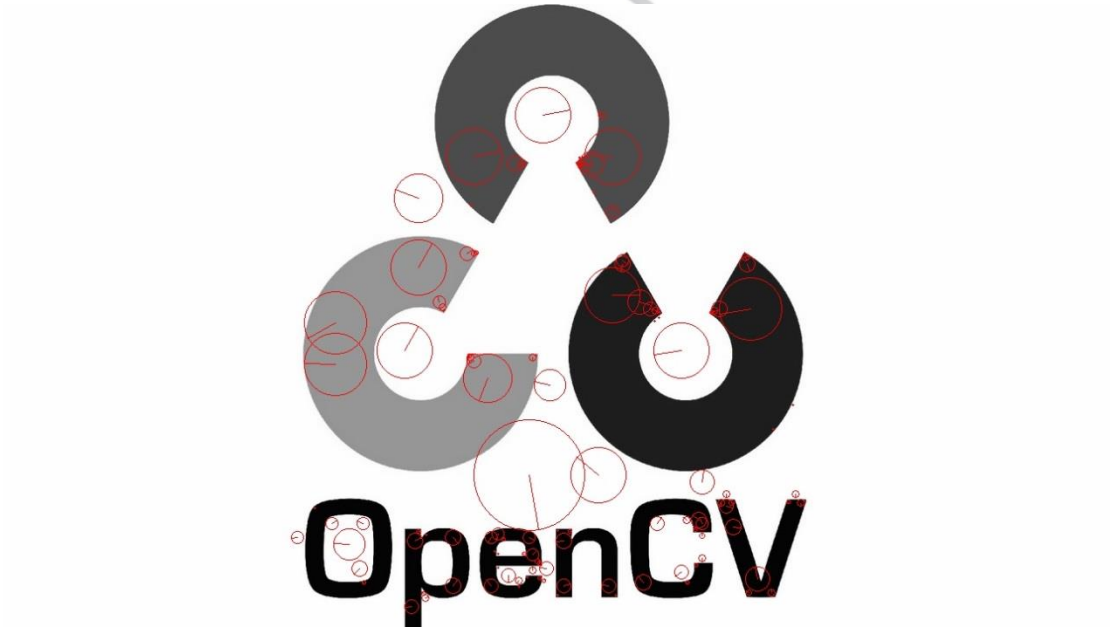
3-xxvi NVIDIA logo: Παράγοντας Επιτάχυνσης ανά ύψος εικόνας (Acceleration Factor per image size)

OpenCV logo

Η αρχική εικόνα είναι:

3-xxvii OpenCV logo (original)

Η εικόνα μετά την εκτέλεση του αλγορίθμου και την αποτύπωση των SIFT keypoints:

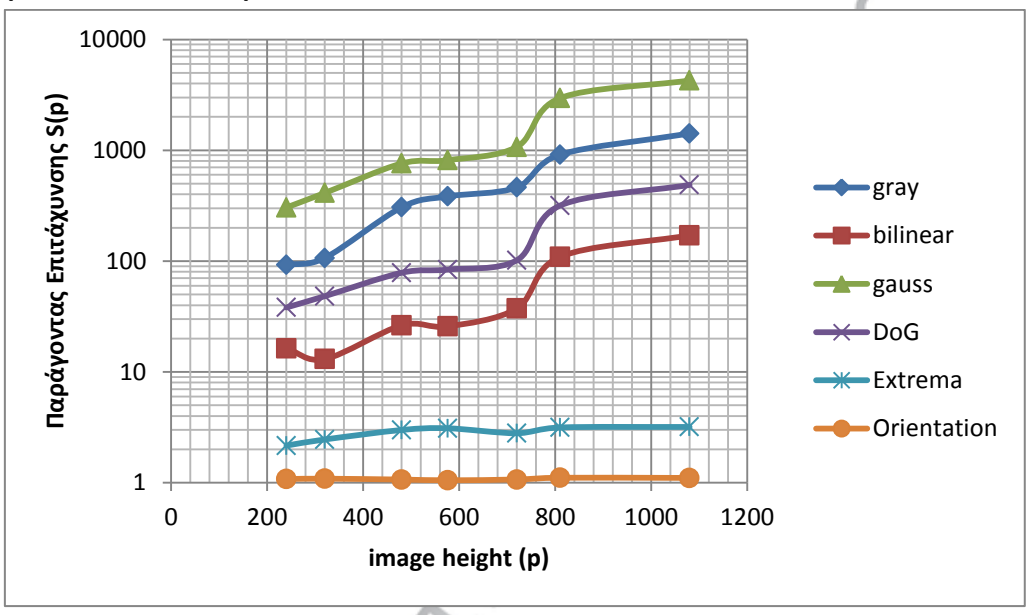


3-xxviii OpenCV logo (SIFT)

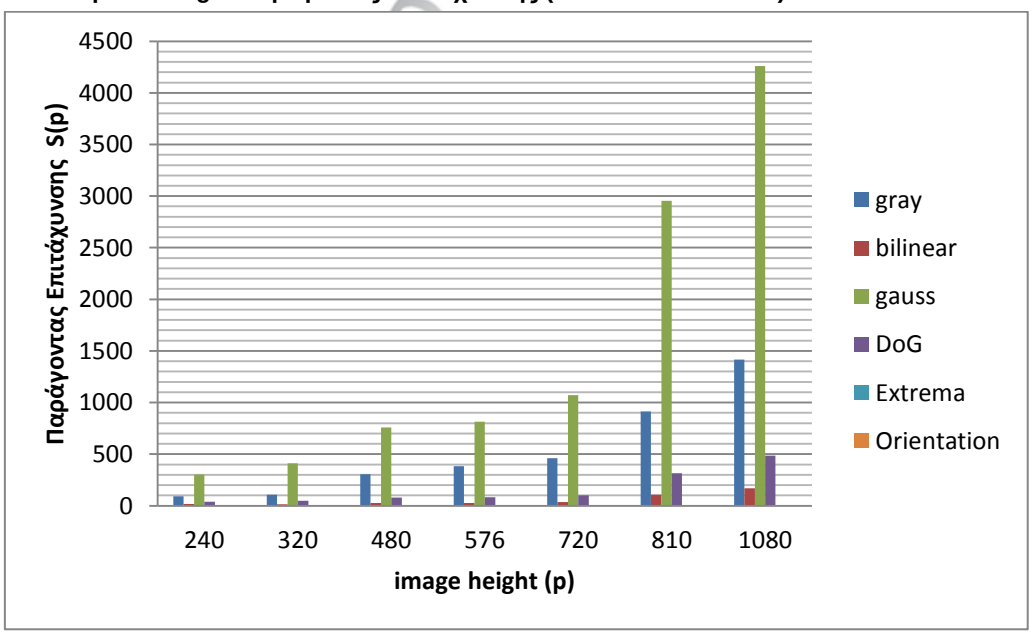
Μετά την εκτέλεση του αλγορίθμου, έχουν εξαχθεί μετρήσεις χρονισμών για τις δύο εκδοχές του αλγορίθμου CPU και GPU. Με τις μετρήσεις αυτές υπολογίζεται ο παράγοντας επιτάχυνσης $S(p)$ για κάθε διάσταση της εικόνας που δοκιμάστηκε στην εκτέλεση αυτή. Βλέπουμε πως ένα απλούστερο λογότυπο αναλύεται σε λιγότερα αλλά σταθερότερα σημεία ενδιαφέροντος. Ακολουθεί ο πίνακας των επιταχύνσεων ανά τμήμα του αλγόριθμου και οι γραφικές παραστάσεις των δεδομένων.

OpenCV logo	*gray*	*bilinear*	*gauss*	*DoG*	Extrema	Orientation
240p	92.3871	16.31579	304.3938	38.12556	2.161504	1.078431
320p	106.7083	13.04348	411.9283	48.21725	2.455026	1.086879
480p	305.5714	26.28302	759.8716	78.43499	2.988628	1.067293
576p	383.55	25.72973	814.1716	83.84331	3.098044	1.051044
720p	461.375	37.3875	1071.773	101.6182	2.791623	1.068859
810p	913.2778	109.0571	2953.371	316.8013	3.150356	1.108108
1080p	1416.19	170.6	4261.048	486.6802	3.184533	1.104013

3-xxix OpenCV logo: Μετρήσεις Παράγοντα Επιτάχυνσης (Acceleration Factor)measurements



3-xxx OpenCV logo: Παράγοντας Επιτάχυνσης (Acceleration Factor)



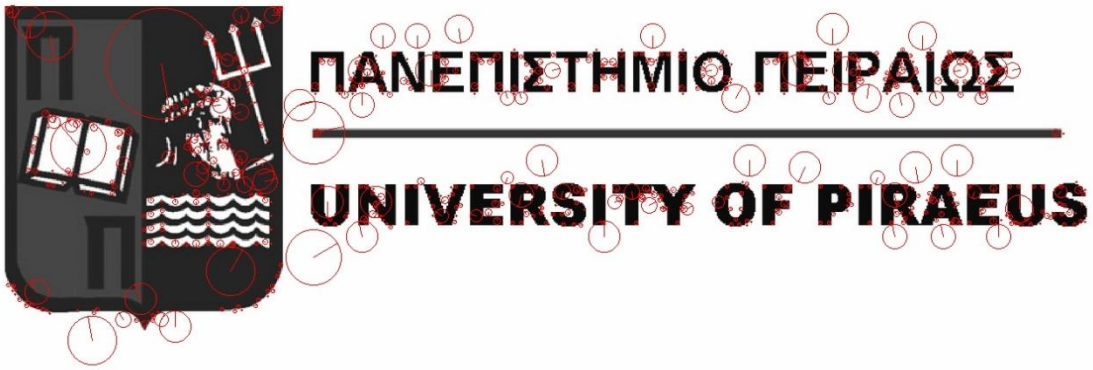
3-xxxi OpenCV logo: Παράγοντας Επιτάχυνσης ανά ύψος εικόνας (Acceleration Factor per image size)

(University of Piraeus) UniPi logo

Η αρχική εικόνα είναι:

3-xxxii UniPi logo (original)

Η εικόνα μετά την εκτέλεση του αλγορίθμου και την αποτύπωση των SIFT keypoints:

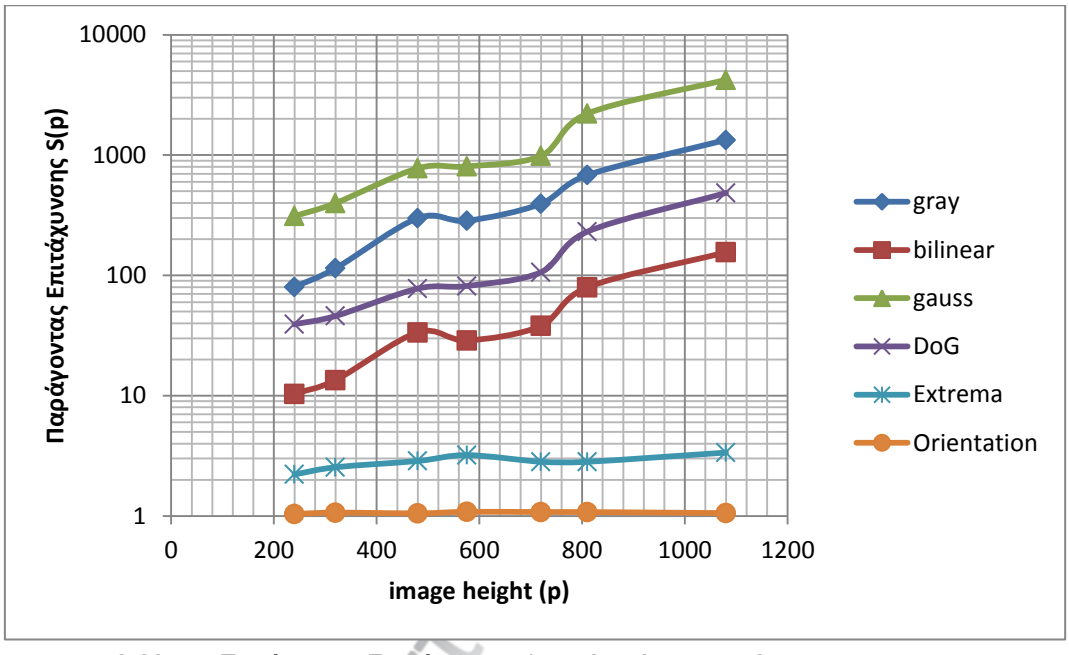


3-xxxiii UniPi logo (SIFT)

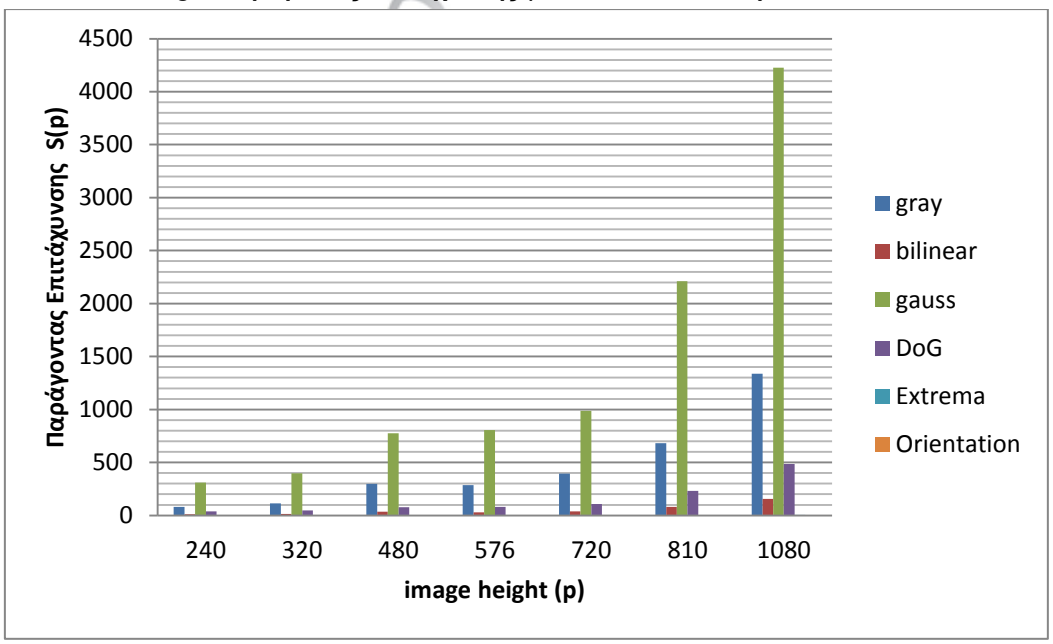
Μετά την εκτέλεση του αλγορίθμου, έχουν εξαχθεί μετρήσεις χρονισμών για τις δύο εκδοχές του αλγορίθμου CPU και GPU. Με τις μετρήσεις αυτές υπολογίζεται ο παράγοντας επιτάχυνσης $S(p)$ για κάθε διάσταση της εικόνας που δοκιμάστηκε στην εκτέλεση αυτή. Μία εικόνα με μικρότερη ευκρίνεια αλλά και πολλές λεπτομέρειες όπως τα γράμματα παράγει περισσότερα αλλά πιο ασταθή σημεία ενδιαφέροντος. Ακολουθεί ο πίνακας των επιταχύνσεων ανά τμήμα του αλγόριθμου και οι γραφικές παραστάσεις των δεδομένων.

UniPi logo	*gray*	*bilinear*	*gauss*	*DoG*	Extrema	Orientation
240p	80.06061	10.36667	310.9522	39.22581	2.216216	1.041131
320p	114.5652	13.47826	398.85	45.97538	2.541619	1.066038
480p	299.3667	33.61538	776.6535	77.67991	2.868145	1.053005
576p	284.7895	28.80282	804.3312	81.67747	3.207056	1.083473
720p	393.6364	38.06173	985.627	106.2736	2.824193	1.078378
810p	681.5862	79.58333	2211.176	230.6005	2.824193	1.078378
1080p	1336.273	155.7209	4225.17	484.5407	3.366415	1.057084

3-xxxiv UniPi logo: Μετρήσεις Παράγοντα Επιτάχυνσης (Acceleration Factor measurements)



3-xxxv UniPi logo: Παράγοντας Επιτάχυνσης (Acceleration Factor)



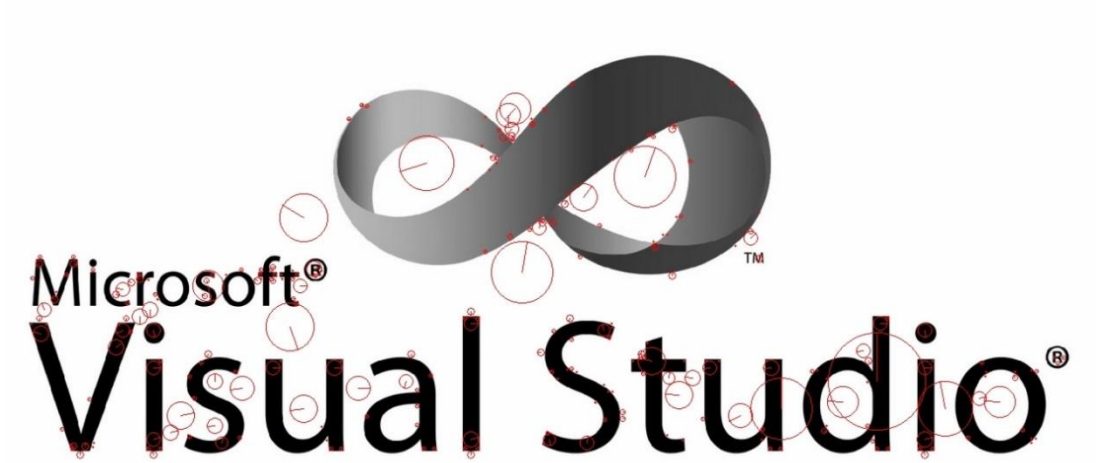
3-xxxvi UniPi logo: Παράγοντας Επιτάχυνσης ανά ύψος εικόνας (Acceleration Factor per image size)

(Microsoft Visual Studio) Ms VS logo

Η αρχική εικόνα είναι:

3-xxxvii Ms VS logo (original)

Η εικόνα μετά την εκτέλεση του αλγορίθμου και την αποτύπωση των SIFT keypoints:

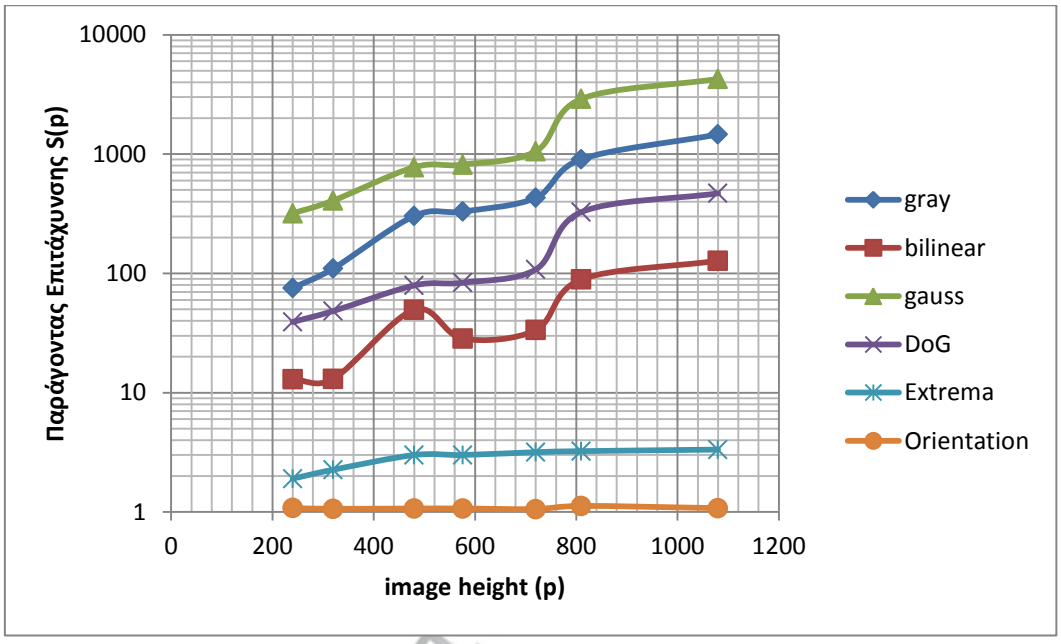


3-xxxviii Ms VS logo (SIFT)

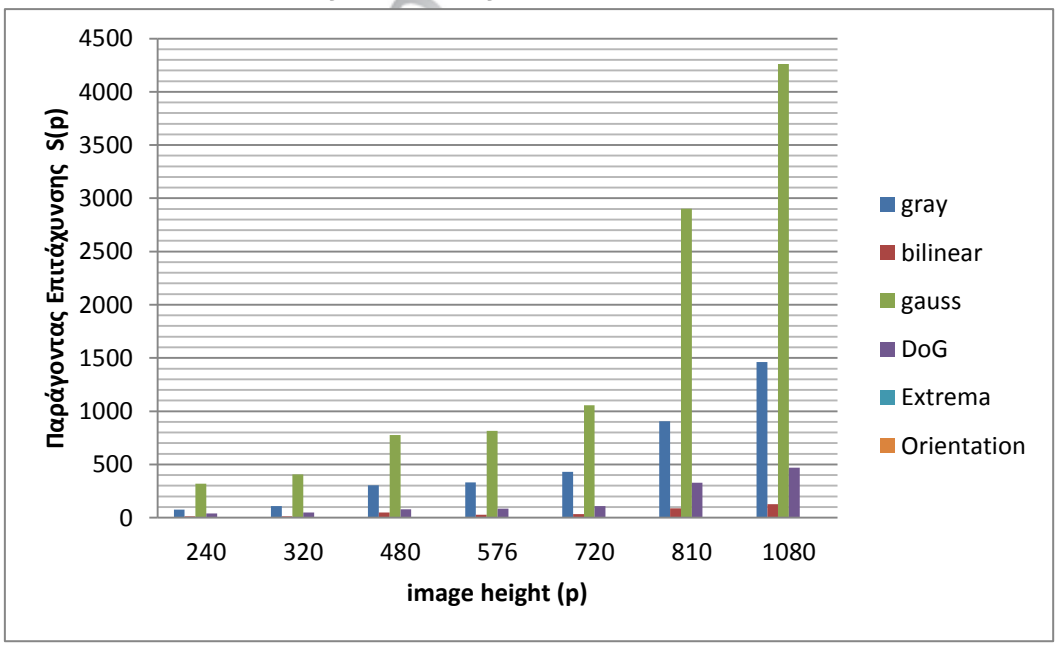
Μετά την εκτέλεση του αλγορίθμου, έχουν εξαχθεί μετρήσεις χρονισμών για τις δύο εκδοχές του αλγορίθμου CPU και GPU. Με τις μετρήσεις αυτές υπολογίζεται ο παράγοντας επιτάχυνσης $S(p)$ για κάθε διάσταση της εικόνας που δοκιμάστηκε στην εκτέλεση αυτή. Τα σημεία ενδιαφέροντος σε λογότυπο με γράμματα είναι περισσότερα και σταθερά όταν η εικόνα είναι ευκρινής. Τα γράμματα όμως εξακολουθούν να φέρουν τον κίνδυνο θορύβου. Ακολουθεί ο πίνακας των επιταχύνσεων ανά τμήμα του αλγόριθμου και οι γραφικές παραστάσεις των δεδομένων.

MsVS logo	*gray*	*bilinear*	*gauss*	*DoG*	Extrema	Orientation
240p	75.39286	13	318.743	39.16744	1.904229	1.079566
320p	109.8333	13.04348	407.6217	48.36482	2.261842	1.063616
480p	303.1429	49.62	776.4931	79.29904	3.004466	1.070046
576p	330.037	28.38571	815.6671	83.66143	3.004466	1.070046
720p	429.625	33.50562	1054.484	107.8442	3.174325	1.05863
810p	906.0952	88.97674	2900.995	326.6793	3.236862	1.124904
1080p	1460.3	127.566	4262.487	468.7087	3.334601	1.077013

3-xxxix Ms VS logo: Μετρήσεις Παράγοντα Επιτάχυνσης (Acceleration Factor measurements)



3-xl Ms VS logo: Παράγοντας Επιτάχυνσης (Acceleration Factor)



3-xli MsVS logo: Παράγοντας Επιτάχυνσης ανά ύψος εικόνας (Acceleration Factor per image size)

3.6 Συμπεράσματα

Οι πειραματικές μετρήσεις δείχνουν με σαφήνεια πως ο αλγόριθμος SIFT επωφελείται σημαντικά από την εκμετάλλευση της παραλληλίας του και την υλοποίηση της μαζικά παράλληλης εκδοχής του. Η ωφέλεια είναι διακριτή σε όλα τα μεγέθη εικόνων που δοκιμάστηκαν, αλλά πολύ σημαντικότερη στα μεγαλύτερα μεγέθη εικόνων. Οι λόγοι επιτάχυνσης που επετεύχθησαν κυμαίνονται από μερικές εκατοντάδες φορές στις συναρτήσεις μικρότερης υπολογιστικής περιπλοκότητας έως μερικές χιλιάδες φορές στις περισσότερο σύνθετες. Ειδικά για τις εικόνες Full HD σημειώθηκαν τιμές τμηματικής επιτάχυνσης πάνω από 4200 φορές εν σχέσει με την ταχύτητα της ακολουθιακής εκδοχής του αλγορίθμου. Όπως έχει προαναφερθεί, ο παράγοντας αυτός πιθανόν να αυξανόταν περισσότερο σε εικόνες μεγαλύτερής ανάλυσης, όμως περιορισμοί μνήμης της κάρτας γραφικών καθιστούν την υλοποίηση τέτοιων δοκιμών ανέφικτη με την παρούσα μορφή του αλγορίθμου. Για να επιτευχθεί αυτό, θα απαιτούσε τον κατακερματισμό των μεγάλων εικόνων σε μικρότερες και την επαναληπτική μερική επεξεργασία τους. Για να υλοποιηθεί κάτι τέτοιο, θα έπρεπε να υπολογιστούν οι περιοχές του κατακερματισμού και στην συνέχεια να τεμαχιστεί η εικόνα με τρόπο που τα μέρη της να έχουν επικαλυπτόμενες περιοχές, προκειμένου να είναι εφικτή η πραγματοποίηση των συνελίξεων με τους συντελεστές Gauss και στην συνέχεια την επανασύνδεσή τους μετά την επεξεργασία. Επίσης η επιμέριση και η επανασυρραφή χρήζουν ειδικών πολύπλοκων χειρισμών της μνήμης με δείκτες που ακόμα και στην παρούσα υλοποίηση φτάνουν να διευθυνσιοδοτούν κομμάτια μνήμης σε 5 διαστάσεις. Ως εκ τούτου, οι υπολογισμοί αυτοί θα προσέθεταν υπολογιστικό φόρτο στο ακολουθιακό τμήμα του αλγορίθμου κάτι που θα είχε ως αποτέλεσμα την εισαγωγή νέων καθυστερήσεων στον αλγόριθμο που ενδεχομένως να αναιρούσαν μερικώς τα οφέλη της επιτάχυνσης. Να σημειωθεί επίσης ότι σε αυτή την περίπτωση θα αυξανόταν σημαντικά και ο χρόνος αντιγραφής και δέσμευσης της μνήμης της κάρτας γραφικών εντός του χρόνου εκτέλεσης των παράλληλων τμημάτων του κώδικα.

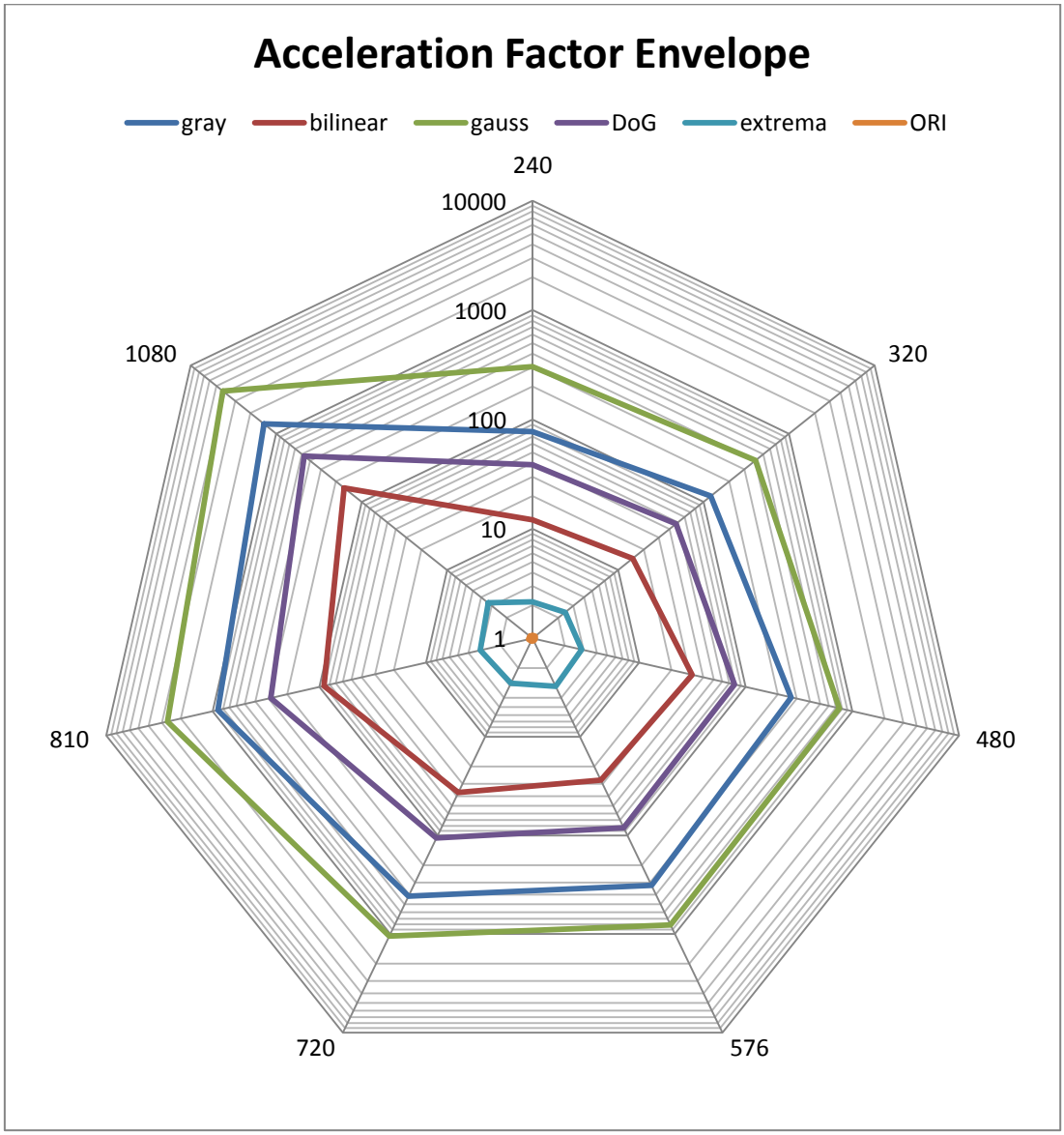
Αντίστοιχη διαδικασία κατακερματισμού και σύνθετης διαχείρισης μνήμης σε μικρότερη κλίμακα, ενδεχομένως να οδηγούσε σε επιπλέον επιτάχυνση του αλγορίθμου. Η διαδικασία αυτή θα λάμβανε χώρο στο επίπεδο της κοινής μνήμης των CUDA blocks προκειμένου να καταστεί εκμεταλλεύσιμη η σημαντικά μεγαλύτερη ταχύτητα επεξεργασίας και προσπέλασης της μνήμης L1 Cache. Με την αρχιτεκτονική της παρούσας υλοποίησης όμως αυτό θα είχε αρνητικά αποτελέσματα, καθώς η ευθεία προσπέλαση στην καθολική μνήμη της κάρτας γραφικών προσφέρει αλγοριθμική απλότητα και εξοικονόμηση χρόνου που αλλιώς θα χανόταν στους επιπλέον υπολογισμούς για την διαχείριση της μνήμης και την προσπέλαση σε αυτή. Επιπροσθέτως, η αναγκαιότητα της επικάλυψης των κατακερματισμένων εικόνων για μεγάλους συνελικτικούς πυρήνες Gauss θα αύξανε σημαντικά την χρήση της μνήμης της κάρτας γραφικών, με πιθανό αποτέλεσμα την αδυναμία της εφαρμογής να επεξεργαστεί δοκίμια υψηλής ανάλυσης.

Για την χρήση του αλγορίθμου αυτού σε ροή βίντεο απαιτούνται ορισμένες τροποποιήσεις, απλοποιήσεις και παραδοχές. Η μνήμη των απαραίτητων μεταβλητών και δυναμικών πινάκων θα πρέπει να δεσμευτεί μία φορά στην αρχικοποίηση του προγράμματος και θα εξαρτάται από τις σταθερές διαστάσεις των εικόνων της ροής. Από το σημείο εκείνο και μετά όμως, η μεταφορά των εικόνων από την μία συνάρτηση στην επόμενη θα γίνεται με απευθείας χειρισμούς των δεικτών της μνήμης γραφικών. Ο παράγοντας επιτάχυνσης που μετρήθηκε στην παρούσα εφαρμογή θα ήταν πλήρως αξιοποιήσιμος και στην περίπτωση βίντεο. Φυσικά μία τέτοια εφαρμογή θα πρέπει να συνδυαστεί με κάποιο άλλο αλγόριθμο μηχανικής μάθησης προκειμένου να γίνεται η αναγνώριση και παρακολούθηση του αντικειμένου σε πραγματικό χρόνο. Κάτι τέτοιο όμως κρίθηκε εκτός των στόχων της παρούσας εργασίας, καθώς αγγίζει σημαντικά διαφορετικούς θεωρητικούς κλάδους.

Για τις πέντε εικόνες που δοκιμάστηκαν, ο μέσος όρος των αντίστοιχων παραγόντων επιτάχυνσης παρουσιάζει την ακόλουθη κατανομή.

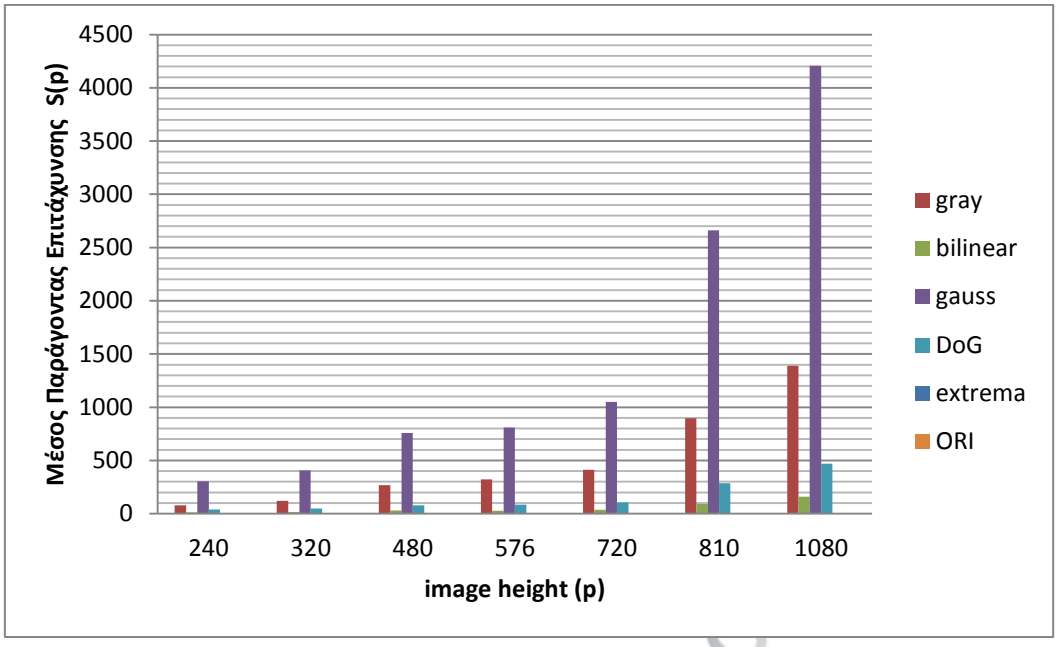
p	*gray*	*bilinear*	*gauss*	*DoG*	*extrema*	*ORI*
240	77.59961	12.12337	303.3109	38.53187	2.157966	1.065433
320	121.5358	14.82833	406.8169	47.51305	2.406422	1.078802
480	266.868	31.42146	756.8473	78.28404	2.894642	1.060882
576	321.3385	27.51012	810.3781	83.86372	3.080446	1.063566
720	412.1947	36.75525	1051.04	106.4776	2.858969	1.072682
810	896.4724	90.29245	2659.958	285.2402	3.095485	1.093776
1080	1389.379	159.8509	4206.655	469.5801	3.302217	1.073784

3-xlii Μέσος παράγοντας επιτάχυνσης ανά μέγεθος εικόνας

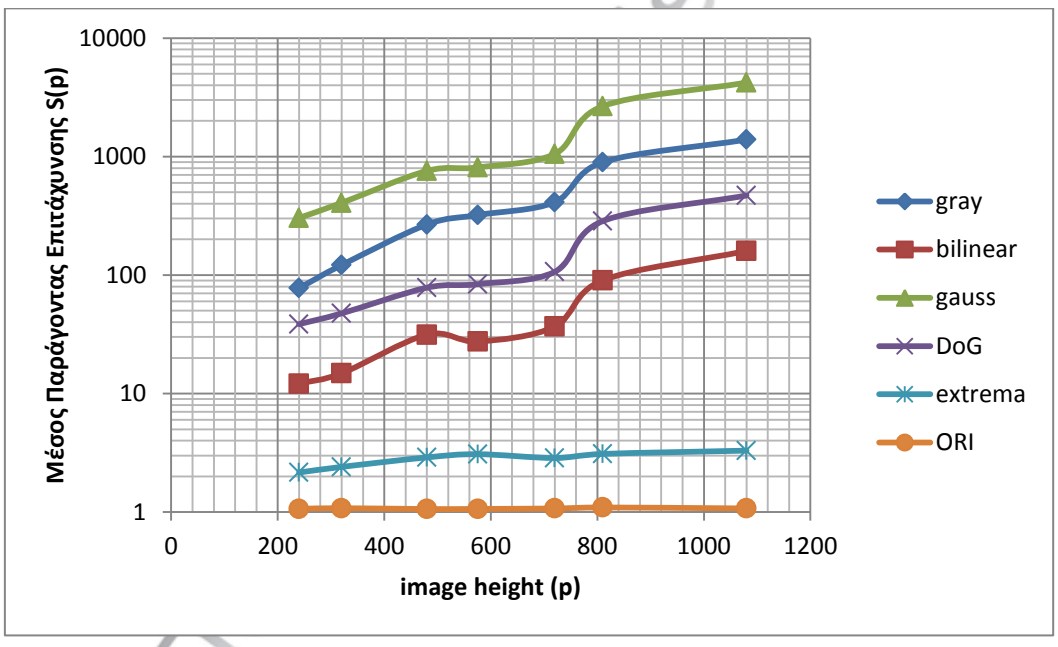


3-xliii Γράφημα Envelope Μέσου Όρου Παράγοντα Επιταχύνσεων (Average Acceleration Factor Enveloope)

Συγκρίνοντας τις επιταχύνσεις του κάθε βήματος του αλγορίθμου σε ένα radar (διακριτό polar) chart, επιβεβαιώνεται και γραφικά με envelope του παράγοντα επιτάχυνσης, δηλαδή ότι οι μεγαλύτερες επιταχύνσεις τείνουν να επιτευχθούν στις μεγάλες αναλύσεις εικόνας, για όλα τα υλοποιημένα στάδια του αλγορίθμου. Είναι ωστόσο παραπάνω από επαρκής ακόμα και στις εικόνες πολύ χαμηλής ανάλυσης.



3-xliv Μέσος όρος Παράγοντα Επιτάχυνσης ανά μέγεθος εικόνας (Average Acceleration Factor per image size)



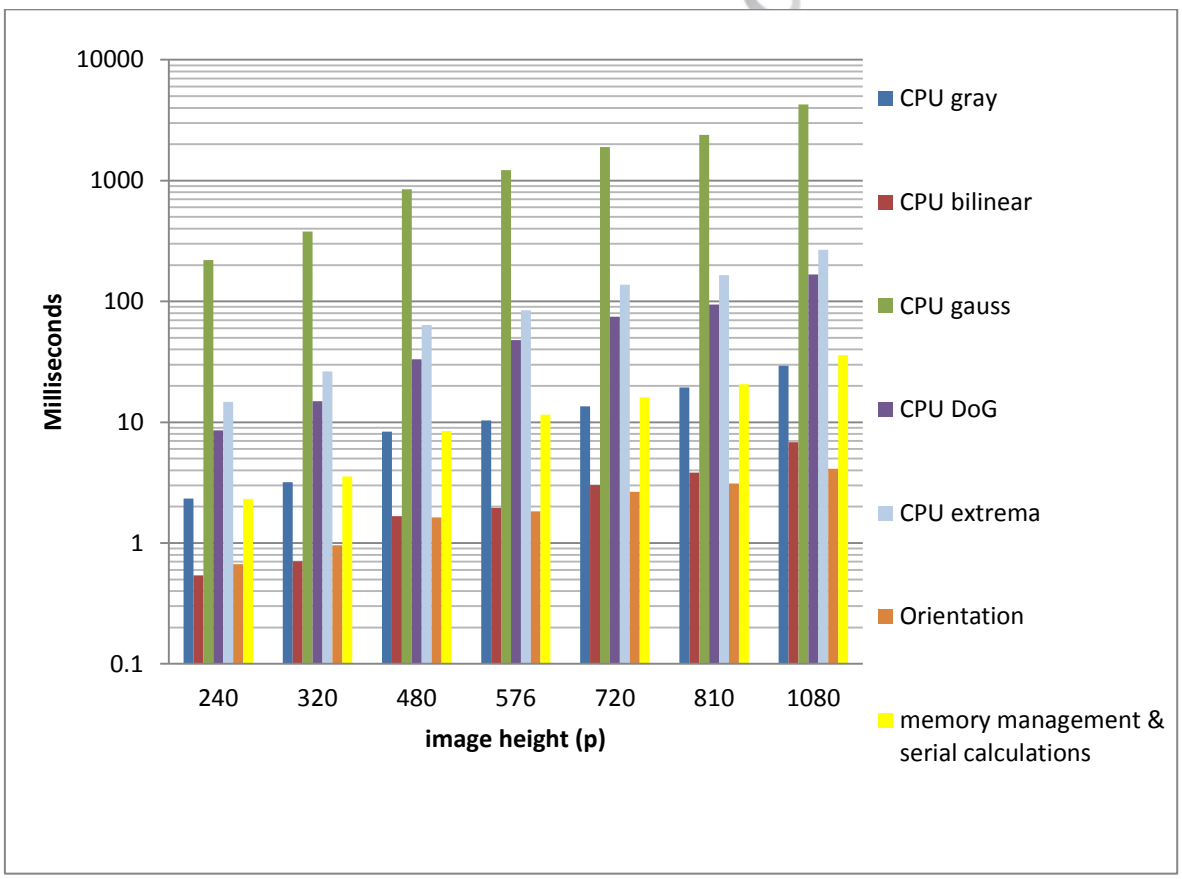
3-xlv Μέσος Όρος Παράγοντα Επιτάχυνσης (Average Acceleration Factor)

Η ελαφρά πτώση του παράγοντα επιτάχυνσης για την συνάρτηση του bilinear interpolation ενδεχομένως να οφείλεται στην κατανομή των νημάτων και των blocks σε συνδυασμό με τους Streaming Multiprocessors για την συγκεκριμένη διάσταση εικόνας και είναι αμελητέα σε σχέση με το κέρδος επιτάχυνσης για τις ίδιες διαστάσεις σε άλλα βήματα του αλγορίθμου. Συνεπώς δεν κρίνεται σκόπιμο να διορθωθεί καθώς αυτό ενδεχομένως να επηρεάσει τον παράγοντα επιτάχυνσης για τις υπόλοιπες συναρτήσεις που έτσι και αλλιώς είναι πιο χρονοβόρες. Υπολογίζοντας τους μέσους όρους των χρονικών διαρκειών της εκτέλεσης των υλοποιημένων συναρτήσεων τόσο για την ακολουθιακή, όσο και για την παράλληλη υλοποίηση του αλγορίθμου, επιβεβαιώνουν την άνιση χρονική κατανομή των βημάτων του αλγορίθμου. Από τις μετρήσεις αυτές γίνεται σαφές ότι η συνολική επιτάχυνση του αλγορίθμου εξαρτάται σε σημαντικά μεγαλύτερο βαθμό από τον παράγοντα επιτάχυνσης των πιο χρονοβόρων συναρτήσεων, όπως φαίνεται και στις σχετικές γραφικές απεικονίσεις που ακολουθούν.

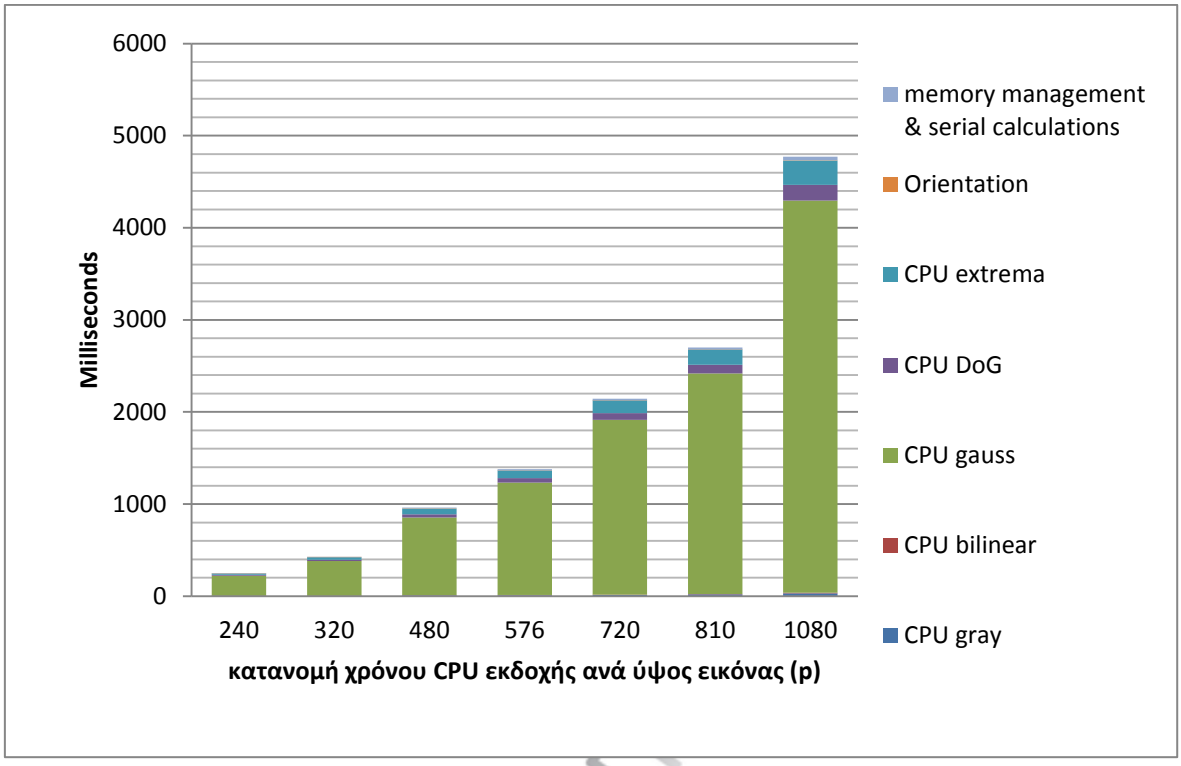
Avg ms	CPU gray	CPU bilinear	CPU gauss	CPU DoG	CPU extrema	Orient.	RAM & math	TOTAL	FPS
240p	2.3396	0.54	220.3384	8.5586	14.778	0.6678	2.3238	249.5462	*4.007274*
320p	3.1806	0.7046	377.4184	14.9414	26.2986	0.959	3.5486	427.0512	*2.34164*
480p	8.359	1.6654	844.9166	33.279	63.7188	1.6322	8.4808	962.0518	*1.039445*
576p	10.36	1.9514	1219.995	47.842	84.3054	1.8316	11.5618	1377.847	*0.72577*
720p	13.5214	3.0224	1896.349	74.574	137.5276	2.6518	16.0694	2143.716	*0.46648*
810p	19.4538	3.8262	2395.523	94.3198	164.7974	3.112	20.6752	2701.707	*0.370136*
1080p	29.4192	6.8156	4260.94	167.353	266.7082	4.1146	35.8136	4771.165	*0.209592*

3-xlvi SIFT: χρόνοι (ms) CPU

Είναι σαφές από τις μετρήσεις των χρόνων εκτέλεσης των επιμέρους τμημάτων του αλγορίθμου ότι οι περισσότερο χρονοβόρες συναρτήσεις είναι αυτές των συνελίξεων με το φίλτρο θόλωσης Gauss, και φυσικά το γεγονός ότι για εκείνες επιτεύχθηκε ο σημαντικότερος παράγοντας επιτάχυνσης όπως παρουσιάστηκε ανωτέρω, δίνει σημαντικό πλεονέκτημα χρόνου στην παράλληλη υλοποίηση.



3-xlvii SIFT: χρόνοι CPU



3-xlviii SIFT: Κατανομή χρόνων CPU ανά μέγεθος εικόνας

Το συμπέρασμα για την ωφέλεια της παράλληλης υλοποίησης στην επιτάχυνση των πιο χρονικά επιβαρυμένων συναρτήσεων και ειδικά των θολώσεων Gauss, επιβεβαιώνεται από τις μετρήσεις χρόνου της παράλληλης εκτέλεσης και την νέα κατανομή του χρόνου εκτέλεσης, όπως φαίνεται στον παρακάτω πίνακα και τις ακόλουθες γραφικές παραστάσεις των δεδομένων του.

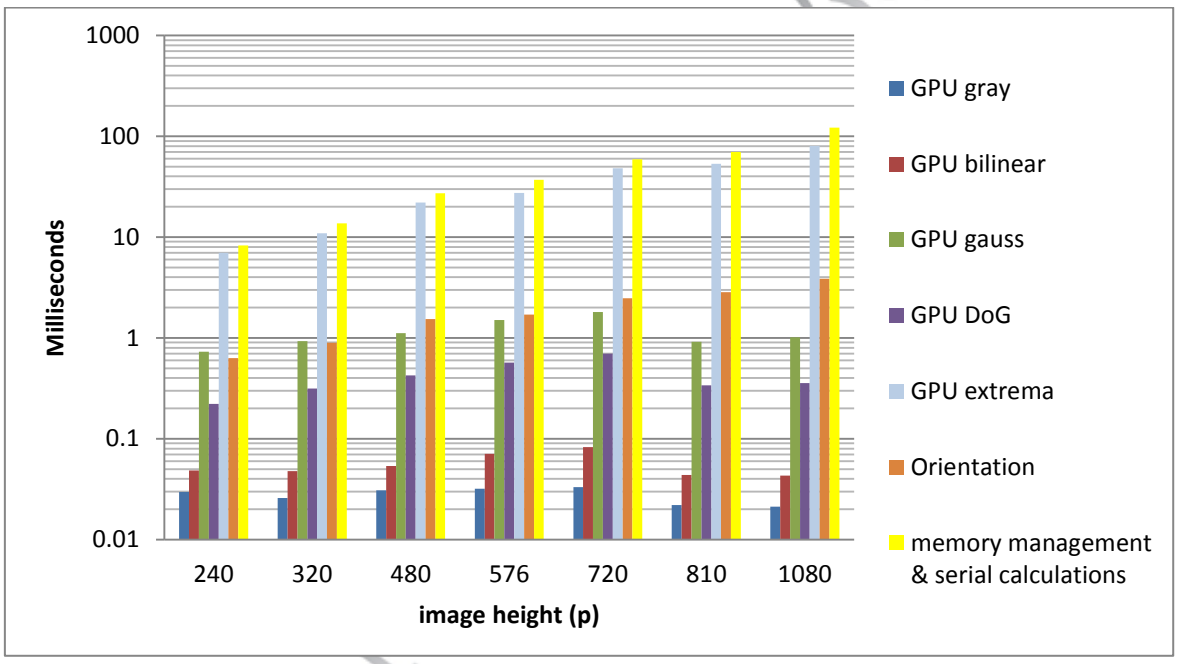
Για τον υπολογισμό των συνολικών παραγόντων επιτάχυνσης συνυπολογίζονται στην παράλληλη εκδοχή του αλγορίθμου οι διάρκειες των τμημάτων που δεν παραλληλοποιήθηκαν αλλά και οι χρόνοι καθυστέρησης λόγω συναρτήσεων διαχείρισης της μνήμης γραφικών. Συγκεκριμένα, οι αρχικές δεσμεύσεις και κινήσεις στην κύρια μνήμη που δεν σχετίζονται με την θεωρητική λειτουργία του αλγορίθμου, όπως οι αναγνώσεις της εικόνας και η μορφοποίηση των δεδομένων της, αποτελούν σταθερά μεγέθη στον υπολογισμό τόσο της ακολουθιακής, όσο και της παράλληλης υλοποίησης.

Οι αρχικές δεσμεύσεις μνήμης της κάρτας γραφικών δεν συνυπολογίζονται επειδή σε μία εφαρμογή του αλγορίθμου σε ροή βίντεο πραγματικού χρόνου, αυτές λαμβάνουν χώρα μόνο μία φορά στην εκκίνηση του προγράμματος κατόπιν προσδιορισμού των διαστάσεων της εικόνας προς επεξεργασία. Καθώς αυτές οι διαστάσεις αυτές παραμένουν σταθερές στην διάρκεια της εκτέλεσης, ο χρόνος αρχικοποίησης τους εξαιρείται από τις μετρήσεις για τον προσδιορισμό του παράγοντα επιτάχυνσης.

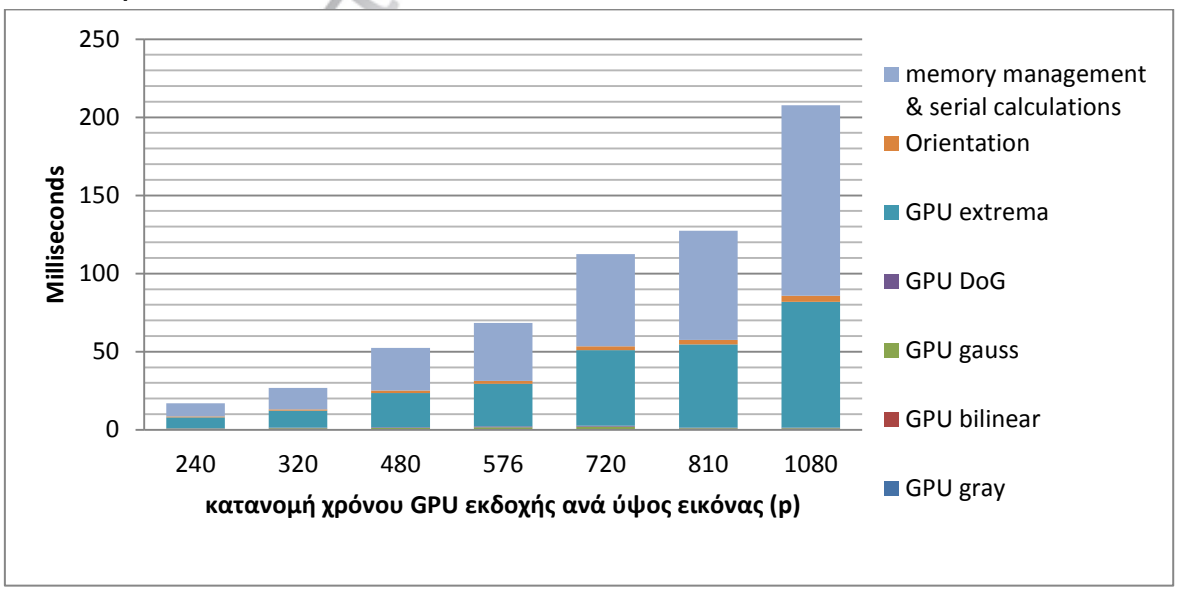
Οι χρόνοι μεταφοράς των δεδομένων από και προς την GPU ωστόσο συνυπολογίζονται στον παράγοντα επιτάχυνσης του συνολικού αλγορίθμου.

Avg ms	GPU gray	GPU bilinear	GPU gauss	GPU DoG	GPU extrema	Orient.	RAM & math	TOTAL	FPS
240p	0.0298	0.0486	0.7292	0.2222	6.9826	0.6292	8.265	16.9066	*59.1485*
320p	0.0258	0.0478	0.9278	0.3146	10.9472	0.893	13.7452	26.9014	*37.17279*
480p	0.0308	0.0538	1.1172	0.4252	21.9622	1.5398	27.2328	52.3618	*19.09789*
576p	0.032	0.071	1.5056	0.5706	27.3776	1.7108	37.0792	68.3468	*14.63126*
720p	0.0332	0.0824	1.8062	0.7008	48.3298	2.4692	59.0792	112.5008	*8.888826*
810p	0.022	0.0438	0.9176	0.3398	53.3722	2.8474	69.8432	127.386	*7.850156*
1080p	0.0212	0.0432	1.0132	0.357	80.5068	3.8564	121.8588	207.6566	*4.815643*

3-xlix SIFT: χρόνοι (ms) GPU



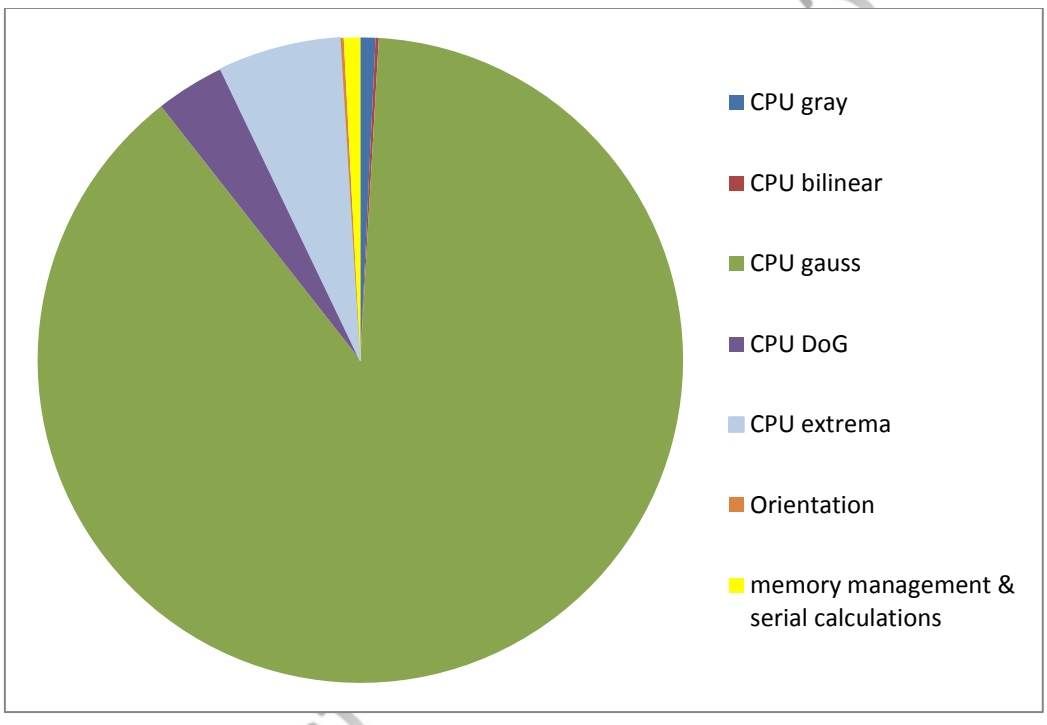
3-l SIFT: Χρόνοι GPU



3-li Κατανομή χρόνων GPU ανά μέγεθος εικόνας

Avg time %	CPU gray	CPU bilinear	CPU gauss	CPU DoG	CPU extrema	Orient.	RAM & math
240p	0.94%	0.22%	88.30%	3.43%	5.92%	0.27%	0.93%
320p	0.74%	0.16%	88.38%	3.50%	6.16%	0.22%	0.83%
480p	0.87%	0.17%	87.82%	3.46%	6.62%	0.17%	0.88%
576p	0.75%	0.14%	88.54%	3.47%	6.12%	0.13%	0.84%
720p	0.63%	0.14%	88.46%	3.48%	6.42%	0.12%	0.75%
810p	0.72%	0.14%	88.67%	3.49%	6.10%	0.12%	0.77%
1080p	0.62%	0.14%	89.31%	3.51%	5.59%	0.09%	0.75%
average	0.75%	0.16%	88.50%	3.48%	6.13%	0.16%	0.82%

3-lii Ποσόστωση χρόνων CPU



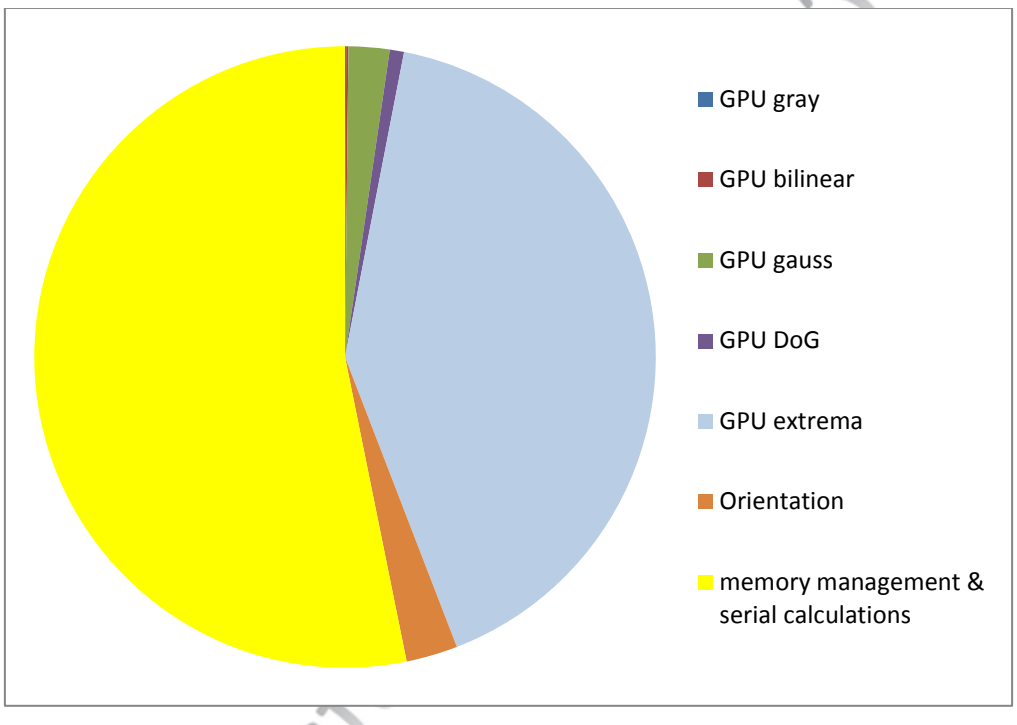
3-liii Μέση κατανομή χρόνου CPU

Η γραφική απεικόνιση της κατανομής του χρόνου εκτέλεσης στην ακολουθιακή εκδοχή του αλγορίθμου επιβεβαιώνει πως το όφελος της επιτάχυνσης θα εξαρτάται σημαντικά από τον παράγοντα επιτάχυνσης του σταδίου της παραγωγής των πυραμίδων Χώρου Κλίμακας Gauss. Σταθερά και για κάθε μέγεθος δοκιμίου, παρατηρείται η ασύμμετρα μεγάλη κατάληψη περισσότερου από το 87% του χρόνου από την συνάρτηση αυτή.

Όπως είναι παρατηρίσιμο στον επόμενο πίνακα και την ακόλουθη γραφική απεικόνιση, ο στόχος της επιτάχυνσης επετεύχθη με σημαντικό βαθμό επιτυχίας καθώς η πλέον χρονοβόρα διεργασία της παραγωγής πυραμίδων Gauss, στην παράλληλη υλοποίηση καταλαμβάνει σταθερά ποσοστό μικρότερο του 4.4% του συνολικού χρόνου εκτέλεσης.

Avg time %	*GPU gray*	*GPU bilinear*	*GPU gauss*	*GPU DoG*	*GPU extrema*	Orient.	*RAM & math*
240p	0.18%	0.29%	4.31%	1.31%	41.30%	3.72%	48.89%
320p	0.10%	0.18%	3.45%	1.17%	40.69%	3.32%	51.09%
480p	0.06%	0.10%	2.13%	0.81%	41.94%	2.94%	52.01%
576p	0.05%	0.10%	2.20%	0.83%	40.06%	2.50%	54.25%
720p	0.03%	0.07%	1.61%	0.62%	42.96%	2.19%	52.51%
810p	0.02%	0.03%	0.72%	0.27%	41.90%	2.24%	54.83%
1080p	0.01%	0.02%	0.49%	0.17%	38.77%	1.86%	58.68%
average	0.06%	0.11%	2.13%	0.74%	41.09%	2.68%	53.18%

3-liv Ποσόστωση χρόνων GPU



3-lv Μέση κατανομή χρόνου GPU

Με βάση αυτά τα δεδομένα πλέον μπορεί να εξαχθεί ασφαλώς το συμπέρασμα για τον παράγοντα επιτάχυνσης του συνολικού αλγορίθμου. Για τον υπολογισμό αυτό χρησιμοποιούνται οι εξισώσεις του Amdahl. Ωστόσο σε αυτές, δεν συμπεριλαμβάνονται οι τεχνικές λεπτομέρειες που αφορούν τον χειρισμό μνήμης σε Ετερογενή συστήματα. Γι αυτόν τον σκοπό οι χρόνοι διαχείρισης, δέσμευσης και μετακίνησης δεδομένων από και προς τις μνήμες της κάρτας γραφικών και του συστήματος, συνυπολογίζονται ως ακολουθιακά μέρη της εκάστοτε εκδοχής του αλγορίθμου.

Οι γενικές μαθηματικές και θεωρητικές αρχές που διέπουν τον προσδιορισμό της επιτάχυνσης σε παράλληλα συστήματα ισχύουν χωρίς παραλλαγή και για το παρόν Ετερογενές σύστημα και η αναμενόμενη μέση επιτάχυνση του αλγορίθμου αναμένεται να είναι προσεγγιστικά γραμμική.

Η μερική επιτάχυνση S_p τμήματος (συνάρτησης) του κώδικα δίνεται από τον λόγο του χρόνου T_s της μη ακολουθιακής (επιταχυμένης εκτέλεσης) προς τον χρόνο T_p της παράλληλης (επιταχυμένης).

$$S_p = \frac{T_s}{T_p}$$

Το πρόγραμμα στην ακολουθιακή του εκδοχή έχει:

- L τμήματα κώδικα που είναι παραλληλίσιμα με χρόνο εκτέλεσης $T_{S_e}(l)$ το καθένα.
- M τμήματα κώδικα με χρόνο εκτέλεσης $T_{S_s}(m)$ έκαστο, που δεν είναι παραλληλίσιμα (σύντομοι μαθηματικοί υπολογισμοί και συναρτήσεις) στα οποία συμπεριλαμβάνονται αρχικοί χειρισμοί της μνήμης που γίνονται υποχρεωτικά στην CPU.
- N τμήματα κώδικα με χρόνο εκτέλεσης $T_{S_m}(n)$ το καθένα που διαχειρίζονται την κύρια μνήμη και αντικαθίστανται από αντίστοιχα στην παράλληλη εκδοχή.

Το πρόγραμμα στην παράλληλη του εκδοχή έχει:

- I τμήματα κώδικα που δεν είναι παραλληλίσιμα με χρόνο εκτέλεσης $T_{P_s}(i)$ το καθένα.
- J τμήματα κώδικα που έχουν παραλληλιστεί που εμπεριέχουν τους αρχικούς χειρισμούς της κύριας μνήμης και έχουν χρόνο εκτέλεσης $T_{P_p}(j)$ έκαστο.
- K τμήματα κώδικα με χρόνο εκτέλεσης $T_{P_m}(k)$ που χειρίζονται την μνήμη της GPU (cudaMalloc, cudaMemcpy).

Έτσι έχουμε τους ακόλουθους υπολογιζόμενους χρόνους εκτέλεσης:

- Συνολικός χρόνος ακολουθιακής εκδοχής του αλγορίθμου:

$$T_{S_o} = \sum_{l=0}^{L} T_{S_e}(l) + \sum_{m=0}^{M} T_{S_s}(m) + \sum_{n=0}^{N} T_{S_m}(n)$$

- Συνολικός χρόνος ακολουθιακής εκδοχής του αλγορίθμου:

$$T_{P_o} = \sum_{i=0}^{I} T_{P_s}(i) + \sum_{j=0}^{J} T_{P_p}(j) + \sum_{k=0}^{K} T_{P_m}(k)$$

Συνεπώς μπορεί να υπολογιστεί βάσει των εξισώσεων προσδιορισμού της επιτάχυνσης, και σε συμφωνία με τα κριτήρια του νόμου Amdahl [92][109], ο συνολικός παράγοντας επιτάχυνσης του αλγορίθμου για κάθε μέγεθος των δοκιμίων (εικόνων).

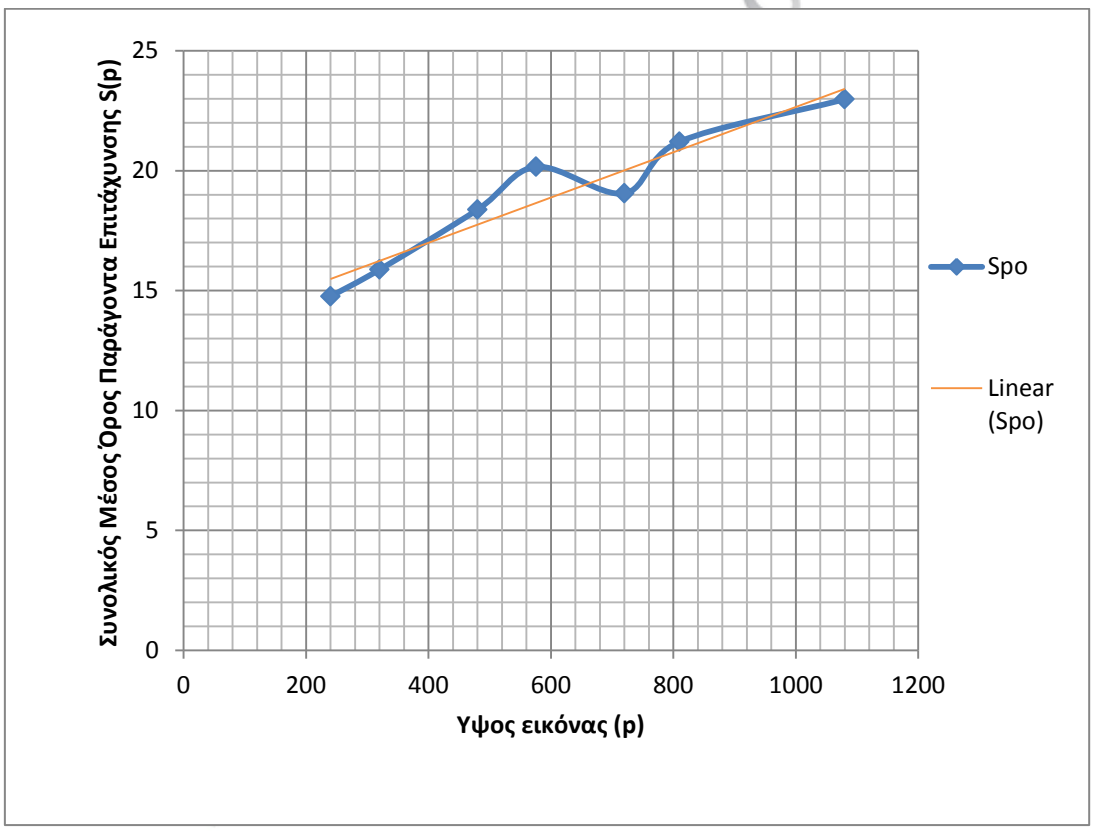
$$S_{P_o} = \frac{T_{S_o}}{T_{P_o}} = \frac{\sum_{l=0}^{L} T_{S_e}(l) + \sum_{m=0}^{M} T_{S_s}(m) + \sum_{n=0}^{N} T_{S_m}(n)}{\sum_{i=0}^{I} T_{P_s}(i) + \sum_{j=0}^{J} T_{P_p}(j) + \sum_{k=0}^{K} T_{P_m}(k)}$$

Οι παράγοντες επιτάχυνσης κατά μέσο όρο ανά μέγεθος εικόνας είναι οι ακόλουθοι:

	S_{P_o}	T_{S_o}	T_{P_o}
240p	14.76028	*249.5462*	*16.9066*
320p	15.87468	*427.0512*	*26.9014*
480p	18.37316	*962.0518*	*52.3618*
576p	20.15964	*1377.847*	*68.3468*
720p	19.05512	*2143.716*	*112.5008*
810p	21.20882	*2701.707*	*127.386*
1080p	22.97622	*4771.165*	*207.6566*

3-lvi Μ.Ο. Παράγοντας επιτάχυνσης και χρόνοι εκτέλεσης ανά μέγεθος εικόνας

Γραφικά η επιτάχυνση του συνολικού αλγορίθμου στην τρέχουσα παραλληλισμένη μορφή του παρίσταται ακολούθως και προσεγγίζει την γραμμικότητα:



3-lvii Συνολικός Μέσος Όρος Παράγοντα Επιτάχυνσης S(p) ανά μέγεθος εικόνας

Καταλήγοντας, παρατηρούμε ότι ο μέσος όρος του παράγοντα επιτάχυνσης είναι σε όλα τα μεγέθη εικόνας ανώτερος του 14 και μέχρι 23. Οι επιταχύνσεις αυτές είναι απόδειξη της επίτευξης του στόχου της παράλληλης εκδοχής του αλγορίθμου και της ωφέλειας της χρήσης τεχνολογιών παράλληλης επεξεργασίας και Γενικού Σκοπού Προγραμματισμού Επεξεργαστών Γραφικών σε Ετερογενή συστήματα τα οποία υλοποιούν εφαρμογές Μηχανικής Όρασης.

4. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] E. Olmedo, J. De La Calleja, A. Benitez, M. A. Medina, and L. de P. por Computadora, "Point to point processing of digital images using parallel computing," *Int. J. Comput. Sci. Issues*, vol. 9, no. 33, p. 3, 2012.

[2] H. Xiao, Y.-P. Song, and Q.-L. Zhou, "Multi-GPU Accelerated Parallel Algorithm of Wallis Transformation for Image Enhancement," *Int. J. Grid Distrib. Comput.*, vol. 7, no. 2, pp. 99–114, 2014.

[3] V. Prisacariu and I. Reid, "fastHOG-a real-time GPU implementation of HOG," *Dep. Eng. Sci.*, vol. 2310, 2009.

[4] M. Hirabayashi, S. Kato, M. Edahiro, K. Takeda, T. Kawano, and S. Mita, "GPU implementations of object detection using HOG features and deformable models," in *Cyber-Physical Systems, Networks, and Applications (CPSNA), 2013 IEEE 1st International Conference on*, 2013, pp. 106–111.

[5] B. A. Zalesky and E. N. Seredin, "Real Time Object Tracking Algorithm," 2013.

[6] D. Oro, C. Fernández, J. R. Saeta, X. Martorell, and J. Hernando, "Real-time GPU-based face detection in HD video sequences," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 530–537.

[7] E. Apostolidis, V. Mezaris, and I. Kompatsiaris, "Fast object re-detection and localization in video for spatio-temporal fragment creation," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, 2013, pp. 1–6.

[8] W. Yan, X. Shi, X. Yan, and L. Wang, "Computing OpenSURF on OpenCL and General Purpose GPU," 2013.

[9] J. Marzat, Y. Dumortier, and A. Ducrot, "Real-time dense and accurate parallel optical flow using CUDA," 2009.

[10] J. Fung and S. Mann, "Using graphics devices in reverse: GPU-based image processing and computer vision," in *Multimedia and Expo, 2008 IEEE International Conference on*, 2008, pp. 9–12.

[11] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with OpenCV," *Commun. ACM*, vol. 55, no. 6, pp. 61–69, 2012.

[12] Y. Allusse, P. Horain, A. Agarwal, and C. Saipriyadarshan, "GpuCV: A GPU-accelerated framework for image processing and computer vision," in *Advances in Visual Computing*, Springer, 2008, pp. 430–439.

[13] H. Sun, L. Yan, P. Mooney, and R. Liang, "A new method for moving object detection using variable resolution bionic compound eyes," *Int. J. Phys. Sci.*, vol. 6, no. 24, pp. 5618–5622, 2011.

[14] H. Sedding, F. Deger, H. Dammertz, J. Bouecke, and H. P. A. Lensch, “Massively Parallel Multiclass Object Recognition.,” in *VMV*, 2010, pp. 251–257.

[15] J. Fasola and M. Veloso, “Real-time object detection using segmented and grayscale images,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 4088–4093.

[16] J. P. D’Amato, F. Mayorano, A. Rubiales, J. Massa, and P. Tristan, “Migrating multispectral image processing to the GPU. 38^{o},” in *Jornadas Argentinas de Informática e Investigación Operativa, High-Performance Computing Symposium*, 2009, vol. 38, pp. 1–11.

[17] R. Juránek, M. Hradiš, and P. Zemčík, “Real-time Algorithms of Object Detection using Classifiers,” *Real-Time Syst.*, pp. 1–22, 2012.

[18] M. Kaur and Hempriya, “Comparison of Various Algorithms used for Object Detection.,” in *International Journal of Engineering Research and Applications (IJERA). National Conference on Advances in Engineering and Technology*, 2014.

[19] J. C. S. Miguel and J. M. Martínez, “Robust unattended and stolen object detection by fusing simple algorithms,” in *Advanced Video and Signal Based Surveillance, 2008. AVSS’08. IEEE Fifth International Conference on*, 2008, pp. 18–25.

[20] R. Campbell and J. Krumm, “Object recognition for an intelligent room,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, vol. 1, pp. 691–697.

[21] G. Louloudis, B. Gatos, I. Pratikakis, and K. Halatsis, “A block-based Hough transform mapping for text line detection in handwritten documents,” in *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

[22] S. W. Smith, “The scientist and engineer’s guide to digital signal processing,” 1997.

[23] L. G. Shapiro and G. C. Stockman, “Computer Vision: Theory and Applications. 2001.” Prentice Hall.

[24] B. Heflin, B. Parks, W. Scheirer, and T. Boult, “Single image deblurring for a real-time face recognition system,” in *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, 2010, pp. 1185–1192.

[25] M. Strengert, M. Kraus, and T. Ertl, “Pyramid methods in gpu-based image processing,” *Proc. Vision, Model. Vis. 2006*, pp. 169–176, 2006.

[26] A. C. Bovik, *Handbook of image and video processing*. Academic press, 2010.

[27] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Pearson Education India, 2004.

[28] Ι. Πανάρετος and Ε. Ξεκαλάκη, *Εισαγωγή στη Στατιστική σκέψη, Τόμος ΙΙ*. 2000.

[29] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 24, no. 4, pp. 509–522, 2002.

[30] J. Canny, “A computational approach to edge detection,” *Pattern Anal. Mach. Intell. IEEE Trans.*, no. 6, pp. 679–698, 1986.

[31] L. Shapiro and R. Haralick, “Computer and robot vision,” *Read. Addison-Wesley*, vol. 8, 1992.

[32] D. Ioannou, W. Huda, and A. F. Laine, “Circle recognition through a 2D Hough transform and radius histogramming,” *Image Vis. Comput.*, vol. 17, no. 1, pp. 15–26, 1999.

[33] S. Inverso, “Ellipse detection using randomized Hough transform,” *Final Proj. Introd. to Comput. Vis.*, pp. 4005–4757, 2002.

[34] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[35] C. Popescu, “A Contour Based Descriptor for Object Recognition,” *SACI Trans. Timisoara*, 2004.

[36] T. Gevers, J. Van De Weijer, and H. Stokman, “Color feature detection,” *Color image Process. methods Appl.*, vol. 9, pp. 203–226, 2006.

[37] S. Rege, R. Memane, M. Phatak, and P. Agarwal, “2D geometric shape and color recognition using digital image processing,” *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 2, no. 6, pp. 2479–2487, 2013.

[38] J. C. Nascimento and J. S. Marques, “Performance evaluation of object detection algorithms for video surveillance,” *Multimedia, IEEE Trans.*, vol. 8, no. 4, pp. 761–774, 2006.

[39] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision.,” in *IJCAI*, 1981, vol. 81, pp. 674–679.

[40] X. Li and Y. Wu, “Image Object Detection Algorithm Based on Improved Gaussian Mixture Model,” *J. Multimed.*, vol. 9, no. 1, pp. 152–158, 2014.

[41] D. Crandall and J. Luo, “Robust color object detection using spatial-color joint probability functions,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 1, pp. I–379.

[42] M. Björkman, N. Bergström, and D. Kragic, “Detecting, segmenting and tracking unknown objects using multi-label MRF inference,” *Comput. Vis. Image Underst.*, vol. 118, pp. 111–127, 2014.

[43] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter, “Comparison of view-based object recognition algorithms using realistic 3D models,” in *Artificial Neural Networks—ICANN 96*, Springer, 1996, pp. 251–256.

[44] Y. Boykov and D. P. Huttenlocher, “A new bayesian framework for object recognition,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2.

[45] F. Yin, D. Makris, and S. A. Velastin, “Performance evaluation of object tracking algorithms,” in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil*, 2007.

[46] H. P. Moravec, “Visual mapping by a robot rover,” in *Proceedings of the 6th international joint conference on Artificial intelligence-Volume 1*, 1979, pp. 598–600.

[47] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Alvey vision conference*, 1988, vol. 15, p. 50.

[48] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, 1999, vol. 2, pp. 1150–1157.

[49] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[50] K. Mikolajczyk and C. Schmid, “An affine invariant interest point detector,” in *Computer Vision—ECCV 2002*, Springer, 2002, pp. 128–142.

[51] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 24, no. 5, pp. 603–619, 2002.

[52] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 8, pp. 888–905, 2000.

[53] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 1995, pp. 694–699.

[54] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 8, pp. 747–757, 2000.

[55] N. M. Oliver, B. Rosario, and A. P. Pentland, “A Bayesian computer vision system for modeling human interactions,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 22, no. 8, pp. 831–843, 2000.

[56] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, vol. 1, pp. 255–261.

[57] J. Zhong and S. Sclaroff, “Segmenting foreground objects from a dynamic textured background via a robust kalman filter,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 44–50.

[58] C. P. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Computer vision, 1998. sixth international conference on*, 1998, pp. 555–562.

[59] H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 20, no. 1, pp. 23–38, 1998.

[60] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–511.

[61] A. Patel, D. R. Kasat, S. Jain, and V. M. Thakare, “Performance Analysis of Various Feature Detector and Descriptor for Real-Time Video based Face Tracking,” *Int. J. Comput. Appl.*, vol. 93, no. 1, 2014.

[62] P. M. Panchal, S. R. Panchal, and S. K. Shah, “A comparison of SIFT and SURF,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 2, pp. 323–327, 2013.

[63] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Found. Trends® Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, 2008.

[64] M. Guerrero, “A comparative study of three image matcing algorithms: SIFT, SURF, and Fast,” 2011.

[65] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2548–2555.

[66] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 510–517.

[67] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: an efficient alternative to SIFT or SURF,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564–2571.

[68] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer vision–ECCV 2006*, Springer, 2006, pp. 404–417.

[69] S. A. Yaseen and S. Sasi, “Robust Algorithm for Object Detection and Tracking in a Dynamic Scene,” *J. Image Graph.*, vol. 2, no. 1, 2014.

[70] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 19, no. 7, pp. 780–785, 1997.

[71] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Computer Vision—ECCV 2000*, Springer, 2000, pp. 751–767.

[72] L. Li and M. K. H. Leung, “Integrating intensity and texture differences for robust change detection,” *Image Process. IEEE Trans.*, vol. 11, no. 2, pp. 105–112, 2002.

[73] J. Rittscher, J. Kato, S. Joga, and A. Blake, “A probabilistic background model for tracking,” in *Computer Vision—ECCV 2000*, Springer, 2000, pp. 336–350.

[74] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann, “Topology free hidden markov models: Application to background modeling,” in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, vol. 1, pp. 294–301.

[75] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, “Background modeling and subtraction of dynamic scenes,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 1305–1312.

[76] V. Zeljković, Ž. Trpovski, and V. Šenk, “New algorithm for moving object detection,” *Yugosl. J. Oper. Res. ISSN 0354-0243 EISSN 2334-6043*, vol. 14, no. 1, 2013.

[77] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, 2004.

[78] S. Rowe and A. Blake, “Statistical mosaics for tracking,” *Image Vis. Comput.*, vol. 14, no. 8, pp. 549–564, 1996.

[79] M. Irani and P. Anandan, “Video indexing based on mosaic representations,” *Proc. IEEE*, vol. 86, no. 5, pp. 905–921, 1998.

[80] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation,” *Pattern Anal. Mach. Intell. IEEE Trans.*, vol. 15, no. 11, pp. 1101–1113, 1993.

[81] X. H. Fang, W. Xiong, B. J. Hu, and L. T. Wang, “A moving object detection algorithm based on color information,” in *Journal of Physics: Conference Series*, 2006, vol. 48, no. 1, p. 384.

[82] N. Paragios and R. Deriche, “Geodesic active regions and level set methods for supervised texture segmentation,” *Int. J. Comput. Vis.*, vol. 46, no. 3, pp. 223–247, 2002.

[83] Š. Obdržálek, J. Matas, and O. Chum, “On the interaction between object recognition and colour constancy,” 2003.

[84] L. Grewe and A. C. Kak, “Interactive learning of a multiple-attribute hash table classifier for fast object recognition,” *Comput. Vis. Image Underst.*, vol. 61, no. 3, pp. 387–416, 1995.

[85] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[86] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[87] A. Coates, P. Baumstarck, Q. Le, and A. Y. Ng, “Scalable learning for object detection with GPU hardware,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 4287–4293.

[88] M. Stojmenovic, “Algorithms for real-time object detection in images,” *Handb. Appl. algorithms’(Wiley, 2008)*, pp. 317–346, 2007.

[89] C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, “Motion estimation through efficient matching of a reduced number of reliable singular points,” in *Proc. SPIE*, 2008, vol. 6811, p. 68110N.

[90] C. J. Veenman, E. A. Hendriks, and M. J. T. Reinders, “A fast and robust point tracking algorithm,” in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, 1998, pp. 653–657.

[91] I. N. Έλληνας, *Αρχιτεκτονική Υπολογιστών - Οικογένεια Επεξεργαστών 80X86*. Έλληνας, I N, 2007.

[92] Χαράλαμπος Κωνσταντόπουλος, “Ειδικά Θέματα Αλγορίθμων,” 2011.

[93] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, “GPU computing,” *Proc. IEEE*, vol. 96, no. 5, pp. 879–899, 2008.

[94] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard, “Cg: A system for programming graphics hardware in a C-like language,” in *ACM Transactions on Graphics (TOG)*, 2003, vol. 22, no. 3, pp. 896–907.

[95] J. X. Chen, “OpenGL Shading Language,” in *Guide to Graphics Software Tools*, Springer, 2009, pp. 1–29.

[96] I. Buck, T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston, and P. Hanrahan, “Brook for GPUs: stream computing on graphics hardware,” in *ACM Transactions on Graphics (TOG)*, 2004, vol. 23, no. 3, pp. 777–786.

[97] B. Zhang, S. Xu, F. Zhang, Y. Bi, and L. Huang, “Accelerating matlab code using gpu: A review of tools and strategies,” in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, 2011, pp. 1875–1878.

[98] C. Nvidia, “Programming guide.” 2008.

[99] K. O. W. Group, “The opencl specification,” *version*, vol. 1, no. 29, p. 8, 2008.

[100] C. Bruyns and B. Feldman, “Image processing on the GPU: A canonical example,” *Proj. Berkeley*, 2003.

[101] M. C. Roldan, M. Naiouf, and A. E. De Giusti, “Parallelizing tracking algorithms,” *J. Comput. Sci. Technol.*, vol. 1, 2002.

[102] G. Bradski, “Dr. dobb’s journal of software tools,” 2000.

[103] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[104] Z. Ye and D. She, “Simplified SIFT C implementation,” *Eidnhoven Polytechnic University*, 2011. [Online]. Available: https://sites.google.com/site/5kk73gpu2011/assignments/sift/sift-c.zip?attredirects=0.

[105] J. Cheng, M. Grossman, and T. McKercher, *Professional Cuda C Programming*. John Wiley & Sons, 2014.

[106] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, “A Comparative Study of SIFT and its Variants,” *Meas. Sci. Rev.*, vol. 13, no. 3, pp. 122–131, 2013.

[107] V. Podlozhnyuk, “Image convolution with CUDA,” *NVIDIA Corp. white Pap. June*, vol. 2097, no. 3, 2007.

[108] G. Agam, “Introduction to programming with OpenCV,” *Online Doc.*, vol. 27, 2006.

[109] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2011.

[110] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. “ O’Reilly Media, Inc.”

[111] Dijkstra, T., Milan, S., Vaclav, H., & Roger, B. (1994). Image processing, analysis and machine vision. London: Elsevier.

[112] Gevers, T., Weijer, J. V. D., & Stokman, H. (2006). Color Image Processing: Emerging Applications. CRC Press, Boca Raton, FL.

[113] Jain, R., Kasturi, R., & Schunck, B. G. (1995). Machine vision, vol. 5. McGraw-Hill, New York.

[114] Myler, H. R., & Weeks, A. R. (2009). *The pocket handbook of image processing algorithms in C*. Prentice Hall Press.

[115] Nayak, A., & Stojmenovic, I. (2007). *Handbook of applied algorithms: Solving scientific, engineering, and practical problems*. John Wiley & Sons.

[116] Wang, L.-T., Chang, Y.-W., & Cheng, K.-T. T. (2009). *Electronic design automation: synthesis, verification, and test*. Morgan Kaufmann.

[117] Farmer, W. M. (2013). The formalization of syntax-based mathematical algorithms using quotation and evaluation. In *Intelligent Computer Mathematics* (pp. 35–50). Springer.

[118] Knuth, D. E. (1970). The analysis of algorithms. In *Actes du congres international des Mathématiciens*