



Χανιά, Δεκέμβριος 2003

Διπλωματική Εργασία

«Συνδυασμός πληροφορίας έντασης και κίνησης για εντοπισμό
αντικειμένων με εφαρμογές στο MPEG-4 και 7»

Τσίχλας Αντώνης

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας μου κ. Ζερβάκη Μιχάλη, για την συνεργασία και την ιδιαίτερη κατανόηση του όλη τη διάρκεια της εργασίας αυτής. Η καθοδήγησή του ήταν πάντοτε το φως στο μονοπάτι της έρευνας. Επίσης θα ήθελα να ευχαριστήσω όλους εκείνους (και ιδιαίτερα την συγκάτοικό μου) που λίγο – πολύ με υποστήριξαν σε όλη μου την πορεία στο Πολυτεχνείο, που χωρίς αυτούς τίποτα δεν θα ήταν εύκολο. Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου για τα πράγματα αυτά που μου προσέφεραν και είναι αδύνατο να περιγράψω, καθώς και για την πίστη τους σε εμένα.

Περιεχόμενα

Περιεχόμενα	3
Πίνακας σχημάτων	4
Περίληψη	5
Εισαγωγή.....	6
Βασικά σημεία του Mpeg4 και Mpeg7	8
Γενικά	8
Διανύσματα κίνησης (motion vectors).....	8
Αντικείμενα	9
Οπτική ροή	12
Γενικά	12
Μέθοδοι υπολογισμού οπτικής ροής.....	13
Υπολογισμός της οπτικής ροής σύμφωνα με Horn – Schunck.....	14
Υπάρχουσες μέθοδοι.....	19
Εύρεση ακμών κίνησης	21
Πρόλογος.....	21
Μέθοδος διανυσμάτων κίνησης	21
Μέθοδος πρώτης παραγώγου	22
Συνδυασμός διανυσμάτων κίνησης και πρώτης παραγώγου	23
Εύρεση στατικών ακμών	25
Διαχωρισμός τοπικά στατικών ακμών.....	27
Πειραματική εφαρμογή	33
Συμπεράσματα - Μελλοντικές επεκτάσεις.....	45
Παραπομπές	47

Πίνακας σχημάτων

Σχήμα 1.	Αντιστοίχιση του video στους άξονες	6
Σχήμα 2.	Πιθανή αλληλουχία καρέ μεταξύ δυο I καρέ	9
Σχήμα 3.	(α) Το αρχικό καρέ, (β) Το background της σκηνής και (γ)(δ) τα αντικείμενα που περιέχει το καρέ	10
Σχήμα 4.	Η οπτική ροή (γ) μεταξύ των καρέ (α) και (β)	12
Σχήμα 5.	(α) Περιστρεφόμενος κύλινδρος, (β) η οπτική ροή και (γ) τα διανύσματα κίνησης	12
Σχήμα 6.	(α)(β) Παράδειγμα από το γνωστό video Hall Monitor σε δυο διαδοχικά καρέ (γ) Ο μέσος όρος των διανυσμάτων κίνησης	22
Σχήμα 7.	Η πρώτη παράγωγος στο χρόνο	23
Σχήμα 8.	Κινούμενες ακμές	24
Σχήμα 9.	Η εφαρμογή των масκών του φίλτρου Kirsch στο καρέ 50 (α) για (β) 0° , (γ) 90° , (δ) 45° , (ε) 135° και (ζ) το συνολικό αποτέλεσμα	26
Σχήμα 10.	(α) Στατική πληροφορία (background) και (β) ακμές στατικής πληροφορίας	28
Σχήμα 11.	Ακίνητες ακμές στιγμιότυπου	28
Σχήμα 12.	Στατικές ακμές στιγμιότυπου	29
Σχήμα 13.	Ακίνητες ακμές αντικειμένου	30
Σχήμα 14.	Ολοκληρωμένη ανίχνευση ακμών	30
Σχήμα 15.	Οι ακμές με μειωμένο θόρυβο	31
Σχήμα 16.	Σχεδιάγραμμα της διαδικασίας εύρεσης των ακμών	32
Σχήμα 17.	Τα αρχικά καρέ 45-46 της σκηνής και το αποτέλεσμα	36
Σχήμα 18.	Τα αρχικά καρέ 112-113 της σκηνής και το αποτέλεσμα	37
Σχήμα 19.	Τα αρχικά καρέ 261-262 της σκηνής και το αποτέλεσμα	38
Σχήμα 20.	Τα αρχικά καρέ 45 - 46 της σκηνής και το αποτέλεσμα	39
Σχήμα 21.	Τα αρχικά καρέ 174 - 175 της σκηνής και το αποτέλεσμα	40
Σχήμα 22.	Τα αρχικά καρέ 219 - 220 της σκηνής και το αποτέλεσμα	41
Σχήμα 23.	Τα αρχικά καρέ 89 - 90 της σκηνής και το αποτέλεσμα	42
Σχήμα 24.	Τα αρχικά καρέ 211 – 212 της σκηνής και το αποτέλεσμα	43

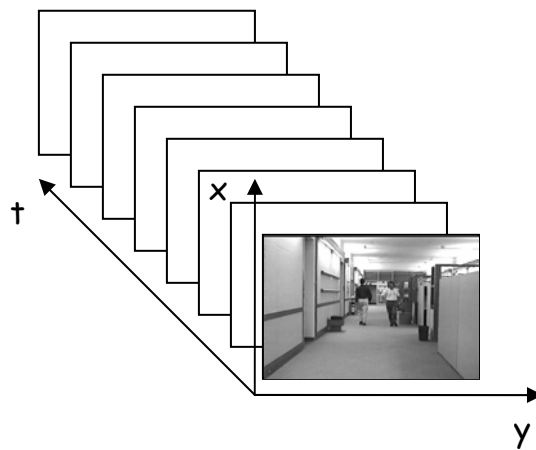
Περίληψη

Για την σωστή και αποδοτική μετατροπή του video σε MPEG-4 και 7 χρειάζεται ένας προσαρμόσιμος και ευέλικτος τρόπος εύρεσης των αντικειμένων που εμφανίζονται σε αυτό. Στην εργασία αυτή προτείνονται τρόποι που εκμεταλλεύονται τόσο την χωρική πληροφορία, όσο και την κίνηση που ανιχνεύεται ανάμεσα στα καρέ του video. Η κίνηση των αντικειμένων χρησιμοποιείται δια μέσου της πρώτης παραγώγου στο χρόνο και της οπτικής ροής. Εντοπίστηκε βέβαια η αδυναμία αυτών των μεθόδων να ανιχνεύσουν το περίγραμμα των αντικειμένων που προσωρινά δεν κινείται ή είναι παράλληλο στον άξονα κίνησης. Τα σημεία αυτά του περιγράμματος έχουν την ιδιότητα ότι δεν εμφανίζονται σε ολόκληρη τη σκηνή και ότι παραμένουν ακίνητα μονάχα προσωρινά. Από την ιδιότητά τους αυτή μπορούμε να τα απομονώσουμε. Έτσι η χρήση της χωρικής πληροφορίας έρχεται να συμπληρώσει το περίγραμμα των κινούμενων αντικειμένων.

Για τη χρήση της χωρικής πληροφορίας θεωρήθηκε ότι η κάμερα είναι ακίνητη. Αυτό συμβαίνει προκειμένου να απλοποιηθούν κάποιες διαδικασίες, καθώς μια φυσική εικόνα είναι αρκετά πολύπλοκη για να αναλυθεί με κάποιο μαθηματικό μοντέλο. Η μέθοδος που προτείνουμε χρησιμοποιήθηκε επιτυχώς, και σε συνδυασμό με τεχνικές ανίχνευση κίνησης της κάμερας θα μπορούσε να αποτελεί ολοκληρωμένη λύση προκειμένου να χρησιμοποιηθεί σε φυσικές σκηνές.

Εισαγωγή

Σκοπός της διπλωματικής αυτής είναι η εύρεση των αντικειμένων που περιέχονται στο video για την χρήση τους από τα πρότυπα συμπίεσης Mpeg4 και Mpeg7. Ως video ορίζεται μια ροή από εικόνες (x, y) στο χρόνο (t) . Οι εικόνες λέγονται καρέ του video και είναι η διδιάστατη διακριτή αναπαράσταση του φυσικού χώρου (Σχήμα 1).



Σχήμα 1. Αντιστοίχιση του video στους άξονες

Σκοπός των προτύπων Mpeg είναι να περιγράψουν τον τρόπο με τον οποίο θα είναι αποθηκευμένη η πληροφορία του video στο μέσο αποθήκευσης μιας ψηφιακής συσκευής (Οπτικός δίσκος, Σκληρός Δίσκος κ. α.). Τα πρότυπα αυτά χρησιμοποιούνται για να επιτευχθεί συμπίεση του video. Η συμπίεση του video είναι απαραίτητη προκειμένου να υπάρχει επάρκεια χώρου στα υπάρχοντα αποθηκευτικά συστήματα (CD,DVD) και να είναι δυνατή η μετάδοση δημιουργώντας όσο το δυνατό λιγότερη κίνηση στο κανάλι επικοινωνίας. Ως κανάλι επικοινωνίας θεωρείται οποιοδήποτε δίκτυο υπολογιστών ή συσκευών δια μέσου του οποίου θα γίνει η μετάδοση του video.

Η περιγραφή του συμπιεσμένου κατά Mpeg4 και Mpeg7 video δεν επιβάλλει συγκεκριμένο τρόπο για την υλοποίηση της συμπίεσης. Ανάλογα με

την φύση του video καθώς και με τις απαιτήσεις του κατασκευαστή σε χώρο, επεξεργαστική ισχύ του hardware και ευκρίνεια υπάρχουν διάφορες παράμετροι που πρέπει να προσδιοριστούν.

Γενικός στόχος της εξέλιξης νέων τρόπων και μεθόδων που υπακούουν στα παραπάνω πρότυπα είναι η μείωση του απαιτούμενου αποθηκευτικού χώρου και η ταχύτερη μετατροπή ανάμεσα στα πρότυπα ιδίως στο Mpeg4 που επικρατεί την χρονική στιγμή συγγραφής αυτής της μελέτης και στο Mpeg7 που είναι το μελλοντικό πρότυπο, όταν τόσο η τεχνολογία αλλά και η τεχνογνωσία το επιτρέψουν.

Στόχος της μελέτης αυτής είναι η πιο αποτελεσματική αναζήτηση των αντικειμένων που υπάρχουν στα καρέ του video, ούτως ώστε να επιτευχθεί μεγαλύτερη και πιο ποιοτική συμπίεση σύμφωνα με τα παραπάνω πρότυπα. Η γνώση των αντικειμένων που απαρτίζουν ένα στιγμιότυπο ενός video είναι μια χρήσιμη πληροφορία. Συγκεκριμένα, μια προσέγγιση των κινούμενων αντικειμένων αρκεί για σωστή συμπίεση. Η βασική ιδέα είναι ότι αντικείμενο σε μια εικόνα είναι οτιδήποτε κινείται, ενώ οτιδήποτε ακίνητο θεωρείται στατική πληροφορία και αντιμετωπίζεται ξεχωριστά από τον αλγόριθμο συμπίεσης. Για αυτό το λόγο αναζητούμε περιοχές στο video που έχουν υψηλή κίνηση (υψηλή ταχύτητα) και προσπαθούμε να τις απομονώσουμε αποτελεσματικά από την υπόλοιπη εικόνα. Για το σκοπό αυτό χρησιμοποιούμε τη χωρική πληροφορία που έχουμε στη διάθεσή μας οριοθετώντας τις περιοχές στην εικόνα (χωρική τμηματοποίηση) καθώς και την χρονική πληροφορία της κίνησης της εικόνας για να εξάγουμε μια δεύτερη τμηματοποίηση (τμηματοποίηση κίνησης). Ο συνδυασμός τους θα μας δώσει μια πιο ακριβή προσέγγιση των αντικειμένων που υπάρχουν και κινούνται σε ένα video, βασικό σημείο στο οποίο στηρίζονται τα πρότυπα συμπίεσης Mpeg4 και 7.

Βασικά σημεία του Mpeg4 και Mpeg7

Γενικά

Προκειμένου να κατανοήσουμε τη φύση του προβλήματος ας δούμε συνοπτικά τα σημεία των αλγόριθμων συμπίεσης Mpeg4 και Mpeg7 που μας ενδιαφέρουν.

Αρχικά πρέπει να γίνει κατανοητό ότι οι επεκτάσεις του Mpeg προτύπου βασίζονται στο προηγούμενο πρότυπο. Δηλαδή το Mpeg7 βασίζεται στο Mpeg4, αυτό στο Mpeg2 και ούτω καθεξής. Έτσι για να μπορέσει κανείς να κατανοήσει το Mpeg7 θα πρέπει να γνωρίζει καλά πρώτα όλα τα προηγούμενα πρότυπα συμπίεσης video και εικόνας στα οποία βασίζεται. Αυτό ξεφεύγει από τα πλαίσια αυτής της εργασίας, για αυτό το λόγο θα περιγραφούν τα σημεία τα οποία μελετήσαμε και μας ενδιαφέρουν.

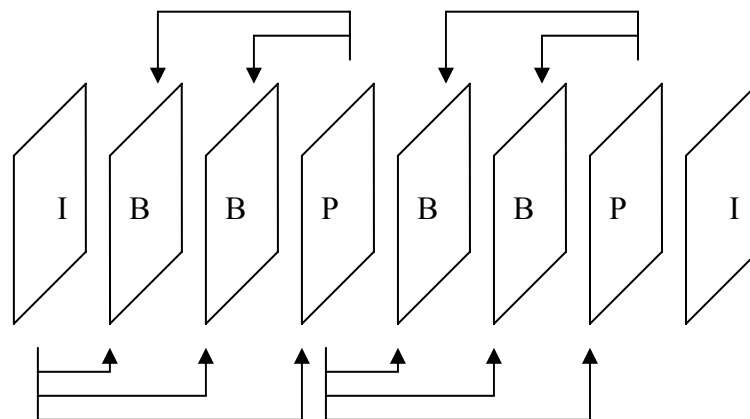
Διανύσματα κίνησης (motion vectors)

Το πρότυπο Mpeg1 καθώς και τα Mpeg2, Mpeg4 και 7 χρησιμοποιούν τριών ειδών καρέ. Τα I-καρέ, τα B-καρέ και τα P-καρέ. Το πρώτο είναι το καρέ αναφοράς και αποτελεί το πρώτο καρέ μιας σκηνής σε ένα video, το οποίο δεν εξαρτάται από τα υπόλοιπα καρέ του video. Το Mpeg1 προσφέρει χαμηλή συμπίεση και χρησιμοποιείται προκειμένου να μπορεί να γίνει γρήγορη τυχαία πρόσβαση. Για την συμπίεσή του χρησιμοποιείται ο προκαλών απώλεια ενέργειας JPEG αλγόριθμος συμπίεσης (lossy jpeg). Μέχρι το επόμενο καρέ αναφοράς, τα υπόλοιπα καρέ είναι τύπου B ή P. Τα P-καρέ χρησιμοποιούν πληροφορία από το προηγούμενο I-καρέ ή και τα προηγούμενα P-καρέ και προσφέρουν μέτρια συμπίεση. Τα B-καρέ είναι αυτά που προσφέρουν τη μεγαλύτερη συμπίεση και χρησιμοποιούν πληροφορία από προηγούμενα ή και μελλοντικά I-καρέ και B-καρέ (Σχήμα 2).

Είναι κατανοητό λοιπόν ότι προκειμένου να επιτευχθεί συμπίεση, χρησιμοποιούμε χρονική πληροφορία, δηλαδή την εκ των προτέρων γνώση

ότι τα περισσότερα καρέ σε ένα video δεν είναι ανεξάρτητα μεταξύ τους (I-καρέ) αλλά είναι σχεδόν ίδια με τα προηγούμενα – επόμενα και μπορούμε να εξοικονομήσουμε χώρο αν δούμε και αξιοποιήσουμε αυτήν την σχέση. Δηλαδή τα I-καρέ εμφανίζονται όταν αλλάζει σκηνή στο video ή όταν οι απαιτήσεις επιβάλλουν I-καρέ για γρήγορη αναζήτηση. Τα υπόλοιπα καρέ του video είναι τύπου B και P και όσον αφορά τα αντικείμενα που αναζητάμε, τα αντιμετωπίζουμε με τον ίδιο τρόπο.

Η εύρεση των διανυσμάτων κίνησης ξεκινάει από το πρότυπο Mpeg1. Τα διανύσματα κίνησης περιγράφουν την κίνηση κάθε σημείου (pixel) ή περιοχής (block) ανάμεσα στα καρέ του video. Είναι η προσπάθεια εισαγωγής της χρονικής πληροφορίας στον αλγόριθμο συμπίεσης Mpeg. Το σύνολο των διανυσμάτων αυτών μεταξύ δυο καρέ του video καλείται οπτική ροή (optical flow).



Σχήμα 2. Πιθανή αλληλουχία καρέ μεταξύ δυο I καρέ

Αντικείμενα

Στα πρότυπα Mpeg4 και Mpeg7 εμφανίζεται η έννοια του αντικειμένου. Δεν γίνεται καμία αναφορά όμως στην περιγραφή τους, ούτε στις συνθήκες που ικανοποιεί μια περιοχή για να ονομαστεί αντικείμενο. Έτσι σύμφωνα με τη θεώρηση ότι αναζητάμε καλύτερη και πιο αποδοτική συμπίεση, ορίζουμε ως αντικείμενο μια κλειστή περιοχή της εικόνας που κινείται διαφορετικά από τη στατική εικόνα (background) με ομαλή κίνηση. Αυτή η θεώρηση είναι βολική για την κωδικοποίηση.

Βασικό πρόβλημα στην εύρεση των αντικειμένων είναι ότι πολλές φορές τα αντικείμενα δεν περιγράφονται από ευδιάκριτα όρια καθώς και κινούμενα αντικείμενα μπορεί να βρίσκονται σε επικάλυψη, να περιστρέφονται και να πλησιάζουν ή να απομακρύνονται από την κάμερα λήψης του video. Η κάμερα από τη δική της μεριά μετατρέπει την τρισδιάστατη εικόνα του φυσικού χώρου σε δυσδιάστατη, με αποτέλεσμα να έχουμε σημαντική μείωση της πληροφορίας και να είναι μερικές φορές πολύ δύσκολη η επεξεργασία. Πρέπει επίσης να γίνει κατανοητό ότι υπάρχει μεγάλη διαφορά ανάμεσα στο πως ο άνθρωπος μπορεί να ξεχωρίσει τα αντικείμενα σε μια εικόνα και τα κινούμενα αντικείμενα σε ένα video. Σε αυτή τη διαδικασία συμβάλει η ανθρώπινη αντίληψη, πράγμα που δεν είναι διαθέσιμο στην ψηφιακή τεχνολογία. Για αυτό το λόγο χρησιμοποιούμε μαθηματικά μοντέλα που είναι εύκολα υλοποιήσιμα, έχουν κατανοητό τρόπο λειτουργίας και σε γενικές γραμμές σταθερό και κατανοητό αποτέλεσμα.

Στο Σχήμα 3 παρατηρούμε το 150 καρέ γνωστού video (α) καθώς και τα δομικά του στοιχεία background (β) και αντικείμενα (γ) (δ). Η βασική ιδέα είναι ότι το background παραμένει σταθερό, άρα το γνωρίζουμε και υπάρχει εκ των προτέρων. Μονάχα τα αντικείμενα αλλάζουν που όμως είναι πολύ μικρότερη πληροφορία από ολόκληρο το καρέ.



Σχήμα 3. (α) Το αρχικό καρέ, (β) Το background της σκηνής και (γ)(δ) τα αντικείμενα που περιέχει το καρέ

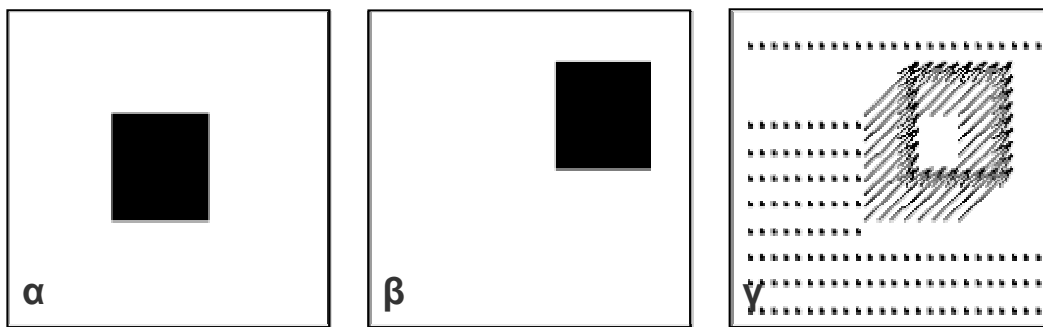
Η αναδόμηση της σκηνής είναι μια διαδικασία που χρησιμοποιεί το background και τα αντικείμενα και δημιουργείται σύμφωνα με τις υποστηριζόμενες διαδικασίες. Αν το background είναι στάσιμο, όπως στη σκηνή του παραδείγματος, δεν είναι απαραίτητη η μετάδοσή του σε κάθε καρέ. Οι διαδικασίες που υποστηρίζονται περιλαμβάνουν την εναπόθεση των αντικειμένων πάνω στο background (superposition), την μετατόπιση, μεγέθυνση, περιστροφή των αντικειμένων κ.α. Επίσης υπάρχει η δυνατότητα χρήσης οποιασδήποτε μεθόδου κρίνεται απαραίτητο, αρκεί να ενημερωθεί κατάλληλο ο αποκωδικοποιητής που σκοπό έχει να συνθέσει το video από τα επιμέρους στοιχεία του (background, αντικείμενα, μεθόδους) και να το αναπαραστήσει.

Επίσης, είναι σημαντικό να αναφερθεί, ότι ο αλγόριθμος Mpeg7 προσεγγίζει τα αντικείμενα πιο κοντά στην ανθρώπινη φύση και κατανόηση. Δηλαδή όπως φαίνεται και στο Σχήμα 3, στην κωδικοποίηση Mpeg7 τα δυο αντικείμενα που υπάρχουν στη σκηνή, είναι δυο άνθρωποι. Ο αλγόριθμος όχι μονάχα τα ξεχωρίζει από το background, αλλά μπορεί να περιέχει πληροφορίες για τους ανθρώπους αυτούς, όπως το όνομά τους καθώς και ότι άλλο κρίνεται απαραίτητο. Αυτό μας δείχνει την αναγκαιότητα του αλγόριθμου ανίχνευσης αντικειμένων, να περιέχει όσο το δυνατό καλύτερη εκτίμηση των αντικειμένων, διατηρώντας τόσο το σχήμα τους όσο και την ενότητά τους.

Οπτική ροή

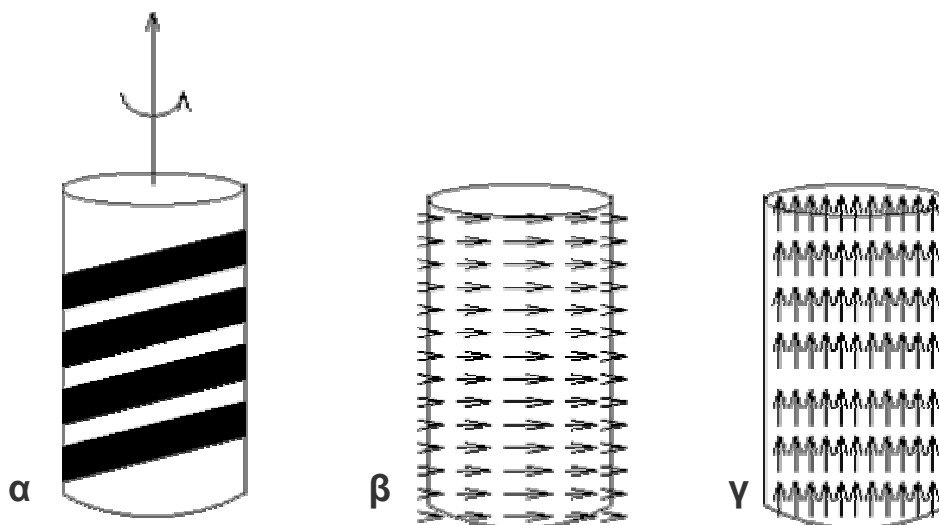
Γενικά

Ως οπτική ροή (optical flow) ορίζουμε το σύνολο των διανυσμάτων της κίνησης (u, v) μεταξύ δυο διαδοχικών εικόνων (καρέ) ενός video όπως φαίνεται και στο Σχήμα 4.



Σχήμα 4. Η οπτική ροή (γ) μεταξύ των καρέ (α) και (β)

Τα διανύσματα κίνησης δεν αντιπροσωπεύουν πάντα την οπτική ροή καθώς για την εύρεσή τους υπάρχει μονάχα η προβολή του χώρου σε δυο διαστάσεις. Αυτό φαίνεται και στο Σχήμα 5, όπου η προβολή του κυλίνδρου σε δυο διαστάσεις προκαλεί την διαφορετική εκτίμηση της κίνησης.



Σχήμα 5. (α) Περιστρεφόμενος κύλινδρος, (β) η οπτική ροή και (γ) τα διανύσματα κίνησης

Αν μοντελοποιήσουμε το βίντεο ως συνάρτηση $I(\mathbf{x}, t)$ με $\mathbf{x} = (x, y)$, οπτική ροή είναι το σύνολο των διανυσμάτων $\mathbf{v}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ τέτοιο ώστε να ισχύει ότι:

$$I(\mathbf{x}, t)\mathbf{v} + I_t(\mathbf{x}, t) = 0$$

Βέβαια λόγω θορύβου, παραμορφώσεων κλπ αυτό δεν είναι πάντα εφικτό να ισχύει, οπότε \mathbf{v} είναι το σύνολο των διανυσμάτων τέτοιο ώστε να ελαχιστοποιεί μια δοσμένη συνάρτηση λάθους ανάλογο των απαιτήσεών μας. Μια τυπική συνάρτηση κόστους είναι η $e(\mathbf{x}, y, t) = I(\mathbf{x}, y, t) - I(\mathbf{x}-u, y-v, t-1)$ που μας δείχνει το λάθος στην εκτίμηση των διανυσμάτων.

Μέθοδοι υπολογισμού οπτικής ροής

Για τον υπολογισμό της οπτικής ροής έχουν χρησιμοποιηθεί διάφορες μέθοδοι [11] [12] [13] [14] [17] [19]. Οι κυριότερες από αυτές είναι οι διαφορικές μέθοδοι (differential), οι μέθοδοι βασισμένες στη συχνότητα (frequency-based) και οι μέθοδοι ταιριάσματος (block matching). Έχουν γίνει αρκετές προσπάθειες συλλογής και αξιολόγησης των μεθόδων υπολογισμού της οπτικής ροής [8] [16] [18].

Οι διαφορικές μέθοδοι υπολογίζουν την οπτική ροή από χωρικές και χρονικές παραγώγους του video. Οι μέθοδοι αυτοί χωρίζονται σε δύο κύριες κατηγορίες ανάλογα με την τάξη των παραγώγων που χρησιμοποιούν.

Οι μέθοδοι πρώτης τάξης συνήθως χρησιμοποιούν πρωτοτάξια κατωδιαβατά (first-order low-pass) ή ζωνοδιαβατά (first-order band-pass) φίλτρα. Ορίζουμε ως $\mathbf{v}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))$ τη συνάρτηση των διανυσμάτων της κίνησης στο χώρο $\mathbf{x} = (x, y)$ και στο χρόνο t . Επίσης $u(\mathbf{x}, t)$ και $v(\mathbf{x}, t)$ είναι οι συνιστώσες της ταχύτητας στο χώρο και $I(\mathbf{x}, t)$ το video. Η συνάρτηση βάση της οποίας γίνεται ο υπολογισμός των διανυσμάτων κίνησης είναι η:

$$\nabla I(\mathbf{x}, t)\mathbf{v} + I_t(\mathbf{x}, t) = 0$$

Οι μέθοδοι δεύτερης τάξης χρησιμοποιούν φίλτρα δεύτερης παραγώγου (Hessian) της εικόνας. Η συνάρτηση βάση της οποίας γίνεται ο υπολογισμός των διανυσμάτων κίνησης είναι η:

$$\begin{bmatrix} I_{xx}(\mathbf{x}, t) & I_{yx}(\mathbf{x}, t) \\ I_{xy}(\mathbf{x}, t) & I_{yy}(\mathbf{x}, t) \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} I_{tx}(\mathbf{x}, t) \\ I_{ty}(\mathbf{x}, t) \end{pmatrix} = 0$$

Οι μέθοδοι βασισμένες στη συχνότητα χρησιμοποιούν τη σχέση $I(\mathbf{x}, t)\mathbf{v} + I_t(\mathbf{x}, t) = 0$ εκφρασμένη στο πεδίο της συχνότητας, βάση της οποίας γίνεται ο υπολογισμός των διανυσμάτων κίνησης:

$$\hat{I}(\mathbf{k}, \omega) = \hat{I}_0(\mathbf{k})\delta(\omega + \mathbf{v}^T \mathbf{k})$$

όπου $\hat{I}(\mathbf{k})$ είναι ο μετασχηματισμός Fourier της εικόνας $I(\mathbf{x}, 0)$, $\delta(\mathbf{k})$ είναι η κρουστική συνάρτηση δέλτα του Dirac, ω είναι η συχνότητα και $\mathbf{k} = (k_x, k_y)$ είναι η χωρική συχνότητα.

Τέλος η πιο γνωστή κατηγορία μεθόδων είναι αυτή του ταιριάσματος (block matching). Στην κατηγορία αυτή ανήκουν οι μέθοδοι βασισμένες στην περιοχή (region – based) και οι μέθοδοι χαρακτηριστικών (feature – based). Οι μέθοδοι βασίζονται στην αναζήτηση των διανυσμάτων κίνησης μονάχα σε μια γειτονιά του σημείου και η εύρεσή τους βασίζεται στην σχέση:

$$I(\mathbf{x}, t) - I(\mathbf{x} + \mathbf{u}, t) = 0$$

Τα βασικά πλεονεκτήματα των μεθόδων αυτών είναι η μεγάλη ανοχή τους στο θόρυβο, στην αλλοίωση της εικόνας και στην απλότητα του υπολογισμού της οπτικής ροής με αποτέλεσμα την χρήση μικρού αριθμού καρέ. Βασικό μειονέκτημα της μεθόδου είναι η ταχύτητα.

Υπολογισμός της οπτικής ροής σύμφωνα με Horn – Schunck

Σύμφωνα με τον Horn και Schunck [7], η οπτική ροή ορίζεται σε σχέση με τη χωρική και τη χρονική αλλαγή της έντασης της φωτεινότητας ανάμεσα στα καρέ του video. Οπτική ροή είναι η διανομή της φαινομενικής ταχύτητας της κίνησης του χρωματικού μοτίβου σε μια εικόνα.

Η ανοιχτότητα του προβλήματος είναι αναπόφευκτη εφόσον δεν είναι ένα πρόβλημα πλήρως ορισμένο [8] [9]. Για αυτό το λόγο γίνονται κάποιες

βολικές υποθέσεις προκειμένου να γίνει μια εκτίμηση και να μπορέσουν να υπολογιστούν τα διανύσματα κίνησης. Έτσι η ταχύτητα ανάμεσα σε δυο καρέ του video, μπορεί να υπολογιστεί από το ολοκλήρωμα:

$$F_t = \int_D (\nabla I \cdot \mathbf{v} + I_t)^2 + \lambda^2 (\|\nabla u\|_2^2 + \|\nabla v\|_2^2) d\mathbf{x}$$

που ορίζεται στα όρια της εικόνας του καρέ. Ιδανικά το ολοκλήρωμα αυτό είναι μηδέν. Με πραγματικά δεδομένα όμως, ο πρώτος όρος $(\nabla I \cdot \mathbf{v} + I_t)^2$ είναι διάφορος του μηδενός λόγω θορύβου, παραμορφώσεων και άλλων φαινομένων που έχουν να κάνουν με τη φύση του video και ο δεύτερος όρος $\|\nabla u\|_2^2 + \|\nabla v\|_2^2$ είναι επίσης διάφορος του μηδενός, για εκτίμηση που δε συμφωνεί με τα γειτονικά διανύσματα κίνησης αφού θεωρούμε ότι ιδεατά τα διανύσματα κίνησης είναι ομαλά (smoothness constraint). Έτσι το ολοκλήρωμα αυτό είναι μηδέν για επιλογή διανυσμάτων κίνησης τέτοιων ώστε να εκφράζουν τόσο την κίνηση με κριτήριο την ένταση (πρώτος όρος) όσο και την κίνηση με κριτήριο την ομοιομορφία των διανυσμάτων στα πλαίσια της περιοχής γύρω του (δεύτερος όρος). Ο παράγοντας λ υπάρχει για να μπορεί να ρυθμίζει την ισχύ των δυο δεσμεύσεων που αποτελούν τη βάση του υπολογισμού των διανυσμάτων κίνησης.

Αναπτύσσοντας το ολοκλήρωμα λοιπόν έχουμε:

$$F_t = \int_D \lambda (I_x u + I_y v + I_t)^2 + (u_x^2 + u_y^2 + v_x^2 + v_y^2) d\mathbf{x}$$

Οπότε σε διακριτή μορφή το ολοκλήρωμα θα πάρει τη μορφή:

$$F_t = \sum_{i,j} \lambda (I_x u + I_y v + I_t)^2 + \frac{1}{4} ((u_{i+1,j} - u_{ij})^2 + (u_{i,j+1} - u_{ij})^2 + (v_{i+1,j} - v_{ij})^2 + (v_{i,j+1} - v_{ij})^2)$$

Έτσι προκειμένου να ελαχιστοποιήσουμε το ολοκλήρωμα αρκεί:

$$\nabla F_t = 0 \quad \text{ή αλλιώς} \quad \begin{cases} \frac{\partial F_t}{\partial u} = 0 \\ \frac{\partial F_t}{\partial v} = 0 \end{cases}$$

Η μερική παράγωγος $\frac{\partial F_t}{\partial u}$ υπολογίζεται παρακάτω:

$$\frac{\partial F_t}{\partial u} = 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_x - \frac{1}{2}[(u_{i+1,j} - u_{ij}) + (u_{i,j+1} - u_{ij})] + \frac{1}{2}[(u_{ij} - u_{i-1,j}) + (u_{ij} - u_{i,j-1})] \Leftrightarrow$$

$$\frac{\partial F_t}{\partial u} = 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_x + 2u_{ij} - 2\left(\frac{1}{4}(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1})\right) \Leftrightarrow$$

$$\frac{\partial F_t}{\partial u} = 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_x + 2u_{ij} - 2\bar{u}_{ij}$$

Ομοίως υπολογίζουμε και την μερική παράγωγο $\frac{\partial F_t}{\partial v}$ οπότε έχουμε:

$$\frac{\partial F_t}{\partial v} = 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_y + 2v_{ij} - 2\bar{v}_{ij}$$

Οπότε το σύστημα που έχουμε να λύσουμε είναι:

$$\begin{cases} 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_x + 2u_{ij} - 2\bar{u}_{ij} = 0 \\ 2\lambda(I_x u_{ij} + I_y v_{ij} + I_t)I_y + 2v_{ij} - 2\bar{v}_{ij} = 0 \end{cases} \Leftrightarrow \begin{cases} (1 + \lambda I_x^2)u_{ij} + \lambda I_x I_y v_{ij} = \bar{u} - \lambda I_x I_t \\ \lambda I_x I_y u_{ij} + (1 + \lambda I_y^2)v_{ij} = \bar{v} - \lambda I_y I_t \end{cases}$$

Σε μορφή πινάκων μπορεί να γραφτεί:

$$\begin{bmatrix} 1 + \lambda I_x^2 & \lambda I_x I_y \\ \lambda I_x I_y & 1 + \lambda I_y^2 \end{bmatrix} \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} \bar{u} - \lambda I_x I_t \\ \bar{v} - \lambda I_y I_t \end{bmatrix}$$

Για την επίλυση του παραπάνω συστήματος εφαρμόσαμε την επαναληπτική μέθοδο Jacobi (Jacobi iterations), όπου η λύση του συστήματος δίνεται από τον τύπο:

$$x^{(m+1)} = D^{-1}(L + U)x^{(m)} + D^{-1}b \mid m = 0, 1, 2, \dots$$

Ο πίνακας των συντελεστών των u, v ονομάζεται A και ο πίνακας των σταθερών όρων b . Από αυτούς κατασκευάζουμε τους πίνακες D, L και U ώστε να ισχύει $A = D - L - U$ όπου D να είναι διαγώνιος, L αυστηρά κάτω τριγωνικός και U αυστηρά πάνω τριγωνικός πίνακας. Έτσι οι πίνακες του σχήματος είναι οι παρακάτω:

$$D = \begin{bmatrix} 1 + \lambda I_x^2 & 0 \\ 0 & 1 + \lambda I_y^2 \end{bmatrix} \quad U = \begin{bmatrix} 0 & -\lambda I_x I_y \\ 0 & 0 \end{bmatrix} \quad L = \begin{bmatrix} 0 & 0 \\ -\lambda I_x I_y & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} \bar{u} - \lambda I_x I_t \\ \bar{v} - \lambda I_y I_t \end{bmatrix} \text{ και } D^{-1} = \begin{bmatrix} \frac{1}{1 + \lambda I_x^2} & 0 \\ 0 & \frac{1}{1 + \lambda I_y^2} \end{bmatrix}$$

Έτσι έχουμε:

$$\begin{aligned}
 x^{(m+1)} &= D^{-1}(L+U)x^{(m)} + D^{-1}b \\
 \begin{bmatrix} u^{(m+1)} \\ v^{(m+1)} \end{bmatrix} &= \begin{bmatrix} 1/(1+\lambda I_x^2) & 0 \\ 0 & 1/(1+\lambda I_y^2) \end{bmatrix} \begin{bmatrix} 0 & -\lambda I_x I_y \\ -\lambda I_x I_y & 0 \end{bmatrix} \begin{bmatrix} u^{(m)} \\ v^{(m)} \end{bmatrix} + \\
 &\quad \begin{bmatrix} 1/(1+\lambda I_x^2) & 0 \\ 0 & 1/(1+\lambda I_y^2) \end{bmatrix} \begin{bmatrix} \bar{u} - \lambda I_x I_t \\ \bar{v} - \lambda I_y I_t \end{bmatrix} \\
 \begin{bmatrix} u^{(m+1)} \\ v^{(m+1)} \end{bmatrix} &= \begin{bmatrix} 0 & \frac{-\lambda I_x I_y}{1+\lambda I_x^2} \\ \frac{-\lambda I_x I_y}{1+\lambda I_y^2} & 0 \end{bmatrix} \begin{bmatrix} u^{(m)} \\ v^{(m)} \end{bmatrix} + \begin{bmatrix} \frac{\bar{u} - \lambda I_x I_t}{1+\lambda I_x^2} \\ \frac{\bar{v} - \lambda I_y I_t}{1+\lambda I_y^2} \end{bmatrix} \\
 \begin{bmatrix} u^{(m+1)} \\ v^{(m+1)} \end{bmatrix} &= \begin{bmatrix} \frac{-\lambda I_x I_y}{1+\lambda I_x^2} v^{(m)} \\ \frac{-\lambda I_x I_y}{1+\lambda I_y^2} u^{(m)} \end{bmatrix} + \begin{bmatrix} \frac{\bar{u} - \lambda I_x I_t}{1+\lambda I_x^2} \\ \frac{\bar{v} - \lambda I_y I_t}{1+\lambda I_y^2} \end{bmatrix} \\
 \begin{bmatrix} u^{(m+1)} \\ v^{(m+1)} \end{bmatrix} &= \begin{bmatrix} \frac{\bar{u} - \lambda I_x I_t - \lambda I_x I_y v^{(m)}}{1+\lambda I_x^2} \\ \frac{\bar{v} - \lambda I_y I_t - \lambda I_x I_y u^{(m)}}{1+\lambda I_y^2} \end{bmatrix}
 \end{aligned}$$

Για $x^{(0)}$ επιλέγονται αυθαίρετες τιμές, συνήθως μηδέν. Βέβαια όταν πρόκειται για παραπάνω των δυο καρέ, στην ακολουθία του video που ζητάμε να βρούμε τα διανύσματα κίνησης, τότε μια καλή αρχικοποίηση είναι η λύση του προηγούμενου ζευγαριού. Αυτό είναι προφανές δεδομένου ότι περιμένουμε μικρές αλλαγές στην κίνηση μεταξύ δυο διαδοχικών καρέ.

Για τον υπολογισμό του παραπάνω σχήματος έγιναν οι ακόλουθες παραδοχές:

- Ότι η κίνηση ανάμεσα στα καρέ του video είναι αρκετά μικρή με αποτέλεσμα να μπορούμε να θεωρήσουμε ότι:

$$\frac{dI}{dt} \approx 0$$

- Ότι υπάρχει η παράγωγος της εικόνας σε κάθε σημείο ως προς x και ως προς y . Δηλαδή:

$$\frac{\partial x}{\partial t} \neq 0 \text{ και } \frac{\partial y}{\partial t} \neq 0$$

Αυτό σημαίνει ότι δεν υπολογίζεται οπτική ροή για σημεία για ομοιόμορφες περιοχές με μηδενικές παραγώγους στο χώρο. Για τα σημεία αυτά προσεγγίζουμε την οπτική ροή, υπολογίζοντάς την έμμεσα, ως το μέσο όρο της οπτικής ροής στα γειτονικά σημεία.

Υπάρχουσες μέθοδοι

Η ανίχνευση των κινούμενων αντικειμένων στις σκηνές του video, είναι μια διαδικασία δύσκολη και με αποτέλεσμα εξαρτώμενο του video – εισόδου. Μέχρι τη στιγμή συγγραφής της εργασίας αυτής δεν υπάρχει αρκετή ερευνητική δουλειά στον τομέα αυτό, καθώς η ολοκληρωμένη διαδικασία, που θα είναι ανεξάρτητη του video – εισόδου, είναι πολύπλοκη και χρειάζεται πολλά βήματα προκειμένου να καταλήξει στο επιθυμητό αποτέλεσμα.

Έτσι οι Wu και Chen [20], προτείνουν ένα σχήμα υλοποίησης, βάση του οποίου η εξαγωγή των κινούμενων αντικειμένων χωρίζεται σε δυο στάδια. Στο πρώτο στάδιο γίνεται η εξαγωγή των κινούμενων αντικειμένων, βάση πρώτης παραγώγου στο χρόνο και στατικού background. Για το σκοπό αυτό, χρησιμοποιείται το χρωματικό μοντέλο HSV (Hue, Saturation, Value), όπου οι τρεις χρωματικές συνιστώσες μπορούν να χρησιμοποιηθούν ανεξάρτητα. Έτσι για τον ορισμό του background, χρησιμοποιείται τόσο η μέση τιμή κάθε σημείου της εικόνας όσο και η συμμεταβλητότητα (covariance) , βάση της οποίας background είναι κάθε σημείο που διαφέρει λιγότερο από το διπλάσιο της συμμεταβλητότητας από την αντίστοιχη μέση τιμή. Καθώς η μέθοδος αυτή φαίνεται χρονοβόρα, επίσης προτείνει την απόλυτη μεσαία απόκλιση των χρωματικών συνιστωσών σε κάθε σημείο. Από τη διαδικασία αυτή, εξάγει τα σημεία της εικόνας που ανήκουν στα κινούμενα αντικείμενα. Προκειμένου να ομαδοποιήσει τα σημεία αυτά, για να αφαιρέσει μεμονωμένα σημεία και να γεμίσει τρύπες που έχουν σχηματιστεί, χρησιμοποιεί αλγόριθμο συνδεδεμένων συστατικών (connected components), βάση του οποίου η ομαδοποίηση των σημείων, γίνεται βάση των γειτονικών σε αυτά σημείων (8-neighbor labeling).

Ο συνδυασμός των παραπάνω πληροφοριών είναι το βασικό σημείο της δουλειάς αυτής. Δηλαδή θεωρεί ότι ακριβής ανίχνευση των ακμών του αντικειμένου γίνεται από την παράγωγο στο χρόνο, αφού χρησιμοποιηθεί αλγόριθμος λέπτυνση των εντοπισμένων ακμών. Από τα κινούμενα αντικείμενα που έχουν απομονωθεί στην αρχή της διαδικασίας χρησιμοποιεί

μια 3×3 laplacian μάσκα, για να εξάγει τις ακμές τους και τις συνδυάζει μαζί με τις ακμές της πρώτης παραγώγου στο χρόνο για να ολοκληρώσει το ακριβές περίγραμμα. Για τη διαδικασία αυτή χρησιμοποιεί αλγόριθμο αλυσίδας (chain code).

Η παραπάνω διαδικασία φαίνεται να έχει αρκετά καλά αποτελέσματα και κυρίως είναι αρκετά γρήγορη και μπορεί να χρησιμοποιηθεί για εφαρμογές πραγματικού χρόνου. Η αδυναμία του παραπάνω αλγορίθμου είναι ότι δεν μπορεί να γίνει σαφής διαχωρισμός των ανεξάρτητων κινούμενων αντικειμένων που περιέχονται στην εικόνα.

Εύρεση ακμών κίνησης

Πρόλογος

Μια πρώτη προσέγγιση των κινούμενων αντικειμένων είναι η εύρεση των ακμών τους. Σκοπός μας είναι ο διαχωρισμός του χώρου πάνω στην εικόνα σε στατική πληροφορία (background) και στα αντικείμενα που κινούνται πάνω σε αυτό (objects). Η κίνηση στο χρόνο των αντικειμένων μπορεί να ανιχνευτεί με δυο τρόπους. Με την ανωμαλία των διανυσμάτων κίνησης στα σημεία που αλλάζει η κίνηση των αντικειμένων και με την παράγωγο του video στο χρόνο.

Μέθοδος διανυσμάτων κίνησης

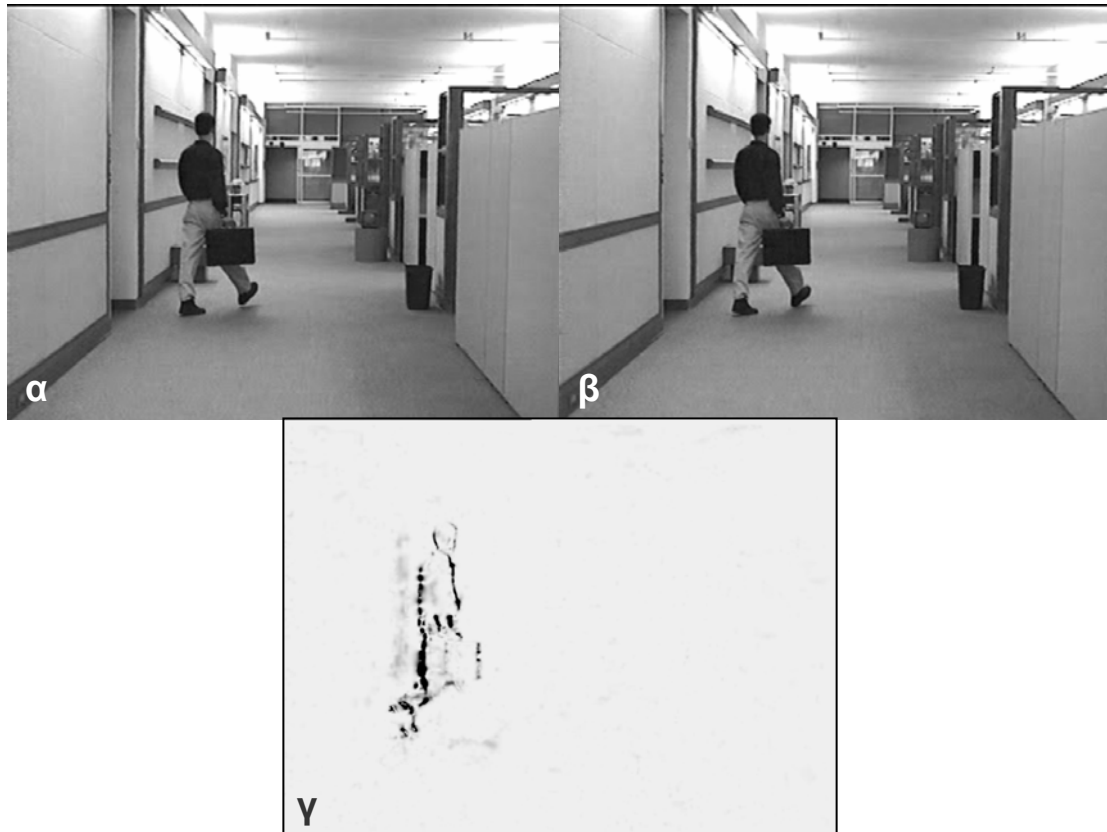
Η οπτική ροή μπορεί να δώσει σημαντικές πληροφορίες για τα αντικείμενα που αποτελούν μια εικόνα. Γνωρίζοντας την κίνηση κάθε περιοχής και παρατηρώντας τις μεταβολές της, μπορούμε να κάνουμε ένα βασικό διαχωρισμό της εικόνας σε κινούμενες και ακίνητες περιοχές. Για την εύρεση των διανυσμάτων κίνησης είναι απαραίτητη η γνώση του επόμενου στιγμιότυπου του video (επόμενο καρέ) $I(x,y,t+1)$.

Η μέθοδος που προτείνουμε βασίζεται στα διανύσματα κίνησης. Η σχέση που μας δίνει το εσωτερικό γινόμενο δυο διανυσμάτων $\vec{A} = (x_1, y_1)$ και $\vec{B} = (x_2, y_2)$ είναι $\vec{A} \cdot \vec{B} = x_1 x_2 + y_1 y_2$. Τα διανύσματα κίνησης εκφράζουν την κίνηση κάθε σημείου σε σχέση με το επόμενο καρέ του video. Προκειμένου να ανιχνεύσουμε τις ακμές προσπαθούμε να ανιχνεύσουμε αλλαγή στο πεδίο της κίνησης (u, v) . Η αλλαγή στο πεδίο της κίνησης εκφράζεται ως το εσωτερικό γινόμενο του κάθε διανύσματος με τα γύρω του. Έτσι για κάθε διάνυσμα εφαρμόζουμε τον τύπο:

$$I_v(\mathbf{x}) = \sum_{i=-1}^1 \sum_{j=-1}^1 \mathbf{u}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x} + (i, j))$$

Αποτέλεσμα αυτής της διαδικασίας είναι το περίγραμμα του κινούμενου αντικειμένου, καθώς μεγάλες τιμές θα εμφανίζονται εκεί που υπάρχει

ασυνέχεια στη ροή της διανυσμάτων κίνησης. Το αποτέλεσμα της διαδικασίας αυτής φαίνεται στο Σχήμα 6.



Σχήμα 6. (α)(β) Παράδειγμα από το γνωστό video Hall Monitor σε δυο διαδοχικά καρέ (γ) Ο μέσος όρος των διανυσμάτων κίνησης

Μέθοδος πρώτης παραγώγου

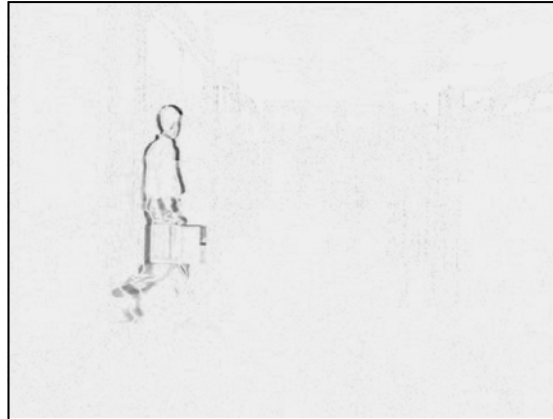
Σημαντική και βασική μέθοδος ανίχνευσης της κίνησης είναι αυτή της πρώτης παραγώγου στο χρόνο. Η φύση του προβλήματος επιβάλλει την χρήση της μεθόδου αυτής καθώς είναι πιο αποδοτική στις ακμές μιας εικόνας, δηλαδή και στις ακμές των αντικειμένων, τις οποίες αναζητάμε.

Η σχέση που δίνει την πρώτη παράγωγο στο χρόνο είναι η εξής:

$$I_t(\mathbf{x}, t) = \left| \frac{\partial I(\mathbf{x}, t)}{\partial t} \right| = |I(\mathbf{x}, t) - I(\mathbf{x}, t+1)|$$

Η απόλυτη τιμή στη σχέση εκφράζει την επιθυμία μας στην εύρεση τόσο των θετικών ακμών όσο και των αρνητικών ακμών. Εκφράζει τις

κινούμενες ακμές καθώς οτιδήποτε στατικό από τη σχέση θα δώσει μηδέν, ενώ σε οτιδήποτε κινείται θα δημιουργήσει το "περίγραμμα" του. Η παράγωγος στο χρόνο δεν είναι ακριβώς το περίγραμμα του κινούμενου αντικειμένου γιατί στον άξονα κίνησης το "περίγραμμα" θα έχει πάχος το μήκος του διανύσματος της κίνησης του ενώ κάθετα σε αυτόν δεν θα υπάρχει καθόλου "περίγραμμα". Τα αποτελέσματα της διαδικασίας αυτής φαίνονται στο Σχήμα 7.



Σχήμα 7. Η πρώτη παράγωγος στο χρόνο

Συνδυασμός διανυσμάτων κίνησης και πρώτης παραγώγου

Ο συνδυασμός των ακμών από τα διανύσματα κίνησης και από την πρώτη παράγωγο στο χρόνο μπορεί να δώσει μια πιο ακριβή εκτίμηση των ακμών. Για το σκοπό αυτό χρησιμοποιούμε το φίλτρο «μέγιστου» που αντιστοιχεί στη λογική πράξη OR. Η έξοδός του σε κάθε σημείο της εικόνας είναι το μέγιστο από τις αντίστοιχες τιμές της εικόνας σύμφωνα με τα αποτελέσματα των δυο παραπάνω μεθόδων. Αν $I_v(\mathbf{x}, t)$ η εκτίμηση των ακμών από τα διανύσματα κίνησης και $I_t(\mathbf{x}, t)$ οι ακμές από την πρώτη παράγωγο, τότε οι κινούμενες ακμές βρίσκονται από τον τύπο:

$$I_m(\mathbf{x}, t) = \max \{I_v(\mathbf{x}, t), I_t(\mathbf{x}, t)\}$$

Το αποτέλεσμα στο παραπάνω παράδειγμα φαίνεται στο Σχήμα 8.



Σχήμα 8. Κινούμενες ακμές

Εύρεση στατικών ακμών

Οι ακμές είναι βασικά χαρακτηριστικά της εικόνας. Φέρουν χρήσιμη πληροφορία για τα όρια των αντικειμένων. Παρά τη θεμελιώδη σημασία τους στην ψηφιακή επεξεργασία και ανάλυση εικόνας, δεν υπάρχει ακόμα ακριβής και ευρέως αποδεκτός μαθηματικός ορισμός της ακμής. Αυτό το γεγονός εξηγείται από την πολυπλοκότητα του περιεχομένου της εικόνας και από την παρεμβολή μηχανισμών όρασης υψηλού επιπέδου στην ανθρώπινη αντίληψη του ορίου ενός αντικειμένου. Έτσι λοιπόν θεωρούμε ως ακμή το σύνορο μεταξύ δύο ομογενών περιοχών μιας εικόνας που έχουν διαφορετικές εντάσεις φωτεινότητας.

Η εύρεση των στατικών ακμών έγινε χρησιμοποιώντας τον ανιχνευτή ακμών Kirsch 3×3 . Η αναζήτηση των ακμών γίνεται βάση της πρώτης παραγώγου της εικόνας σε τέσσερις κατευθύνσεις. Ο υπολογισμός της πρώτης παραγώγου γίνεται σύμφωνα με τις παρακάτω μάσκες:

$$A(I, \mathbf{x}) = I(\mathbf{x}) * \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

που αντιστοιχεί στην παράγωγο στον x άξονα (Σχήμα 9α).

$$B(I, \mathbf{x}) = I(\mathbf{x}) * \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

που αντιστοιχεί στην παράγωγο στον y άξονα (Σχήμα 9β).

$$C(I, \mathbf{x}) = I(\mathbf{x}) * \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

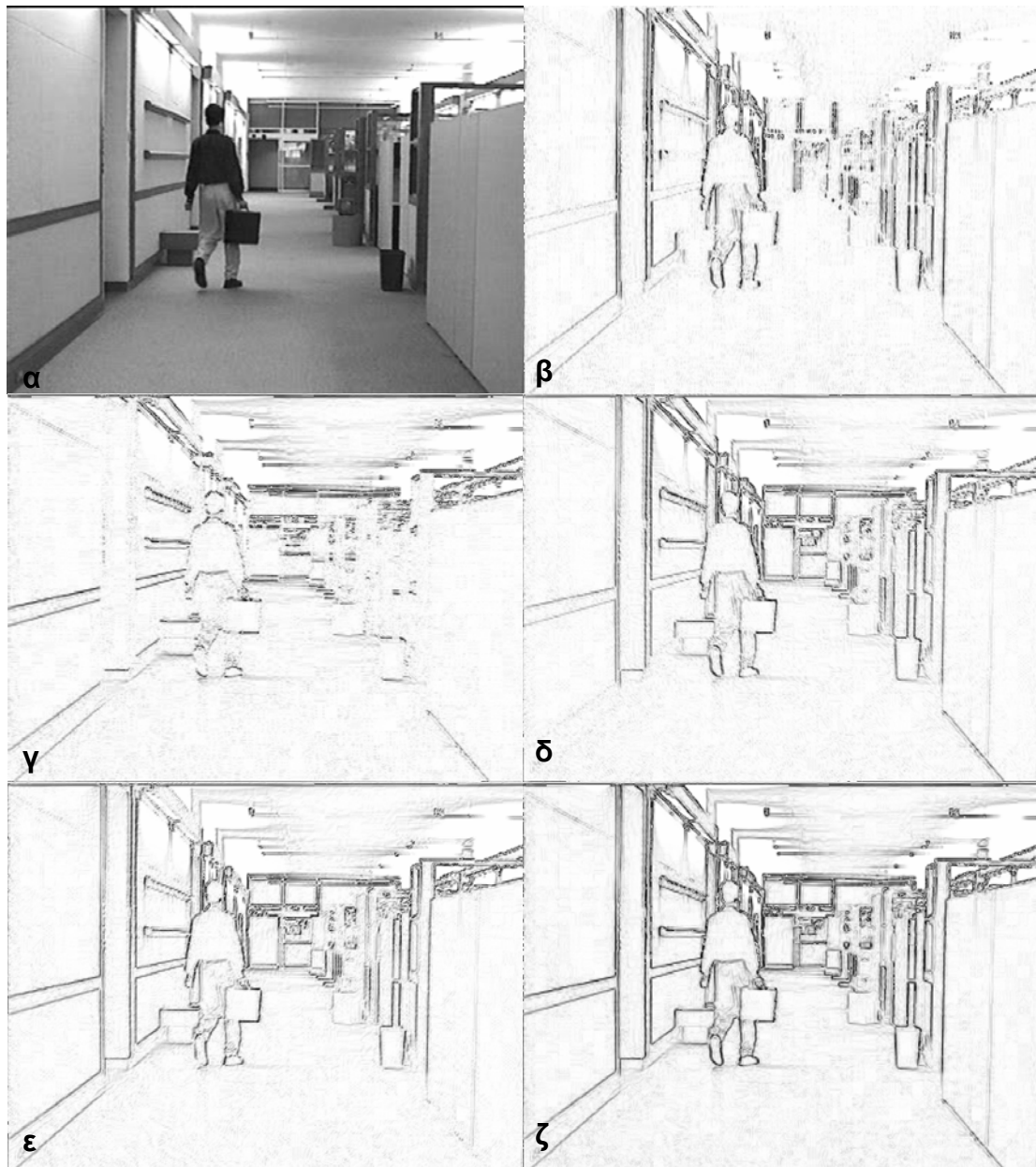
που αντιστοιχεί στην παράγωγο σε άξονα με 45° στροφή του x άξονα (Σχήμα 9γ).

$$D(I, \mathbf{x}) = I(\mathbf{x}) * \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

που αντιστοιχεί σε παράγωγο σε άξονα με 135° στροφή του x άξονα (Σχήμα 9δ). Ο τελεστής (*) στη σχέση συμβολίζει την συνέλιξη της εικόνας $I(x,y)$ με την αντίστοιχη μάσκα. Τελικό αποτέλεσμα του ανιχνευτή σε κάθε σημείο της εικόνας, είναι το μέγιστο της απόλυτης τιμής της κάθε παραγώγου. Δηλαδή:

$$Edges(I, \mathbf{x}) = \max(|A(I, \mathbf{x})|, |B(I, \mathbf{x})|, |C(I, \mathbf{x})|, |D(I, \mathbf{x})|)$$

Η πρώτη παράγωγος στο χρόνο φαίνεται στο Σχήμα 9ε.



Σχήμα 9. Η εφαρμογή των масκών του φίλτρου Kirsch στο καρέ 50 (α) για (β) 0° , (γ) 90° , (δ) 45° , (ε) 135° και (ζ) το συνολικό αποτέλεσμα

Διαχωρισμός τοπικά στατικών ακμών

Υπάρχουν ακμές του αντικειμένου που σταματούν να κινούνται λόγω της σύνθετης κίνησης σε μια πραγματική σκηνή καθώς και λόγω αλλαγής του σχήματος του αντικειμένου κλπ. Αυτό σημαίνει ότι έχουν παράγωγο στο χρόνο ίση με το μηδέν, γεγονός που κάνει αδύνατο τον εντοπισμό τους από τις παραπάνω μεθόδους. Η μέθοδος που εφαρμόζεται σε αυτή την περίπτωση, χρησιμοποιεί την ιδιότητα αυτή των ακμών προκειμένου να τις εντοπίσει.

Αρχικά, τις ακμές της σκηνής χωρίς τα κινούμενα αντικείμενα τις ονομάζουμε ακμές στατικής πληροφορίας και μπορούμε να τις εντοπίσουμε χρησιμοποιώντας το μέσο της σκηνής.

$$\bar{I}(\mathbf{x}, t) = \frac{\sum_{a=N_1}^{a=N_2} I(\mathbf{x}, a)}{N_2 - N_1}$$

όπου N_1, N_2 είναι το καρέ που αρχίζει και που τελειώνει η σκηνή αντίστοιχα. Για το σκοπό αυτό μπορούμε να χρησιμοποιήσουμε όλα τα καρέ του video ($N_1=0$, N_2 =αριθμός καρέ σκηνής) ή να επιλέξουμε βολικά καρέ από την εικόνα. Από το μέσο όρο της σκηνής βρίσκουμε τις ακμές που ανήκουν μονάχα στο background. Για το παράδειγμα «Hall Monitor» χρησιμοποιήσαμε τα πρώτα 12 καρέ του video. Στο Σχήμα 10 παρατηρούμε το αποτέλεσμα της διαδικασίας αυτής.



Σχήμα 10. (α) Στατική πληροφορία (*background*) και (β) ακμές στατικής πληροφορίας

Στη συνέχεια χρησιμοποιούμε την πληροφορία από τις ακμές στο καρέ του video που αναζητάμε και στο χρονικά επόμενο καρέ. Χρησιμοποιήσαμε το φίλτρο «ελαχίστου» που αντιστοιχεί στη λογική πράξη AND. Αποτέλεσμα του φίλτρου είναι η ελάχιστη ένταση μεταξύ των δύο καρέ. Αν $Edges(I, \mathbf{x})$ το αποτέλεσμα της ανίχνευσης ακμών στην εικόνα I , και $I_{t_0}(\mathbf{x}, t)$ οι ακμές που δεν κινούνται στο $[t, t+1]$, τότε μπορούν να βρεθούν από τον τύπο:

$$I_{t_0}(\mathbf{x}, t) = \min\{Edges[I(\mathbf{x}, t+1)], Edges[I_E(\mathbf{x}, t)]\}$$

Τις ονομάζουμε ακίνητες ακμές στιγμιότυπου και το αποτέλεσμά είναι τόσο οι στατικές ακμές της εικόνας για τη δεδομένη χρονική στιγμή του video, δηλαδή οι στατικές ακμές που βρήκαμε με τη μέθοδο του μέσου όρου, όσο και οι ακμές που ανήκουν στο αντικείμενο αλλά δεν παρουσιάζουν κίνηση στη συγκεκριμένη χρονική στιγμή. Το αποτέλεσμα της διαδικασίας φαίνεται στο Σχήμα 11.

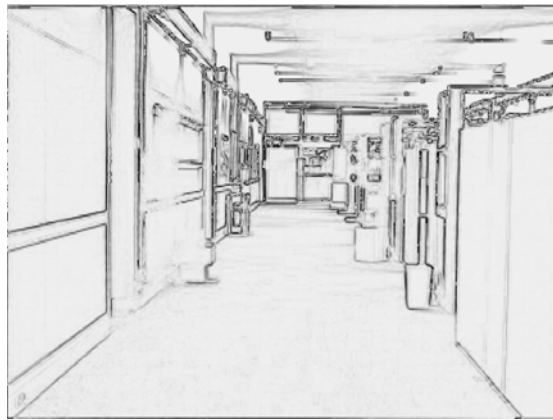


Σχήμα 11. Ακίνητες ακμές στιγμιότυπου

Με τη χρήση φίλτρου «ελαχίστου» στις ακμές στατικής πληροφορίας ($\overline{I(\mathbf{x},t)}$) και στις ακίνητες ακμές στιγμιότυπου ($I_{t_0}(\mathbf{x},t)$), βρίσκουμε τις στατικές ακμές που εμφανίζονται στο συγκεκριμένο στιγμιότυπο. Τις ονομάζουμε στατικές ακμές στιγμιότυπου ($I_{t_N}(\mathbf{x},t)$) και δίνονται από τον τύπο:

$$I_{t_N}(\mathbf{x},t) = \min\{Edges[\overline{I(\mathbf{x},t)}], I_{t_0}(\mathbf{x},t)\}$$

Το αποτέλεσμα της διαδικασίας αυτής φαίνεται στο Σχήμα 12.



Σχήμα 12. Στατικές ακμές στιγμιότυπου

Στη συνέχεια χρησιμοποιώντας τη λογική πράξη NOT μεταξύ των ακίνητων ακμών στιγμιότυπου και των στατικών ακμών στιγμιότυπου, εντοπίζουμε τις ακμές που ανήκουν στα αντικείμενα αλλά δεν θα κινηθούν στο επόμενο καρέ του video. Η λογική πράξη NOT σε grayscale εικόνες υλοποιείται με αφαίρεση βρίσκεται από τη σχέση:

$$I_o(\mathbf{x},t) = ramp(I_{t_0}(\mathbf{x},t) - I_{t_N}(\mathbf{x},t))$$

όπου η συνάρτηση $ramp(x)$ ορίζεται σύμφωνα με την γνωστή βηματική συνάρτηση $u(x)$ ως:

$$ramp(x) = x \cdot u(x)$$

Τις ονομάζουμε ακίνητες ακμές αντικειμένου και το αποτέλεσμα φαίνεται στο Σχήμα 13.

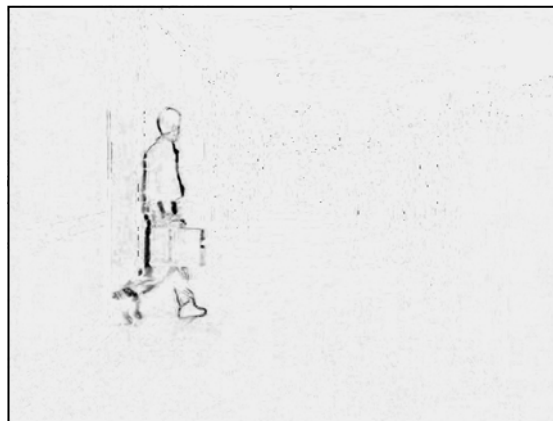


Σχήμα 13. Ακίνητες ακμές αντικειμένου

Έτσι χρησιμοποιούμε τις κινούμενες ακμές (Σχήμα 8) και τις ακίνητες ακμές αντικειμένου (Σχήμα 13) και ολοκληρώνουμε το περίγραμμα των αντικειμένων που κινούνται. Χρησιμοποιούμε το φίλτρο "μεγίστου" και δίνεται από τη σχέση:

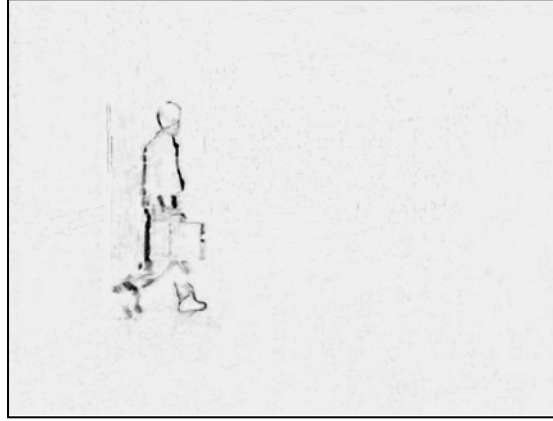
$$I_f(\mathbf{x}, t) = \max \{I_o(\mathbf{x}, t), I_m(\mathbf{x}, t)\}$$

Το αποτέλεσμα της διαδικασίας αυτής φαίνεται στο Σχήμα 14.



Σχήμα 14. Ολοκληρωμένη ανίχνευση ακμών

Προκειμένου να μειώσουμε το θόρυβο που έχει εμφανιστεί εφαρμόζουμε το μη γραμμικό μορφολογικό φίλτρο του μέσου (median) στην παραπάνω εικόνα. Το φίλτρο αυτό έχει ως αποτέλεσμα τη μείωση του θορύβου, ενώ ταυτόχρονα διατηρεί τις ακμές που έχουμε βρει. Το αποτέλεσμα του φίλτρου φαίνεται στο Σχήμα 15.



Σχήμα 15. Οι ακμές με μειωμένο θόρυβο

Συνοπτικά η παραπάνω διαδικασία φαίνεται στις παρακάτω μαθηματικές εκφράσεις και στο Σχήμα 16:

$$\bar{I}(\mathbf{x}, t) = \frac{\sum_{a=N_1}^{a=N_2} I(\mathbf{x}, a)}{N_2 - N_1} \quad (\text{ακμές σκηνης})$$

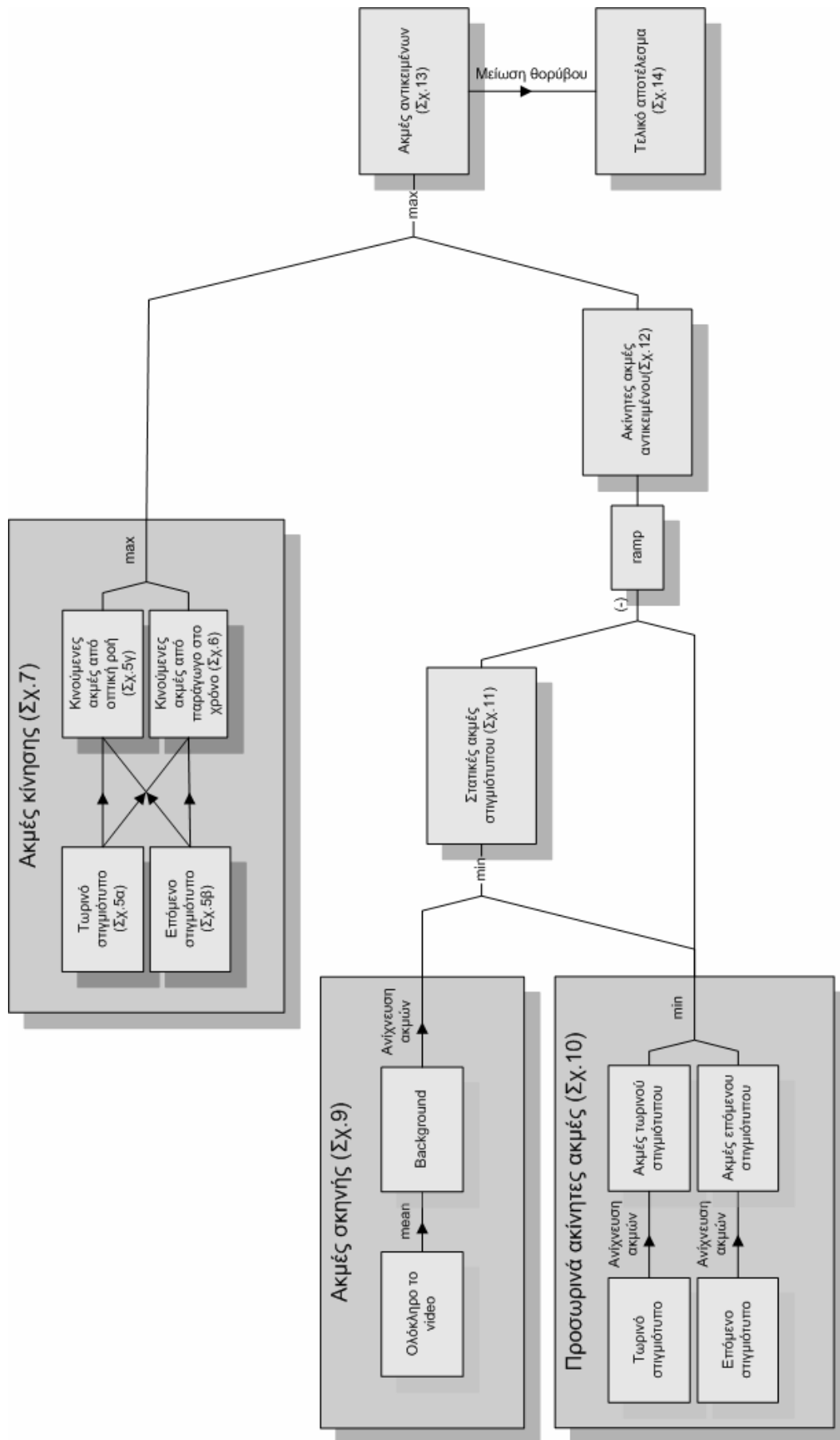
$$I_{t_0}(\mathbf{x}, t) = \min\{Edges[I(\mathbf{x}, t+1)], Edges[I_E(\mathbf{x}, t)]\} \quad (\text{προσωρινά ακίνητες ακμές})$$

$$I_{t_N}(\mathbf{x}, t) = \min\{Edges[\overline{I(\mathbf{x}, t)}], I_{t_0}(\mathbf{x}, t)\} \quad (\text{στατικές ακμές στιγμιότυπου})$$

$$I_o(\mathbf{x}, t) = ramp(I_{t_0}(\mathbf{x}, t) - I_{t_N}(\mathbf{x}, t)) \quad (\text{ακίνητες ακμές αντικειμένου})$$

$$I_m(\mathbf{x}, t) = \max\{I_v(\mathbf{x}, t), I_t(\mathbf{x}, t)\} \quad (\text{ακμές κίνησης})$$

$$I_f(\mathbf{x}, t) = \max\{I_o(\mathbf{x}, t), I_m(\mathbf{x}, t)\} \quad (\text{Ακμές αντικειμένων})$$



Σχήμα 16. Σχεδιάγραμμα της διαδικασίας εύρεσης των ακμών

Πειραματική εφαρμογή

Η παραπάνω διαδικασία (Σχήμα 16) παίρνει ως είσοδο ολόκληρο το video μιας σκηνής και εξάγει τις ακμές των αντικειμένων που περιέχονται σε αυτό. Για την υλοποίησή του χρησιμοποιήθηκε αποκλειστικά η γλώσσα προγραμματισμού C για λόγους ευελιξίας και ταχύτητας. Για λόγους ευκολίας και άμεσης αντιπαράθεσης αποτελεσμάτων, τόσο η είσοδος όσο και η έξοδος του προγράμματος είναι τα καρέ του video υπό την μορφή ασπρόμαυρων ασυμπίεστων εικόνων.

Οι κυριότερες συναρτήσεις που χρησιμοποιήθηκαν για την υλοποίηση αναλύονται παρακάτω:

- **Ανίχνευση ακμών**

Η συνάρτηση αυτή ανιχνεύει τις ακμές της εικόνας εισόδου βάση των μασκών Kirsch 3×3 και δημιουργεί μια καινούργια εικόνα για τα αποτελέσματα. Πρέπει να αναφερθεί ότι πρόκειται για αργή συνάρτηση γιατί προϋποθέτει τον υπολογισμό και συνδυασμό τεσσάρων εικόνων στις διαστάσεις της εικόνας προκειμένου να υπολογιστεί το τελικό αποτέλεσμα.

- **Παράγωγος στο χρόνο**

Η συνάρτηση της παραγώγου στο χρόνο παίρνει ως είσοδο δυο διαδοχικά καρέ του video και επιστρέφει την απόλυτη τιμή της διαφοράς τους σε εικόνα ιδίου μεγέθους.

- **Φίλτρο MAX**

Η συνάρτηση αυτή παίρνει ως είσοδο δυο εικόνες και δημιουργεί μια καινούργια εικόνα που το αποτέλεσμα της σε κάθε σημείο της εικόνας είναι αυτό με τη μεγαλύτερη ένταση, μεταξύ των αντίστοιχων σημείων των αρχικών εικόνων.

- **Φίλτρο MIN**

Η συνάρτηση αυτή παίρνει ως είσοδο δυο εικόνες και δημιουργεί μια καινούργια εικόνα που το αποτέλεσμα της σε κάθε σημείο της εικόνας είναι αυτό με τη μικρότερη ένταση ανάμεσα στα αντίστοιχα σημεία των εικόνων εισόδου.

- **Φίλτρο NOT**

Η συνάρτηση αυτή παίρνει ως είσοδο δυο εικόνες και δημιουργεί μια καινούργια εικόνα που το αποτέλεσμα της σε κάθε σημείο της εικόνας είναι η διαφορά ανάμεσα στην πρώτη και δεύτερη εικόνα, με μηδενισμό των αρνητικών τιμών.

- **Φίλτρο MEDIAN**

Η συνάρτηση αυτή παίρνει ως είσοδο μια εικόνα και δημιουργεί μια καινούργια εικόνα που το αποτέλεσμα της σε κάθε σημείο της εικόνας είναι το «μεσαίο» στοιχείο μιας γειτονιάς γύρω από το αντίστοιχο σημείο της αρχικής εικόνας. Ως «μεσαίο» στοιχείο ονομάζουμε το στοιχείο που βρίσκεται στη μέση, αν ταξινομηθούν όλα τα σημεία βάση της έντασης τους. Για κάθε σημείο της εικόνας χρειάζεται να γίνει ταξινόμηση 9 στοιχείων (3×3 γειτονιά) που κάνει τον αλγόριθμο αυτό αρκετά πολύπλοκο και σχετικά αργό.

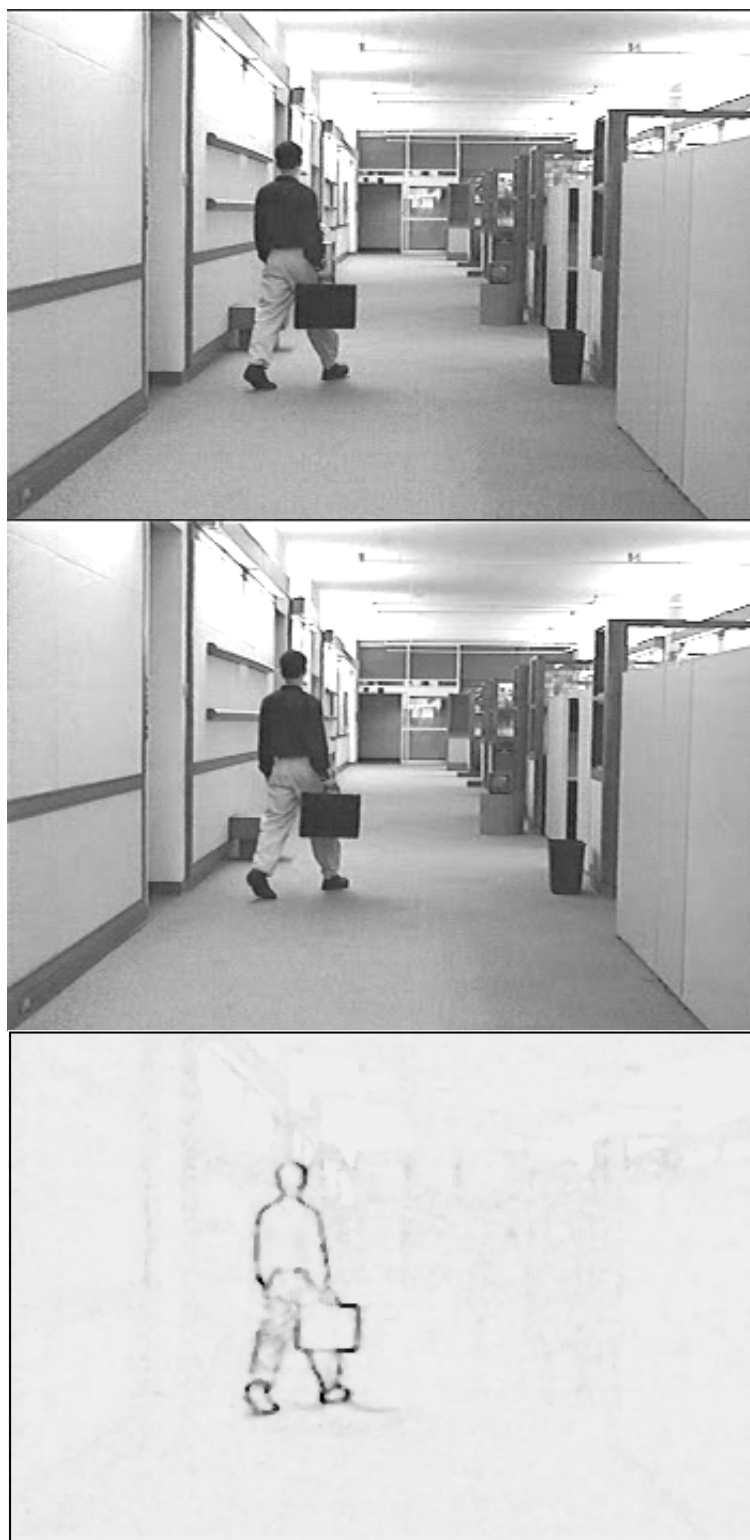
- **Διανύσματα κίνησης**

Για την εξαγωγή των διανυσμάτων κίνησης, χρησιμοποιήθηκε το επαναληπτικό σχήμα που βρήκαμε από την ανάλυση των διανυσμάτων κίνησης. Τα αποτελέσματα ήταν ικανοποιητικά. Ο αλγόριθμος κατάφερε να αντιμετωπίσει κάθε περίπτωση. Δεδομένης της σύγκλισης του αλγόριθμου, χρησιμοποιήσαμε την αλλαγή των διανυσμάτων κίνησης ύστερα από κάθε επανάληψη για να οριοθετήσουμε το τέλος της διαδικασίας. Δηλαδή σε κάθε

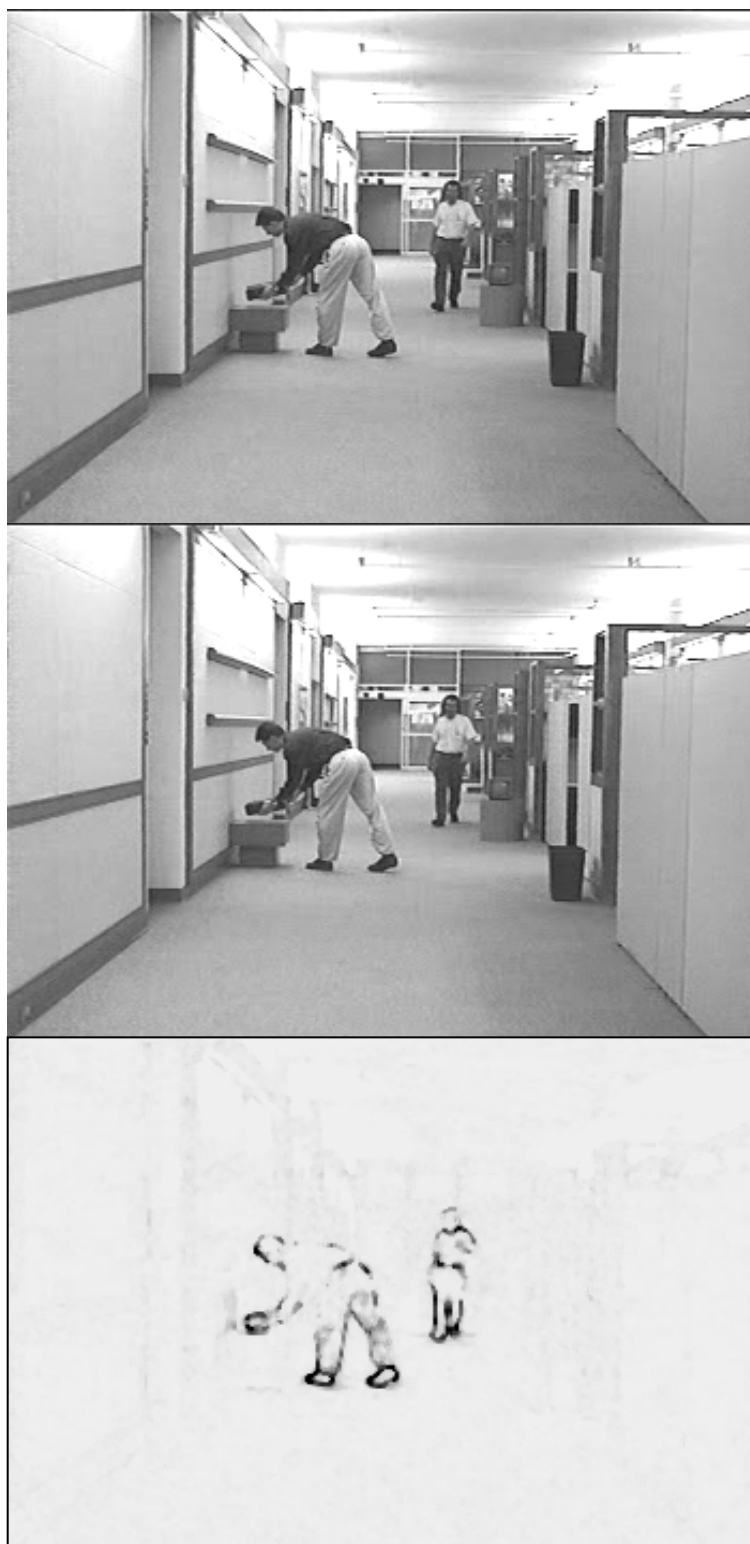
επανάληψη υπολογίζουμε το $\sqrt{\left(\frac{\partial \mathbf{u}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{u}}{\partial y}\right)^2}$ για κάθε διάνυσμα

και αθροίζουμε όλα τα αποτελέσματα. Η διαδικασία τελειώνει όταν σταματήσουν να αλλάζουν θέση τα διανύσματα κίνησης ύστερα από κάθε επανάληψη ή καλύτερα όταν μεταβάλλονται αργά.

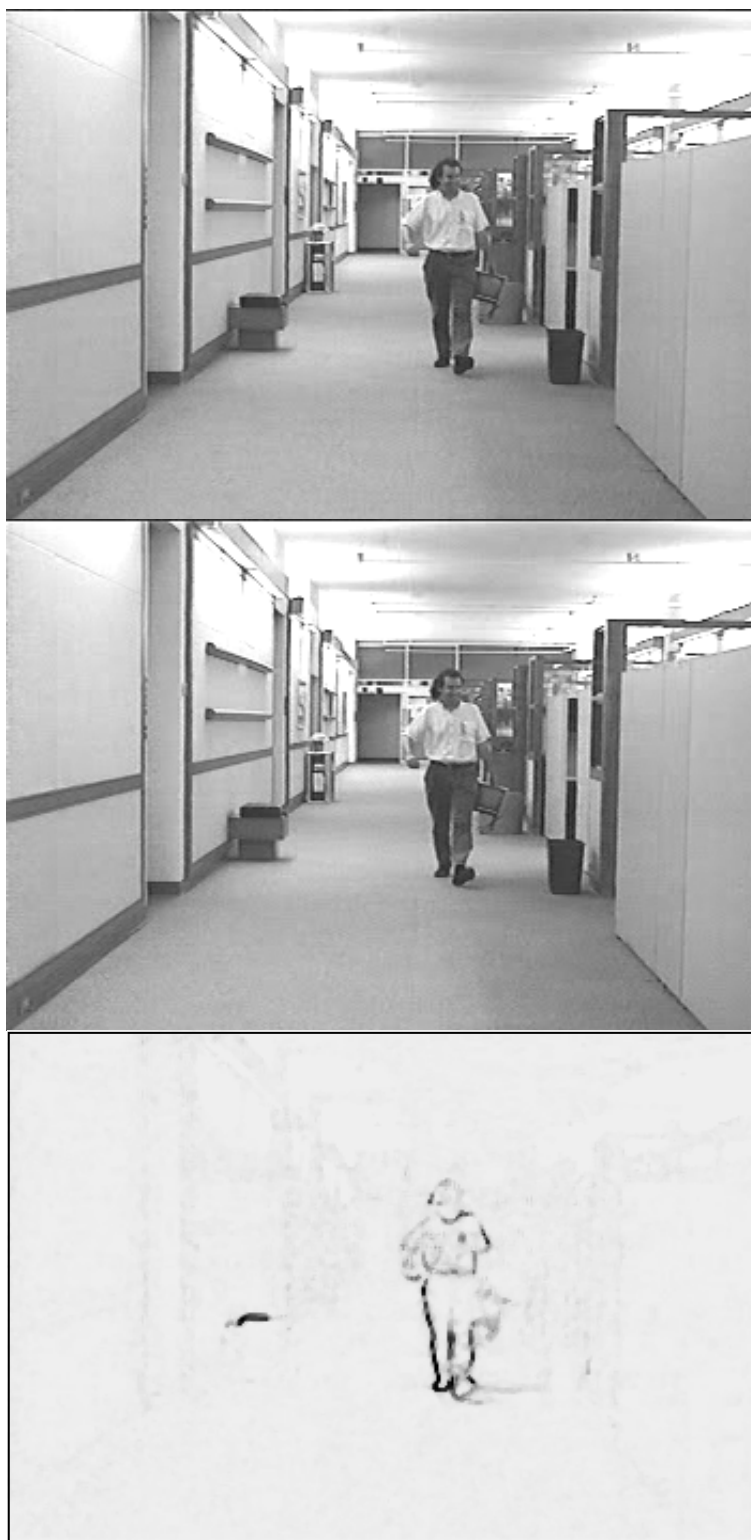
Για την εφαρμογή του αλγόριθμου χρησιμοποιήθηκαν τρία διαφορετικά video, που περιείχαν σκηνές από ακίνητη κάμερα και έχουν χρησιμοποιηθεί παλαιότερα στη βιβλιογραφία. Πρέπει να αναφερθεί ότι η διαδικασία ήταν η ίδια και για τις τρεις σκηνές, δηλαδή τηρήθηκε αυστηρά ο αλγόριθμος που αναφέραμε παραπάνω και τα αποτελέσματα είναι η έξοδός του, χωρίς καμία εξωτερική παρέμβαση.



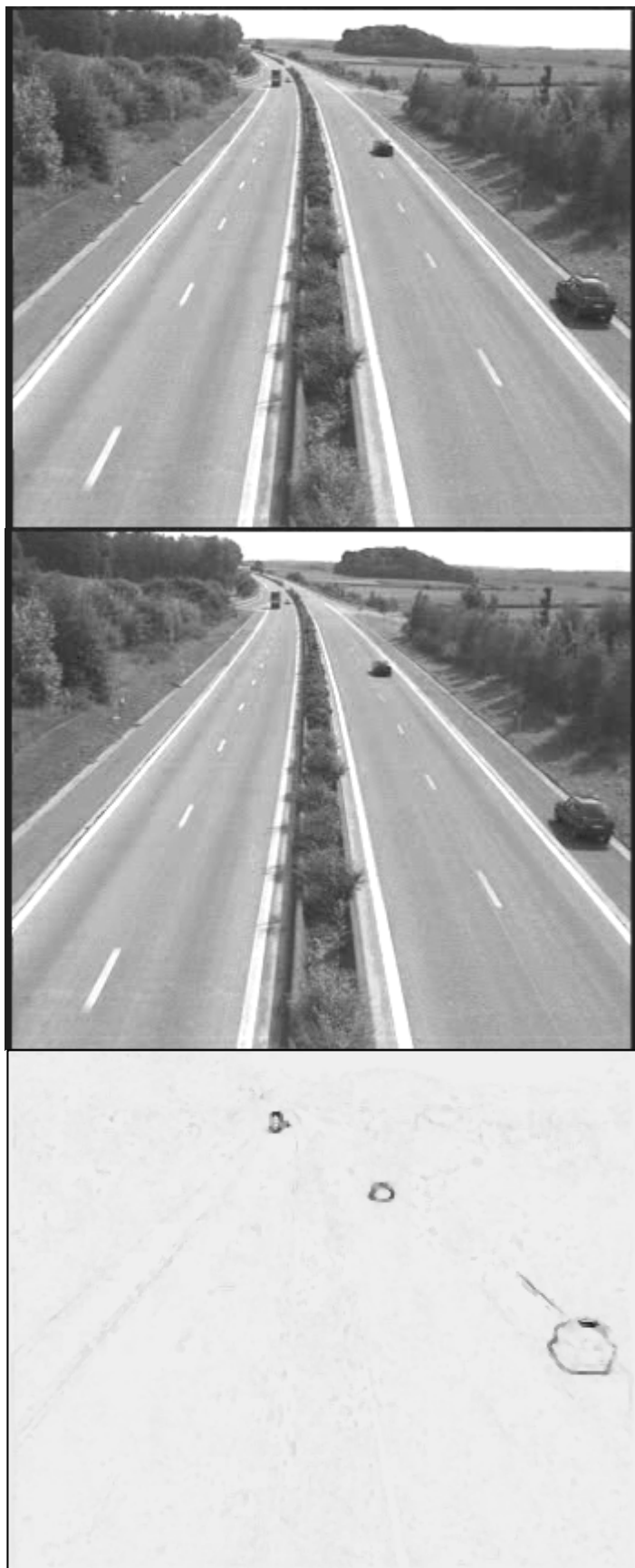
Σχήμα 17. Τα αρχικά καρέ 45-46 της σκηνής και το αποτέλεσμα



Σχήμα 18. Τα αρχικά καρέ 112-113 της σκηνής και το αποτέλεσμα



Σχήμα 19. Τα αρχικά καρέ 261-262 της σκηνης και το αποτέλεσμα



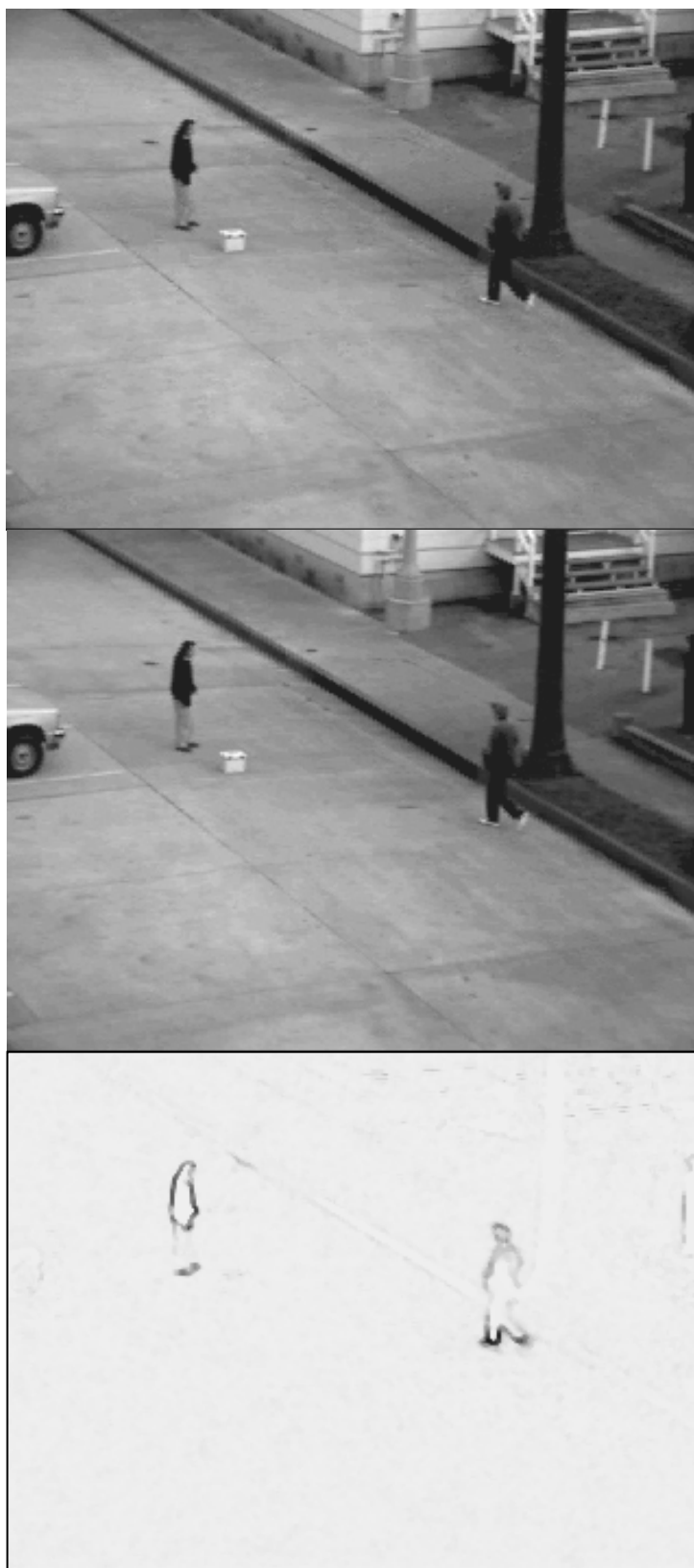
Σχήμα 20. Τα αρχικά καρέ 45 - 46 της σκηνής και το αποτέλεσμα



Σχήμα 21. Τα αρχικά καρέ 174 - 175 της σκηνής και το αποτέλεσμα



Σχήμα 22. Τα αρχικά καρέ 219 - 220 της σκηνής και το αποτέλεσμα



Σχήμα 23. Τα αρχικά καρέ 89 - 90 της σκηνής και το αποτέλεσμα



Σχήμα 24. Τα αρχικά καρέ 211 – 212 της σκηνής και το αποτέλεσμα

Αυτό που παρατηρούμε είναι ότι η ικανότητα του αλγορίθμου να εντοπίσει τις ακμές των κινούμενων αντικειμένων εξαρτάται από το

background. Έτσι ανάλογα με τη διαφορά έντασης μεταξύ αντικειμένου – background, έχουμε διαφορετική τιμή για την πρώτη παράγωγο στο χρόνο ($\frac{\partial I(\mathbf{x}, t)}{\partial t}$). Τόσο η μέθοδος ανίχνευσης κινούμενων ακμών όσο και τα διανύσματα κίνησης, έχουν μεγάλη σχέση με την παράγωγο στο χρόνο. Όταν η παράγωγος αυτή είναι πολύ μικρή, ο αλγόριθμος αδυνατεί να εντοπίσει κινούμενες ακμές.

Συμπεράσματα - Μελλοντικές επεκτάσεις

Η μέθοδος που χρησιμοποιήσαμε έδειξε ότι μπορεί να αντεπεξέλθει σε κάθε περίπτωση σκηνής με ακίνητη κάμερα. Όπως φάνηκε από τα αποτελέσματα των πειραμάτων σημαντικές ελλείψεις υπάρχουν στον τομέα του θορύβου, όπου πρέπει να μελετηθεί και να γίνουν αλλαγές στη διαδικασία που να αποσκοπούν στη μείωση του. Τα διανύσματα κίνησης χρησιμοποιήθηκαν για την ανίχνευση των κινούμενων ακμών, όμως προκαλούν σημαντική μείωση στην ταχύτητα του αλγόριθμου.

Σημαντική δουλειά έχει γίνει πάνω στην χρήση ήδη κωδικοποιημένου video κατά Mpeg για την εξαγωγή των αντικειμένων που υπάρχουν σε αυτό [10]. Το συμπιεσμένο video όμως χρειάζεται να περιέχει διανύσματα κίνησης από σωστό κωδικοποιητή, καθώς οι περισσότερες υλοποιήσεις είναι βελτιστοποιημένες για καλύτερη ταχύτητα συμπίεσης με αποτέλεσμα τα διανύσματα κίνησης να μην είναι ομαλά.

Επίσης οι μέθοδοι ανίχνευσης κινούμενων αλλά και ακίνητων ακμών, έδειξαν την αδυναμία τους να εντοπίσουν ακμές σε περιοχές όπου η ένταση του αντικειμένου δεν έχει μεγάλη διαφορά από το background. Σε αυτές τις περιοχές ισχύει ότι $\frac{\partial I(\mathbf{x}, t)}{\partial t} \approx 0$, πράγμα που καθιστάει αδύνατο τον εντοπισμό τόσο κίνησης όσο και ακμών.

Η βασική αδυναμία του αλγορίθμου είναι η χρήση του σε σκηνές με κινούμενη κάμερα. Αυτό απαγορεύει την χρήση του σε σχεδόν κάθε φυσική σκηνή όπου η κίνηση της κάμερας είναι δεδομένη. Ως επέκταση προτείνουμε την χρήση αλγορίθμου για την ανίχνευση κίνησης της κάμερας και κατάλληλη ολίσθηση των καρτέ ωστε να συμπίπτουν.

Η μελέτη έγινε χωρίς να ληφθεί υπόψη η εξαγωγή του θορύβου που περιέχει τόσο το αρχικό video, όσο και οι διαδικασίες που χρησιμοποιήσαμε. Αυτό εξηγεί και τα υψηλά επίπεδα θορύβου που περιέχονται σε κάθε στάδιο της διαδικασίας. Επέκταση της μελέτης αυτής, λοιπόν, είναι τεχνικές που να αποσκοπούν στην μείωση του θορύβου σε κάθε στάδιο της διαδικασίας.

Για την μελέτη αυτή χρησιμοποιήθηκαν εξ' ολοκλήρου ασπρόμαυρες σκηνές (grayscale). Το χρώμα στην εικόνα περιέχει μεγάλη πληροφορία που είναι πάρα πολύ χρήσιμη και σημαντική. Μελλοντική επέκταση του αλγορίθμου αυτού, είναι η μελέτη και ο συνδυασμός της συμπεριφοράς του σε κάθε χρωματική συνιστώσα ξεχωριστά. Η χρήση του χρώματος της εικόνας θα βοηθήσει επίσης στη μείωση των περιοχών με παράγωγο στο χρόνο ίση με το μηδέν, πράγμα που θα κάνει πιο αποδοτικό τον αλγόριθμο.

Οι έξοδοι του προγράμματος είναι οι κινούμενες ακμές της εικόνας. Προκειμένου να οριοθετήσουμε τα αντικείμενα από τις ακμές τους, λύση μπορούν να δώσουν τα "φίδια", που είναι δυναμικά κινούμενες κλειστές περιοχές. Στηρίζονται στην ένταση της εικόνας και μπορούν να περιγράψουν τα αντικείμενα που έχουμε ανιχνεύσει, ολοκληρώνοντας το περίγραμμα [15].

Επίσης σημαντική βελτίωση στην ανίχνευση των ακμών θα είναι η χρήση block – matching αλγορίθμων για την εύρεση των διανυσμάτων κίνησης, καθώς λειτουργούν πιο αποδοτική και ανεξάρτητα του background της εικόνας.

Ακόμη υποσχόμενη είναι η χρήση τεχνικών που αποσκοπούν τόσο στην επεξεργασία (διόρθωση, λέπτυνση) των ακμών όσο και στην σύνδεσή τους, με αποτέλεσμα την πλήρη περιγραφή του κινούμενου αντικειμένου.

Παραπομπές

- [1]. J.Canny, Finding Edges and Lines in Images, Report, AI-TR-720, M.I.T. Artificial Intelligence Lab., Cambridge, 1983.
- [2]. J.Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8, vol. 6, pp. 679-698, 1986.
- [3]. R.Deriche, Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector, International Journal of Computer Vision, vol. 1, no. 2, pp. 167-187, 1987.
- [4]. R.Deriche, Optimal Edge Detection Using Recursive Filtering, Proc. of the First International Conference on Computer Vision, London, pp. 501-505, 1987.
- [5]. R.Deriche, Fast Algorithms for Low-Level Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-12, no. 1, pp. 78-87, 1990.
- [6]. S.Castan, J.Zhao and J.Shen,, Optimal Filter for Edge Detection Methods and Results, Proc. of the First European Conference on Computer Vision, Antibes, Lecture Notes on computer Science, no. 427, pp. 12-17, 1990.
- [7]. B. K. Horn and B. G. Schunck, Determining Optical Flow, Artificial Intelligence, Vol. 17, 1981, pp. 185- 204.
- [8]. J. L. Barron, S. S. Beauchemin and D. J. Fleet, On Optical Flow, 6th Int. Conf. on Artificial Intelligence and Information-Control Systems of Robots (AIICSR),Bratislave, Slovakia, pp. 3-14, 1994
- [9]. S.S. Beauchemin and J.L. Barron, The computation of optical flow, ACM Computing Surveys, Vol. 27, No.3, pp. 433-467, 1995
- [10]. H. Zen, T. Hasegawa, S. Ozawa, Moving object detection from Mpeg coded picture, IEEE Computer Society, Volume IV,pp. 25-29, 1999
- [11]. H.W. Haussecker and D.J.Fleet, Computing Optical Flow with Physical Models of Brightness Variation, IEEE Conference on Computer Vision and Pattern Recognition, vol. II, pp. 760-767, 2000
- [12]. M. Fleury, A. C. Downton and A. F. Clark, Confidence testing optical flow estimates, Electronics Letters 34:5, 446-7, 1998
- [13]. S. Ghosal and P. Vanek, Fast Algebraic Multigrid for Discontinuous Optical Flow Estimation, 1994
- [14]. L. Biegler, J. Nocedal, C. Schmitt and D. Ternet, Numerical experience with a reduced (Hessian) method for large scale constrained optimization, 1997
- [15]. B. Galvin, B. McCane and K. Novins, Virtual Snakes for Occlusion Analysis, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999.

- [16]. B. Galvin, B. McCane, K. Novins, D. Mason, S. Mills, Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms, British Machine Vision Conference
- [17]. Y. Keller, A. Averbuch, R. Kimmel, M. Israeli, Fast motion estimation using second order gradient methods, IEEE Transactions on Image Processing, October, 2002
- [18]. B. McCane, B. Galvin, K. Novins, On the evaluation of Optical Flow Algorithms, Computer Science Department, University of Otago, New Zealand
- [19]. R. Fablet, P. Bouthemy and M. Gelgon, Moving object detection in color image sequences using region – level graph labelling, IEEE Int. Conf. on Image Processing, ICIP'99, Kobe, October, 1999
- [20]. Z. Wu and C. Chen, A new Foreground extraction scheme for video streams, Dept. of Computer Science, College of Information Science and Engineering