



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**Μελέτη κίνησης αθηρωματικών πλακών καρωτίδας από
ακολουθίες υπερήχων Β-σάρωσης**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Κ. Δημόπουλος

Επιβλέπουσα: Κωνσταντίνα Σπ. Νικήτα
Αναπληρώτρια Καθηγήτρια

Αθήνα, Ιούλιος 2004



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Μελέτη κίνησης αθηρωματικών πλακών καρωτίδας από ακολουθίες υπερήχων Β-σάρωσης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Κ. Δημόπουλος

Επιβλέπουσα: Κωνσταντίνα Σπ. Νικήτα
Αναπληρώτρια Καθηγήτρια

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21^η Ιουλίου 2004.

.....
Κ. Νικήτα
Αναπληρώτρια Καθηγήτρια

.....
Ν. Ουζούνογλου
Καθηγητής

.....
Δ. Κουτσούρης
Καθηγητής

Αθήνα, Ιούλιος 2004

.....
Αλέξανδρος Κ. Δημόπουλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αλέξανδρος Δημόπουλος, 2004.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Κεφάλαιο 1°

ΑΘΗΡΟΣΚΛΗΡΩΣΗ	9
1.1 Εισαγωγή.....	9
1.2 Επιδημιολογία - παράγοντες κινδύνου.....	11
1.2.1 Κάπνισμα.....	12
1.2.2 Υπέρταση.....	12
1.2.3 Σακχαρώδης Διαβήτης.....	13
1.2.4 Χοληστερόλη ορού.....	13
1.3 Μορφολογία υγιούς και αθηροσκληρυντικής αρτηρίας.....	14
1.3.1 Η αρτηρία.....	14
1.3.2 Μορφολογία αθηροσκληρυντικής βλάβης.....	17
1.4 Θεωρίες ανάπτυξης της αθηροσκλήρωσης.....	18
1.4.1 Η θεωρία της απόκρισης στον τραυματισμό.....	18
1.4.2 Η θεωρία της οξειδωτικής τροποποίησης.....	20
1.4.3 Πρόσφατες απόψεις για την αθηρογένεση.....	21
1.5 Διαβήτης και αθηροσκλήρωση.....	22
1.5.1 Υπεργλυκαιμία και αθηροσκλήρωση.....	22
1.5.2 Οξειδωτικό στρες και διαβήτης.....	23
1.6 Κίνηση της αθηρωματικής πλάκας.....	24
1.6.1 Μελέτη της κίνησης της αθηρωματικής πλάκας.....	26
1.6.2 Αρχή λειτουργίας ultrasound B mode.....	27
1.7 Σκοπός.....	30

Κεφάλαιο 2°

ΑΝΑΛΥΣΗ ΚΙΝΗΣΗΣ	31
2.1 Επεξεργασία ιατρικών εικόνων.....	31
2.2 Τεχνικές εκτίμησης κίνησης.....	31
2.2.1 Τεχνικές επαναλαμβανόμενου pixel.....	33
2.2.2 Τεχνικές block matching.....	34
2.2.3 Τεχνικές στο πεδίο της συχνότητας.....	35
2.3 Οπτική Ροή.....	35
2.3.1 Βασικές αρχές.....	36
2.3.1.1 Περιοριστικές παραδοχές.....	37
2.3.1.2 Περιοριστική παραδοχή Horn and Schunck.....	37
2.3.2 Υπολογισμός διαφορών.....	39
2.3.2.1 Υπολογισμός χωρικών διαφορών.....	39
2.3.2.2 Υπολογισμός χρονικών διαφορών.....	40
2.3.3 Υπολογισμός optical flow.....	41
2.3.4 Επιλογή σημείων.....	41
2.4 Υλοποίηση αλγορίθμου - Εφαρμογή.....	42
2.4.1 Επιλογή ROI.....	42

2.4.2 Επιλογή σημείων ενδιαφέροντος	43	
2.4.3 Εμφάνιση πληροφοριών για τα σημεία ενδιαφέροντος.....	44	
2.4.4 Αποθήκευση πληροφοριών για τα σημεία ενδιαφέροντος	46	
2.4.5 Αποθήκευση/ανάκτηση επιλογών ROI και σημείων ενδιαφέροντος.	46	
 Κεφάλαιο 3°		
ΑΣΘΕΝΕΙΣ - ΑΠΟΤΕΛΕΣΜΑΤΑ	49	
3.1 Ασθενείς και καταγραφή εικόνων	49	
3.2 Αποτελέσματα.....	50	
 Κεφάλαιο 4°		
ΣΥΖΗΤΗΣΗ	59	
4.1 Αξιολόγηση της αναπτυχθείσας μεθόδου	59	
4.2 Αδυναμίες της αναπτυχθείσας μεθόδου.....	60	
4.3 Προτάσεις για περαιτέρω βελτίωση της μεθόδου	62	
4.4 Σχολιασμός αποτελεσμάτων.....	63	
4.5 Συμπέρασμα	64	
 Κεφάλαιο 5°		
ΚΩΔΙΚΑΣ	67	
5.1 Γενικά	67	
5.2 MainFrame.cpp/h	67	
5.3 CImage.cpp/h.....	67	
5.4 stdafx.cpp/h	68	
5.5 Resource.h	68	
5.6 GdiPlus.h.....	68	
5.7 ChildView.cpp/h.....	68	
 ΒΙΒΛΙΟΓΡΑΦΙΑ		75
 Παράρτημα Α		81
ChildView.cpp	81	
ChildView.h	113	
Cimage.cpp	116	
Cimage.h	118	
MainFrame.cpp.....	119	
MainFrame.h.....	121	
ProgDlg.cpp	122	
ProgDlg.h	123	
Resource.h	124	
stdafx.cpp.....	126	
stdafx.h.....	126	

Παράρτημα Β.....	129
Ασυμπτωματικές πλάκες.....	129
Συμπτωματικές πλάκες.....	130
Περίληψη	133
Περίληψη στα Αγγλικά.....	135

Κεφάλαιο 1°

ΑΘΗΡΟΣΚΛΗΡΩΣΗ

1.1 Εισαγωγή

Οι καρδιαγγειακές παθήσεις αποτελούν τη σημαντικότερη πηγή νοσηρότητας και θνησιμότητας στο δυτικό κόσμο. Μόνο για το έτος 1997, το τελευταίο για το οποίο υπάρχουν διαθέσιμα στατιστικά στοιχεία, οι καρδιαγγειακές παθήσεις προκάλεσαν το 41% όλων των θανάτων στις Ηνωμένες Πολιτείες¹. Το μέγεθος του προβλήματος είναι συγκλονιστικό. Οι καρδιαγγειακές ασθένειες στοιχίζουν κάθε χρόνο περισσότερες ζωές από όσες στοιχίζουν συνολικά οι υπόλοιπες κύριες αιτίες θανάτου. Περίπου το $\frac{1}{6}$ των ανθρώπων που χάνουν τη ζωή τους από καρδιαγγειακές παθήσεις είναι κάτω από την ηλικία των 65 ετών, και η κατά εκτίμηση εξάπλωση των καρδιαγγειακών παθήσεων, με βάση την ηλικία, είναι 30% για τους άνδρες, και 24% για τις γυναίκες. Στον μαύρο πληθυσμό, αυτός ο αριθμός εκτινάσσεται στο 41% για τους άνδρες, και στο 40% για τις γυναίκες¹. Μόνο στις Ηνωμένες Πολιτείες, ξοδεύονται περισσότερα από 20 δισεκατομμύρια δολάρια για τις άμεσες δαπάνες υγειονομικής περίθαλψης των καρδιαγγειακών παθήσεων, και εάν συμπεριληφθεί η χαμένη παραγωγικότητα, το κόστος ανέρχεται σε ποσό μεγαλύτερο των 250 δισεκατομμυρίων δολαρίων ετησίως¹.

Πολλοί θεωρούν ότι πρόκειται κυρίως για ένα πρόβλημα του αναπτυσσόμενου κόσμου, όμως η Παγκόσμια Οργάνωση Υγείας (World Health Organization, WHO) προβλέπει ότι η πορεία προς παγκόσμια οικονομική ευημερία θα οδηγήσει πιθανότατα σε μια επιδημία καρδιαγγειακών παθήσεων καθώς οι αναπτυσσόμενες χώρες υιοθετούν συνήθειες του δυτικού κόσμου. Επομένως γίνεται σαφές ότι οι καρδιαγγειακές παθήσεις είναι ένα σημαντικό πρόβλημα

δημόσιας υγείας που θα έχει ευρύτερες επιπτώσεις κατά τη διάρκεια των επόμενων δεκαετιών².

Οι κύριες εκδηλώσεις των καρδιαγγειακών παθήσεων είναι η καρδιακή προσβολή και η εγκεφαλική συμφόρηση - εκδηλώσεις που αντιπροσωπεύουν την κλινική έκβαση μιας συστηματικής αγγειακής διαδικασίας γνωστής ως αθηροσκλήρωση. Στοιχεία από επιδημιολογικές μελέτες των τελευταίων 40 χρόνων φανερώνουν μια προδιάθεση του δυτικού κόσμου για καρδιαγγειακές παθήσεις. Οι επιδημιολογικές μελέτες δείχνουν ότι τα πρώιμα στάδια της αθηροσκλήρωσης μπορούν να επισπευσθούν από διάφορες κλινικές καταστάσεις, με πιο βασικές, τα υπερβολικά αυξημένα επίπεδα χοληστερόλης της χαμηλής πυκνότητας λιποπρωτεΐνης (LDL-C), τον σακχαρώδη διαβήτη, την υπέρταση, και το κάπνισμα^{3,4}.

Ένας άλλος σημαντικός παράγοντας κινδύνου πρώιμης αθηροσκλήρωσης θεωρείται το οικογενειακό ιστορικό που περιλαμβάνει συγγενή πρώτου βαθμού με πρώιμη αθηροσκλήρωση. Σαν πρώιμη αθηροσκλήρωση θεωρείται η περίπτωση ύπαρξης ενός αρσενικού γένους συγγενή πρώτου βαθμού με τα κλινικά στοιχεία της καρδιαγγειακής πάθησης σε ηλικία μικρότερη των 55 ετών ή ενός θηλυκού γένους συγγενή πρώτου βαθμού με τις κλινικές εκδηλώσεις της καρδιαγγειακής πάθησης πριν από την ηλικία των 60 ετών.

Από τους παράγοντες κινδύνου για την καρδιαγγειακή πάθηση, η υπερβολική LDL-C έχει λάβει την περισσότερη προσοχή τα τελευταία χρόνια. Η LDL-C ήταν από τους πρώτους παράγοντες κινδύνου που προσδιορίστηκαν για τις καρδιαγγειακές παθήσεις και σήμερα παραμένει κύριος στόχος η πτώση των επιπέδων της με φαρμακευτικά σκευάσματα, προκειμένου να ελαττωθεί τυχόν μελλοντικός κίνδυνος καρδιαγγειακής πάθησης².

Η αθηροσκλήρωση έχει χαρακτηριστεί ως μια δόλια διαδικασία αφού μπορεί να λανθάνει για πολλά έτη προτού να εμφανιστούν κλινικές εκδηλώσεις. Αυτό συμβαίνει διότι οι διαδικασίες που περιλαμβάνονται στην αθηρογένεση, κατά

την ανάπτυξη πρώιμων βλαβών, απαιτούν την παρατεταμένη έκθεση στους παράγοντες που προδιαθέτουν. Μόνο τα τελευταία στάδια της ασθένειας είναι αυτά που προχωρούν μάλλον γρήγορα και οδηγούν στις κλινικές εκδηλώσεις. Οι διαδικασίες ανάπτυξης αθηροσκληρυντικών βλαβών και τα κλινικά συμπτώματα της αθηροσκλήρωσης είναι διαφορετικά. Για το λόγο αυτό, θεραπείες που είναι αποτελεσματικές στην παρεμπόδιση της ανάπτυξης βλαβών μπορεί να μην αποδειχθούν αποτελεσματικές στην παρεμπόδιση των κλινικών εκδηλώσεων της ασθένειας. Ο κύριος λόγος είναι ότι στις περισσότερες των περιπτώσεων τα άτομα εισάγονται για νοσηλεία και η θεραπεία ξεκινά αφότου οι αθηροσκληρυντικές βλάβες έχουν ήδη αναπτυχθεί². Μάλιστα, δεδομένου ότι τα περισσότερα άτομα αναπτύσσουν πρώιμη αθηροσκλήρωση προτού καν γίνουν έφηβοι⁵, είναι πολύ δύσκολο να επινοηθεί μια στρατηγική που θα αποτρέψει την ανάπτυξη αθηροσκληρυντικών βλαβών.

1.2 Επιδημιολογία - παράγοντες κινδύνου

Στην επιδημιολογία, για να οριστεί ένας παράγοντας ως παράγοντας κινδύνου, ένα γνώρισμά του πρέπει να συνδεθεί ως αίτιο με μια συγκεκριμένη ασθένεια και μάλιστα με έναν τρόπο που να συσχετίζεται ισχυρά με την ασθένεια αυτή. Ακριβώς επειδή η αθηρογένεση είναι μια πολυπαραγοντική διαδικασία, η πολυπλοκότητα της καθιστά δύσκολο τον σαφή καθορισμό της επικινδυνότητας που αποδίδεται σε οποιοδήποτε παράγοντα κινδύνου. Επειδή οι περισσότεροι παράγοντες κινδύνου συνυπάρχουν σε ένα περιβάλλον με άλλους παράγοντες, η φαινομενική δύναμη της συσχέτισης εξαρτάται από το εάν οι αναλύσεις περιλαμβάνουν ή όχι τον έλεγχο για άλλες ταυτόχρονες συνθήκες που μπορεί να συμβάλουν στην ασθένεια.

Στη συνέχεια αναφέρονται περιληπτικά οι κυριότεροι παράγοντες κινδύνου.

1.2.1 Κάπνισμα

Ήδη από τις δεκαετίες του '40 και του '50 έχει αναφερθεί η συσχέτιση μεταξύ καπνίσματος και καρδιακών παθήσεων, όταν μια σειρά μελετών συσχέτισε κατηγορηματικά το κάπνισμα και τις καρδιακές παθήσεις⁶⁻⁸. Πιο πρόσφατα, εκτιμήθηκε ότι το κάπνισμα αυξάνει την αρτηριοσκληρυντική ασθένεια περισσότερο από 50% και διπλασιάζει τα περιστατικά των στεφανιαίων καρδιακών παθήσεων⁹. Αυτός ο αυξημένος κίνδυνος για στεφανιαία ασθένεια μειώνεται άμεσα μέσω της διακοπής του καπνίσματος. Μάλιστα, ο κίνδυνος για καρδιακή προσβολή στους πρώην καπνιστές μειώνεται βαθμιαία και σε διάρκεια δύο ετών ο κίνδυνος φτάνει στα επίπεδα των μη καπνιστών¹⁰.

1.2.2 Υπέρταση

Ως υπέρταση ορίζεται η κατάσταση κατά την οποία η συστολική πίεση του αίματος ξεπερνά τα 140 mm Hg ή η διαστολική πίεση του αίματος ξεπερνά τα 90 mm Hg². Οι τρέχουσες εκτιμήσεις δείχνουν ότι η υπέρταση επικρατεί περισσότερο στους μαύρους από ότι στους λευκούς, και περισσότερο στους άνδρες από στις γυναίκες. Περίπου το 30% του αμερικανικού πληθυσμού είναι υπερτασικοί. Οι ηλικιωμένοι εμφανίζονται επίσης να έχουν μια αυξανόμενη προδιάθεση για υπέρταση, με το 75% των ανθρώπων άνω των 75 ετών να χαρακτηρίζονται από αυτόν τον παράγοντα κινδύνου².

Φαίνεται να υπάρχει μια περίπου γραμμική συσχέτιση της αύξησης της πίεσης του αίματος με τα αυξανόμενα περιστατικά των αρτηριοσκληρυντικών αγγειακών ασθενειών, με μια αύξηση των 7 mm Hg της διαστολικής πίεσης αίματος να αντιστοιχεί σε μια αύξηση κατά 27% όσον αφορά στο έμφραγμα του μυοκαρδίου, και κατά 42% αύξηση όσον αφορά στην εγκεφαλική συμφόρηση¹¹. Εντούτοις η αντιυπερτασική θεραπεία έχει αποδειχθεί αποτελεσματικότερη στη μείωση των περιστατικών εγκεφαλικής συμφόρησης,

όπου μια μείωση της πίεσης του αίματος κατά 5-6 mm Hg αντιστοιχεί σε 42% μείωση του κινδύνου για εγκεφαλική συμφόρηση. Αντίθετα η ίδια μείωση της πίεσης του αίματος οδηγεί σε μείωση για τον κίνδυνο εμφράγματος του μυοκαρδίου μόλις κατά 14%¹².

1.2.3 Σακχαρώδης Διαβήτης

Σύμφωνα με μια μελέτη που περιέλαβε περίπου 14.000.000 ανθρώπους στις Ηνωμένες Πολιτείες, η αρτηριοσκληρυντική στεφανιαία νόσος είναι μια σημαντική επιπλοκή του σακχαρώδους διαβήτη. Στους ασθενείς με σακχαρώδη διαβήτη ο κίνδυνος για στεφανιαία νόσο είναι 3-5 φορές μεγαλύτερος από ότι στους μη διαβητικούς¹³. Διάφοροι άλλοι γνωστοί παράγοντες κινδύνου για τη στεφανιαία νόσο, όπως η υπέρταση και τα αυξημένα επίπεδα λιπιδίων στο αίμα, είναι επίσης πιο κοινοί στους διαβητικούς σε σχέση με το γενικό πληθυσμό¹³. Εντούτοις, παρά τη συσχέτιση αυτή, δεν μπορεί να αποδοθεί σε αυτούς τους γνωστούς παράγοντες κινδύνου περισσότερο από 25% του αυξημένου κινδύνου για ανάπτυξη στεφανιαίας αθηροσκληρυντικής νόσου¹⁴. Κατά συνέπεια, ο διαβήτης αντιπροσωπεύει έναν σημαντικό παράγοντα κινδύνου που συμβάλλει στην ανάπτυξη αθηροσκλήρωσης στον αναπτυγμένο κόσμο και ένα μεγάλο μέρος αυτού του αυξημένου κινδύνου έχει ξεφύγει του προσδιορισμού, ως αποτέλεσμα της χρησιμοποίησης αναλύσεων των παραδοσιακών παραγόντων κινδύνου. Αν και μια βελτίωση στα επίπεδα της γλυκόζης στους διαβητικούς συσχετίζεται με μείωση διάφορων επιπλοκών του διαβήτη, η στεφανιαία νόσος δε φαίνεται να ακολουθεί πάντοτε τη μείωση αυτών των επιπλοκών².

1.2.4 Χοληστερόλη ορού

Η συσχέτιση μεταξύ της χοληστερόλης και της αθηροσκλήρωσης είναι αδιαμφισβήτητη. Μεταξύ των ισχυρότερων στοιχείων για αυτήν την συσχέτιση

είναι ένα πείραμα της φύσης. Η οικογενής υπερχοληστερολαιμία (FH) είναι μια αυτοσωματική επικρατούσα διαταραχή που προσβάλλει ένα στα 500 άτομα από το γενικό πληθυσμό. Οι ετεροζυγώτες για αυτήν την ασθένεια εκδηλώνουν μια αύξηση στη χοληστερόλη του πλάσματος κατά 2-3 φορές, που οφείλεται αποκλειστικά σε αύξηση των LDL. Οι ομοζυγώτες για αυτήν την διαταραχή καταδεικνύουν μια αύξηση των επιπέδων της χοληστερόλης του πλάσματος κατά 4-6 φορές.

Ένα από τα σημαντικότερα χαρακτηριστικά γνωρίσματα της ασθένειας αυτής είναι η πρώιμη αθηροσκλήρωση. Στους ετεροζυγώτες, 85% των ατόμων έχουν τύχει ενός μυοκαρδιακού εμφράγματος πριν την ηλικία των 60 ετών, και αυτή η ηλικία μειώνεται στα 15 στους ασθενείς που είναι ομοζυγώτες για την ασθένεια αυτή. Μια παρόμοια ατέλεια έχει αναφερθεί στα κουνέλια που συσχετίζεται με πρώιμη αθηροσκλήρωση².

Όσον αφορά στους ασθενείς γενικά, περίπου 50% όλων των Αμερικανών με ηλικία μεταξύ 20 και 74 έχουν επίπεδα χοληστερόλης που υπερβαίνουν τα 200 mg/dL². Τα υψηλά επίπεδα LDL-C φαίνεται να έχουν γραμμική σχέση με τον κίνδυνο στεφανιαίων καρδιακών παθήσεων, με μια αύξηση 1% στη χοληστερόλη του ορού να αντιστοιχεί σε μια αύξηση 2% στον κίνδυνο στεφανιαίων καρδιακών παθήσεων. Αυτός ο κίνδυνος μπορεί να μειωθεί σε σημαντικό βαθμό με θεραπεία ελάττωσης των επιπέδων της χοληστερόλης.

1.3 Μορφολογία υγιούς και αθηροσκληρυντικής αρτηρίας

1.3.1 Η αρτηρία

Το υγιές αρτηριακό τοίχωμα, όπως φαίνεται στο σχήμα 1.1, αποτελείται από τρία καθορισμένα με σαφήνεια ομόκεντρα στρώματα: το ενδότατο στρώμα που καλείται έσω χιτώνας (intima), το μέσο στρώμα που καλείται μέσος χιτώνας (media) και το πιο εξωτερικό στρώμα που είναι γνωστό ως έξω

χιτώνας (adventitia). Αυτά τα τρία στρώματα οριοθετούνται από τα ομόκεντρα στρώματα της ελαστίνης, γνωστά ως εσωτερικό ελαστικό έλασμα (internal elastic lamina), το οποίο χωρίζει τον έσω από τον μέσο χιτώνα, και εξωτερικό ελαστικό έλασμα (external elastic lamina), το οποίο χωρίζει τον μέσο από τον έξω χιτώνα. Η επιφάνεια του αυλού, που αποτελεί την εσωτερική κοιλότητα των αρτηριών καλύπτεται από ένα μονό και ενιαίο στρώμα ενδοθηλιακών κυττάρων που εφάπτονται μεταξύ τους, και βρίσκεται επάνω σε μια βασική μεμβράνη της εξωκυττάριας ύλης (extracellular matrix), και οριοθετείται από το εσωτερικό ελαστικό έλασμα².



Σχήμα 1.1 Η δομή της φυσιολογικής αρτηρίας

Τα ενδοθηλιακά κύτταρα συνδέονται το ένα με το άλλο με μια σειρά κυτταρικών συνδέσμων (junctional complexes). Το ποσό της εξωκυττάριας ύλης και της ελαστίνης στο εσωτερικό ελαστικό έλασμα είναι το πιο βασικό στις μέσες και μεγάλου μεγέθους αρτηρίες. Τα ενδοθηλιακά κύτταρα διαμορφώνουν ένα εμπόδιο μεταξύ της εσωτερικής επιφάνειας του αυλού της αρτηρίας και του υποενδοθηλιακού χώρου του αρτηριακού τοιχώματος.

Ο ρόλος των ενδοθηλιακών κυττάρων θεωρήθηκε στο παρελθόν παθητικός. Σήμερα είναι γνωστό ότι τα ενδοθηλιακά κύτταρα ρυθμίζουν μια ευρεία σειρά λειτουργιών στο αρτηριακό τοίχωμα συμπεριλαμβανομένης της θρόμβωσης, του αγγειακού τόνου, και της διακίνησης των λευκοκυττάρων μέσα στο αρτηριακό τοίχωμα².

Προχωρώντας προς τα έξω από το εσωτερικό ελαστικό έλασμα, ο μέσος χιτώνας αποτελείται από λεία μυϊκά κύτταρα. Ανάλογα με το μέγεθος της αρτηρίας, υπάρχουν ένα ή περισσότερα στρώματα λείων μυϊκών κυττάρων. Τα κύτταρα διατηρούνται σε συνοχή από μια εξωκυττάρια ύλη που αποτελείται κυρίως από ίνες ελαστίνης και κολλαγόνου. Τα κύτταρα μπορούν επίσης να συνδέονται μεταξύ τους μέσω κυτταρικών συνδέσμων. Η εξωκυττάρια ύλη μέσα στο αρτηριακό τοίχωμα παράγεται κυρίως από τα λεία μυϊκά κύτταρα, και περιλαμβάνει κολλαγόνο, πρωτεογλυκάνες, και ίνες ελαστίνης².

Το μέγεθος του μέσου χιτώνα μπορεί να ποικίλει αρκετά ανάλογα της λειτουργίας μιας δεδομένης αρτηρίας. Παραδείγματος χάριν, στις πολύ μικρές αρτηρίες ο μέσος χιτώνας μπορεί να έχει πάχος μόνο ενός κυττάρου, ή να μην υπάρχει καθόλου, ενώ στις μεγάλες αρτηρίες όπως την αορτή το πάχος του μέσω χιτώνα μπορεί να είναι πολλών στρωμάτων κυττάρων και να αποτελείται από σημαντική ποσότητα εξωκυττάριας ύλης με βασικό συστατικό την ελαστίνη. Ο κύριος λόγος για αυτήν την διαφορά σχετίζεται με την ανάγκη για επαναφορά κατά τη διάρκεια της διαστολής².

Μετά το εξωτερικό ελαστικό έλασμα, υπάρχει ο έξω χιτώνας που είναι το πιο ακραίο στρώμα της αρτηρίας. Ο έξω χιτώνας αποτελείται από μια χαλαρή ύλη με μικρό ποσό ινών ελαστίνης, λεία μυϊκά κύτταρα, ινοβλάστες, και κολλαγόνο. Το μεγαλύτερο μέρος των αγγειοκινητικών νευρικών ινών του αυτόνομου νευρικού συστήματος στα αγγεία του αίματος διαπερνά αυτόν τον χιτώνα.

1.3.2 Μορφολογία αθηροσκληρυντικής βλάβης

Ο τρόπος εκδήλωσης της αθηροσκλήρωσης είναι τέτοιος, που επιτρέπει το διαχωρισμό σε τρία στάδια. Κάθε στάδιο χαρακτηρίζεται από τις ανάλογες αθηροσκληρυντικές βλάβες, που καλούνται πρώιμες, εξελισσόμενες, και ώριμες βλάβες. Οι πρώιμες βλάβες χαρακτηρίζονται από τις κονδυλώδεις περιοχές της απόθεσης λιποειδών, οι οποίες μορφολογικά έχουν ονομαστεί "λιπαρές ραβδώσεις". Οι "λιπαρές ραβδώσεις" αντιπροσωπεύουν μακροφάγα κύτταρα γεμάτα με λιποειδή, τα λεγόμενα αφρώδη κύτταρα, και λεία μυϊκά κύτταρα σε εστιακές περιοχές του έσω χιτώνα. Αυτές οι πρώιμες βλάβες αναπτύσσονται συνήθως έως την ηλικία των 10 ετών, και αυξάνονται για να καταλάβουν τουλάχιστον το 1/3 της αορτικής επιφάνειας κατά την τρίτη δεκαετία της ζωής².

Οι εξελισσόμενες βλάβες καλούνται μερικές φορές "μαργαριταρένιες πλάκες" και αντιπροσωπεύουν το επόμενο στάδιο μετά από τις "λιπαρές ραβδώσεις". Μπορούν να βρεθούν αρχικά στις περιοχές των στεφανιαίων αρτηριών, της κοιλιακής αορτής, και σε μερικές πλευρές των καρωτιδικών αρτηριών κατά την τρίτη προς την τέταρτη δεκαετία της ζωής. Αυτές οι ινώδεις πλάκες είναι θολωτές και σταθερές, και καλύπτονται από ένα ινομυϊκό στρώμα γνωστό ως "ινώδες κάλυμμα"².

Τελικά, οι βλάβες μπορούν να αναπτυχθούν σε τέτοιο βαθμό που να γίνουν περίπλοκες και προηγμένες, οι οποίες χαρακτηρίζονται από ασβεστοποιημένες ινώδεις περιοχές της αρτηρίας, με ορατό έλκος. Ο τύπος των βλαβών αυτών συνδέεται συχνά με τα κλινικά συμπτώματα ή την αρτηριακή εμβολή. Παλαιότερα υπήρχε η πεποίθηση ότι η βλάβη των οργάνων και το έμφραγμα οφείλονταν στη βαθμιαία πρόοδο αυτών των βλαβών, αλλά σήμερα γνωρίζουμε ότι διάφορα λειτουργικά συστατικά στο αρτηριακό τοίχωμα είναι επίσης σημαντικοί και καθοριστικοί παράγοντες για μια τέτοια εξέλιξη². Δηλαδή η διακοπή της ροής του αίματος στο αγγείο από

το Θρόμβο που προκαλείται μετά από τη ρήξη του τοιχώματος της αθηρωματικής πλάκας, εξαρτάται από τη σύσταση ή καλύτερα από την υφή της πλάκας. Για το λόγο αυτό σήμερα δίνεται ιδιαίτερη σημασία στη υφή των πλακών και γίνονται προσπάθειες με διάφορα φάρμακα να τροποποιηθεί η υφή, ώστε να μειώνονται οι πιθανότητες ρήξης της πλάκας και σχηματισμού Θρόμβου.

1.4 Θεωρίες ανάπτυξης της αθηροσκλήρωσης

Για την αθηροσκλήρωση έχουν διατυπωθεί διάφορες θεωρίες. Η επικρατούσα θεωρία σήμερα είναι αυτή της φλεγμονής. Η αθηροσκλήρωση συσχετίστηκε με φλεγμονή για πρώτη φορά από τον Rayer το 1823¹⁵. Στη θεωρία περί φλεγμονής έδωσε συνέχεια λίγες δεκαετίες αργότερα ο Virchow¹⁶. Το 1950 οι Saphir και Gore καταθέτουν αποδείξεις που στηρίζουν τη θεωρία της φλεγμονής¹⁷ ενώ το 1977 οι Joris και Majno τονίζουν τις ομοιότητες μεταξύ των διαδικασιών φλεγμονής και αθηροσκλήρωσης¹⁸.

1.4.1 Η θεωρία της απόκρισης στον τραυματισμό

Με βάση νεκροτομικές μελέτες, ήταν γνωστό για αρκετό χρόνο ότι η Θρόμβωση και τα λιποειδή εμπλέκονταν στην αθηροσκλήρωση. Μια από τις πρώτες θεωρίες που περιελάμβανε την Θρόμβωση ήταν η "θεωρία της επικάλυψης" ("incrustation theory") του Rokitsansky¹⁹ που πρότεινε ότι η πάχυνση του έσω χιτώνα ήταν αποτέλεσμα της απόθεσης ινώδους στο αρτηριακό τοίχωμα. Τα λιποειδή ως στοιχείο της αθηροσκλήρωσης περιελήφθησαν από μια ανταγωνιστική πιο σύγχρονη υπόθεση που εισηγήθηκε ο Virchow²⁰ ο οποίος πρότεινε ότι η μετανάστευση των λιποειδών μέσα στο αρτηριακό τοίχωμα και η σύμπλεξή τους με μυκοπολυσακχαρίτες ήταν το αρχικό στάδιο της αθηροσκλήρωσης. Ο Ross, κυρίως ως αποτέλεσμα των

μελετών για τον πολλαπλασιασμό των λείων μυϊκών κυττάρων, πρότεινε τελικά την θεωρία της "απόκρισης στον τραυματισμό"²¹.

Η επίνοια της αθηροσκλήρωσης ως ένα γεγονός που ξεκινά από τον τραυματισμό του αρτηριακού τοιχώματος, γέννησε την ιδέα ότι η ανάπτυξη των βλαβών και η πρόοδος της αρτηριοσκληρυντικής ασθένειας είναι ένα εν δυνάμει γεγονός και όχι ένα γεγονός που χαρακτηρίζεται από τον κυτταρικό θάνατο (senescence). Το 1973, οι Ross και Glomset²² επιχείρησαν να συγκεντρώσουν και να ταιριάξουν διάφορες παρατηρήσεις σε μια ενοποιημένη υπόθεση που ονόμασαν "απόκριση στον τραυματισμό". Σε αυτήν την υπόθεση, πρότειναν ότι το αρχικό στάδιο της αθηρογένεσης ήταν η απογύμνωση του ενδοθηλίου που οδηγεί σε διάφορες αντισταθμιστικές αποκρίσεις οι οποίες αλλάζουν τις φυσιολογικές ομοιοστατικές ιδιότητες των αγγείων. Σύμφωνα με αυτή τη θεωρία, ο τραυματισμός αυξάνει τη συγκολλητικότητα του ενδοθηλίου για τα λευκοκύτταρα και τα αιμοπετάλια, και μεταβάλλει το τοπικό περιβάλλον από αντιθρομβωτικό σε προθρομβωτικό. Τα προσκολλημένα λευκοκύτταρα και αιμοπετάλια απελευθερώνουν πλήθος από κυτταροκίνες, και διάφορους αγγειοδραστικούς, και αυξητικούς παράγοντες οι οποίοι ετοιμάζουν το περιβάλλον για μια φλεγμονώδη απόκριση. Αυτή η φλεγμονώδης απόκριση χαρακτηρίζεται από τη μετανάστευση και τον πολλαπλασιασμό των λείων μυϊκών κυττάρων από τον μέσο χιτώνα στην περιοχή της φλεγμονής και σχηματισμό της πρωταρχικής βλάβης. Τα μακροφάγα προσελκύονται στο αρτηριακό τοίχωμα και λαμβάνουν τις LDL σχηματίζοντας τα γνωστά "αφρώδη κύτταρα", που αποτελούν το χαρακτηριστικό γνώρισμα μιας πρώιμης αρτηριοσκληρυντικής βλάβης. Η συσσώρευση των αφρωδών κυττάρων διαιωνίζει τη φλεγμονώδη απόκριση η οποία οδηγεί σε μια εντοπισμένη συσσώρευση κυττάρων στο αρτηριακό τοίχωμα που ομοιάζει με ένα απόστημα που θα παρατηρήτο σε άλλους ιστούς. Οι κύριοι κυτταρικοί μεσολαβητές

αυτής της πρώιμης φλεγμονώδους απόκρισης είναι τα αιμοπετάλια²³, τα μακροφάγα κύτταρα, τα Τ-λεμφοκύτταρα και τα λεία μυϊκά κύτταρα²⁴.

Μέσα σε αυτό το εντοπισμένο τραύμα η συνεχής φλεγμονή μπορεί να οδηγήσει στη νέκρωση των κυττάρων και την περαιτέρω προσκόλληση των μονοκυττάρων και των λεμφοκυττάρων με μια συνακόλουθη απελευθέρωση κυτταροκινών, αυξητικών παραγόντων, και πρωτεολυτικών ενζύμων. Αυτό μπορεί να αποτελέσει την έναρξη για εστιακή νέκρωση μέσα στο τραύμα, και μπορεί να προκαλέσει την αυτοκαταλυτική επέκταση του τραύματος. Καθώς το τραύμα διευρύνεται δεν μπορεί πλέον να αντισταθμιστεί η εισβολή των στοιχείων που απαρτίζουν τη βλάβη στον αρτηριακό αυλό, με αποτέλεσμα η ροή του αίματος να μειώνεται.

Η θεωρία της απόκρισης στον τραυματισμό βασίστηκε αρχικά στην απογύμνωση του ενδοθηλίου ως κύριο στάδιο για την έναρξη της αθηροσκλήρωσης²⁵. Εντούτοις πιο πρόσφατα, έγινε σαφές ότι αναπτυσσόμενες αρτηριοσκληρυντικές βλάβες, είναι δυνατόν να καλύπτονται από ένα άθικτο στρώμα ενδοθηλιακών κυττάρων, και ότι η απογύμνωση του ενδοθηλίου αποτελεί την εξαίρεση παρά τον κανόνα. Σε μια πρόσφατη βελτίωση της αρχικής υπόθεσής του, ο Ross πρότεινε ότι η δυσλειτουργία των ενδοθηλιακών κυττάρων είναι επαρκής για την έναρξη της αθηρογένεσης μέσω της αυξημένης διαπερατότητας του ενδοθηλίου από τις αθηρογόνες λιποπρωτεΐνες²⁶.

1.4.2 Η θεωρία της οξειδωτικής τροποποίησης

Σύμφωνα με τη θεωρία της "απόκρισης στον τραυματισμό" ο σχηματισμός των αφρωδών κυττάρων αποτελεί το χαρακτηριστικό γνώρισμα της αθηροσκλήρωσης. Η θεωρία της "οξειδωτικής τροποποίησης" προέκυψε από τη θεωρία της "απόκρισης στον τραυματισμό" ως αποτέλεσμα της εντατικής

έρευνας όσον αφορά τους μηχανισμούς του σχηματισμού των αφρωδών κυττάρων.

1.4.3 Πρόσφατες απόψεις για την αθηρογένεση

Σήμερα έχει αναπτυχθεί μια νέα παγκοσμίως θεωρία για τον τρόπο σχηματισμού των αθηρωματικών πλακών, μέσα από μια νέα προσέγγιση στο θέμα²⁷

Η θεωρία αυτή ενοποιεί τις υπάρχουσες θεωρίες που δέχονται τη φλεγμονή σαν αίτιο για την αθηρωμάτωση και την αρτηριοσκλήρυνση, υποδεικνύει το φλεγμονώδη παράγοντα (PAF, Platelet-Activating Factor) που είναι υπεύθυνος για την πρόκληση της αθηρωμάτωσης και συμπληρώνει το παζλ των γνώσεων, ώστε να είναι δυνατή η εξήγηση των σχετικών πειραματικών δεδομένων που υπάρχουν στη διεθνή βιβλιογραφία, χωρίς τα κενά που παρουσίαζαν οι μέχρι σήμερα αποδεκτές θεωρίες.

Η γενικευμένη αυτή θεωρία, εξηγεί επίσης- με βιοχημικό τρόπο - τη γνωστή στατιστική και επιδημιολογική παρατήρηση ότι η Μεσογειακή Δίαιτα προστατεύει από το σχηματισμό αθηρωματικών πλακών, την αρτηριοσκλήρυνση και τις καρδιαγγειακές παθήσεις. Με βάση τη θεωρία αυτή, πιστεύεται ότι δεν είναι η χοληστερίνη και τα κορεσμένα λιποειδή το γενεσιουργό αίτιο του σχηματισμού των αθηρωματικών πλακών, αλλά ο PAF. Τα υψηλά επίπεδα χοληστερίνης και κορεσμένων λιποειδών στο αίμα εντείνουν την οξείδωση της LDL και κατά συνέπεια την παραγωγή PAF με τα αποτελέσματα που προαναφέραμε. Η προστασία μας από τα υψηλά επίπεδα χοληστερίνης και κορεσμένων λιποειδών μπορεί να γίνεται με αναστολείς του PAF από τη Μεσογειακή δίαιτα.

1.5 Διαβήτης και αθηροσκλήρωση

1.5.1 Υπεργλυκαιμία και αθηροσκλήρωση

Τόσο ο τύπος I όσο και ο τύπος II του διαβήτη αποτελούν ισχυρούς και ανεξάρτητους παράγοντες κινδύνου για τις καρδιοαγγειακές παθήσεις^{28,29}. Η αθηροσκλήρωση ευθύνεται για περίπου το 80% των θανάτων ασθενών που πάσχουν από διαβήτη στη βόρεια Αμερική. Αντιθέτως, στο γενικό πληθυσμό της βόρειας Αμερικής το ποσοστό ανέρχεται στο ένα τρίτο των θανάτων²⁸. Επιπλέον το 75% όλων των εισαγωγών στο νοσοκομείο για επιπλοκές που αφορούν σε διαβήτη αποδίδεται στις καρδιοαγγειακές παθήσεις.

Αν και ο αυξανόμενος κίνδυνος για αθηροσκλήρωση που συνδέεται με το διαβήτη δεν έχει διευκρινισθεί πλήρως, ένας προτεινόμενος μηχανισμός περιλαμβάνει τα υψηλά επίπεδα γλυκόζης στο αίμα δεδομένου ότι η υπεργλυκαιμία είναι ένα χαρακτηριστικό γνώρισμα του διαβήτη. Η παρατεταμένη έκθεση στην υπεργλυκαιμία αναγνωρίζεται στις μέρες μας ως ένας σημαντικός παράγοντας στην παθογένεση των επιπλοκών του διαβήτη³⁰. Αυτή η αποκαλούμενη "υπόθεση της γλυκόζης" προτείνει ότι οι αντιδράσεις οξειδωσης που προκαλούνται από τη γλυκόζη τροποποιούν τις LDL προς μορφές που προωθούν την έναρξη της αθηροσκλήρωσης και την ανάπτυξη βλαβών. Η υπόθεση αυτή υποστηρίζεται από μελέτες που δείχνουν ότι ο αυστηρός έλεγχος της γλυκόζης σε διαβητικούς τύπου I οδηγεί σε μειωμένες αγγειακές επιπλοκές³¹.

Τα αποτελέσματα της υπεργλυκαιμίας είναι συχνά μη αναστρέψιμα και οδηγούν στην προοδευτική δυσλειτουργία των κυττάρων. Στους διαβητικούς ασθενείς μετά από επιτυχή μεταμόσχευση παγκρέατος οι παθολογικές εκδηλώσεις στους νεφρούς συνεχίζουν να προχωρούν για τουλάχιστον 5 έτη αφότου έχει θεραπευτεί ο διαβήτης³². Ο μηχανισμός για αυτές τις παρατηρήσεις είναι ασαφής, αλλά υποδηλώνει ότι οι κυτταρικές διαταραχές

μπορεί να εμμείνουν παρά την επιστροφή του οργανισμού σε φυσιολογικά επίπεδα γλυκόζης, φαινόμενο που καλείται "επίδραση μνήμης" ("memory effect"). Κατά συνέπεια, μεταβολικές αλλαγές που εμμένουν είναι κεντρικής σπουδαιότητας στην παθογένεση των επιπλοκών του διαβήτη.

Ένας από τους σημαντικούς μηχανισμούς που ευθύνεται για την επιτάχυνση της αθηροσκλήρωσης στο διαβήτη είναι η βιολογική οξείδωση από τη γλυκόζη, που προσδιορίστηκε αρχικά το 1912 από τον Maillard³³. Η αντίδραση Maillard αναφέρεται στην μη ενζυμική τροποποίηση των πρωτεϊνών από τη γλυκόζη, που συνοδεύει την υπεργλυκαιμία.

1.5.2 Οξειδωτικό στρες και διαβήτης

Η υπεργλυκαιμία μπορεί να αυξήσει το οξειδωτικό στρες μέσω διάφορων τρόπων. Το οξειδωτικό στρες θεωρείται ότι εμπλέκεται στους παθογενετικούς μηχανισμούς που αφορούν την ανάπτυξη της αθηροσκλήρωσης. Μεταξύ των δευτερογενών συμπτωμάτων της υπεργλυκαιμίας, το οξειδωτικό στρες έχει προταθεί ως πιθανός μηχανισμός για την επιτάχυνση της αθηροσκλήρωσης^{34,35}.

Υπάρχουν επίσης στοιχεία ότι η υπεργλυκαιμία μπορεί να εμποδίσει την φυσική αντιοξειδωτική άμυνα του οργανισμού. Υπό φυσιολογικές συνθήκες, οι ελεύθερες ρίζες εκκαθαρίζονται γρήγορα από αντιοξειδωτικά όπως η ανοιγμένη γλουταθειόνη, η βιταμίνη C, και η βιταμίνη E. Χαμηλή περιεκτικότητα σε γλουταθειόνη³⁶, καθώς επίσης και βιταμίνης E^{37,38} έχει αναφερθεί σε διαβητικούς ασθενείς. Τα επίπεδα βιταμίνης C στο πλάσμα και στους ιστούς είναι 40-50% χαμηλότερα σε διαβητικούς ασθενείς σε σχέση με μη διαβητικά άτομα^{37,39}.

Τα παραπάνω αποτελούν μια εξήγηση για την αυξημένη αθηροσκλήρωση που παρατηρείται σε διαβητικούς ασθενείς.

1.6 Κίνηση της αθηρωματικής πλάκας

Το κύριο πρόβλημα με τις αθηρωματικές πλάκες δεν είναι τόσο η παρεμπόδιση της ροής του αίματος. Μέχρι και 80% παρεμπόδιση της ροής του αίματος δε δημιουργεί σημαντικά προβλήματα. Ο κίνδυνος από τις αθηρωματικές πλάκες προέρχεται κυρίως από τη διαρραγή αυτών και το σχηματισμό θρόμβου οπότε γίνεται ολική παρεμπόδιση της ροής του αίματος, με αποτέλεσμα εμφάνιση εγκεφαλικών ή καρδιακών επεισοδίων.

Η πιθανότητα διαρραγής της πλάκας σχετίζεται άμεσα με την υφή της. Σήμερα δε πολλές θεραπευτικές παρεμβάσεις αποβλέπουν αποκλειστικά και μόνο στην τροποποίηση της υφής της πλάκας με ενίσχυση του ινώδους περιβλήματος της, ώστε να περιορίζεται ο κίνδυνος να διαρραγεί.

Ένας άλλος τρόπος αντιμετώπισης των αθηρωματικών πλακών είναι με χειρουργική παρέμβαση. Υπάρχουν στοιχεία πως ο καθαρισμός της καρωτίδας, με χειρουργική επέμβαση, σε ασθενείς με ασυμπτωματική στένωση της καρωτίδας ελαττώνει την πιθανότητα εγκεφαλικού επεισοδίου⁴⁰. Εντούτοις, ένας μεγάλος αριθμός ασθενών πιθανώς χειρουργείται χωρίς να είναι απαραίτητο. Επομένως, είναι αναγκαίο να ταυτοποιούνται οι ασθενείς υψηλής επικινδυνότητας, οι οποίοι θα υποβάλλονται σε καθαρισμό της καρωτίδας εν αντιθέσει με τους ασθενείς χαμηλής επικινδυνότητας, οι οποίοι θα αποφεύγουν την μη απαραίτητη, δαπανηρή αλλά και συχνά πολύ επικίνδυνη επέμβαση⁴¹.

Έχει αναφερθεί σχέση ανάμεσα στη μορφολογία των αθηρωματικών πλακών και στο ενδεχόμενο εγκεφαλικού επεισοδίου⁴²⁻⁴⁷. Στις περισσότερες από τις μελέτες [42]-[47], τα χαρακτηριστικά των πλακών είχαν καθοριστεί υποκειμενικά ή χρησιμοποιώντας απλά στατιστικά μέτρα. Επιπλέον, η συσχέτιση τους με τα συμπτώματα έγινε μέσω απλοϊκής στατιστικής ανάλυσης.

Έχει υποστηριχθεί⁴⁸ ότι η κίνηση των αθηρωματικών πλακών αποτελεί έναν επιπλέον παράγοντα κινδύνου για εμβολή. Άμεσα, μπορεί η κίνηση να προξενεί είτε την αποκόλληση τμημάτων της πλάκα μέσα στο αγγείο ή έμμεσα να αναπτύσσονται τάσεις ανάμεσα στην πλάκα και τα αρτηριακά τοιχώματα, με αποτέλεσμα τη θραύση της πλάκας και το σχηματισμό θρόμβου. Και στις δύο παραπάνω περιπτώσεις το αποτέλεσμα μπορεί να είναι η πλήρης ή σχεδόν πλήρης παρεμπόδιση της ροής του αίματος με αντίστοιχα ισχαιμικά ή αποφρακτικά κλινικά συμπτώματα.

Επομένως, η ανάλυση της κίνησης της πλάκας μπορεί να παράσχει επιπλέον γνώσεις για το μέχρι σήμερα θεωρούμενο μοντέλο των πλακών καθώς και για τους μηχανισμούς που οδηγούν στη θραύση της πλάκας και κατ' επέκταση στην εμβολή. Αυτή η ιδέα απορρέει από έρευνες που παρέχουν αποδείξεις ότι μεταβλητή ελαστικότητα των αγγειακών τοιχωμάτων, γεωμετρικές ασυμμετρίες στα σημεία διακλάδωσης και διαταραχές στη ροή του αίματος οδηγούν σε αστάθεια της πλάκας. Πιο συγκεκριμένα, ετερογενείς πλάκες αναμένεται να παρουσιάζουν διαφορετικές ελαστικές ιδιότητες με αποτέλεσμα να επιτρέπουν ή να αποτρέπουν την παραμόρφωση των πλακών, ανάλογα με το μέγεθος και τη θέση των ασβεστοποιήσεων, την ινώδη μήτρα, τα λιποειδικά συστατικά και την πιθανότητα μιας ενδοπλασματικής αιμορραγίας⁴⁹ που θα οδηγήσει στα κλινικά συμπτώματα που έχουν ήδη αναφερθεί.

Μολονότι η ακριβής αιτία της διαρραγής της πλάκας παραμένει ασαφής, πιθανούς μηχανισμούς αποτελούν οι τάσεις που αναπτύσσονται⁵⁰, ο τοπικός στροβιλισμός του αίματος^{51,52} καθώς και η αποδυνάμωση του ινώδους καλύμματος κολλαγόνου της πλάκας λόγω ενζυμικής δραστηριότητας⁵³. Οι ενδείξεις υποδηλώνουν ότι ο μηχανισμός ρήξης της πλάκας ενδέχεται να περιέχει χαρακτηριστικά θραύσης απλών υλικών με τρόπο θραύσης τυπικό για μέταλλα που έχουν υποστεί καταπόνηση⁴⁹. Παρατήρηση της σχετικής

θέσης μεταξύ δεικτών τοποθετημένων πάνω στην πλάκα κατά τη διάρκεια άσκησης πίεσης, δείχνει ότι πριν τη διαρραγή υπάρχει μη συμμετρική κίνηση⁵⁴. Πιστεύεται ότι τέτοιες κινήσεις της επιφάνεια της πλάκας οδηγούν σε παραμορφώσεις που με τη σειρά τους οδηγούν στη δημιουργία πολλαπλών ρωγμών και ακολούθως σε σκίσιμο της πλάκας. Κατά πόσο τέτοιες κινήσεις της πλάκας συμβαίνουν in vivo παραμένει σε μεγάλο βαθμό άγνωστο. Μία τέτοια περίπτωση είναι εκείνη της αρρυθμίας σε άτομα μεγάλης ηλικίας.

Η ανάλυση της κίνησης των πλακών από ακολουθίες εικόνων υπερήχων με τη βοήθεια αλγορίθμων ανίχνευσης κίνησης, έχει δώσει μέχρι στιγμής πρώιμα αποτελέσματα, στα οποία παρατηρούνται διαφορετικά μοτίβα κίνησης για συμπτωματικές και ασυμπτωματικές πλάκες⁵⁵. Οι παράμετροι αξιολόγησης της κίνησης της αθηρωματικής πλάκας είναι η μέγιστη ταχύτητα που αναπτύσσεται πάνω στην επιφάνεια της πλάκας, η διαφορά ταχύτητας μεταξύ του σημείου μέγιστης και ελάχιστης ταχύτητας καθώς και η επιτάχυνση κάθε σημείου.

1.6.1 Μελέτη της κίνησης της αθηρωματικής πλάκας

Η απεικόνιση με χρήση υπερήχων της καρωτίδας είναι ευρέως χρησιμοποιούμενη για τη διάγνωση της αθηροσκλήρωσης, επειδή επιτρέπει τη μη-επεμβατική εκτίμηση της στένωσης καθώς και της μορφολογίας της πλάκας. Η ανάλυση ψηφιακών εικόνων αθηρωματικών πλακών παρέχει ποσοτικά μέτρα για τη μορφολογία της πλάκας, τα οποία μπορούν να χρησιμοποιηθούν για την ταυτοποίηση των συστατικών της πλάκας. Επιπλέον, ακολουθίες εικόνων υπερήχων μπορούν να χρησιμοποιηθούν για την εκτίμηση της κίνησης της πλάκας.

1.6.2 Αρχή λειτουργίας ultrasound B mode

Τα ηχητικά κύματα πάνω από τη συχνότητα των 20KHz (ακουστικό όριο) καλούνται υπερηχητικά. Όταν τα υπερηχητικά κύματα διαδίδονται διαδοχικά τότε είναι δυνατόν να παράγουμε μια δισδιάστατη υπερηχητική εικόνα σε πραγματικό χρόνο. Στην εικόνα αυτή απεικονίζεται η ένταση του υπερηχητικού κύματος που επιστρέφει από κάθε σημείο του πεδίου σάρωσης σε επίπεδα του γκρίζου. Με τον τρόπο αυτό παράγονται εικόνες γκρίζας κλίμακας (gray scale) που απεικονίζουν τα σημεία στα οποία ανακλάται ο υπερηχητικός παλμός που εκπέμπεται.

Η ποιότητα της εικόνας που λαμβάνεται εξαρτάται από:

- Την αξονική διακριτική ικανότητα (axial resolution)
- Την πλευρική ή εγκάρσια διακριτική ικανότητα (lateral resolution)

Όπως και στο πεδίο της οπτικής, η διακριτική ικανότητα ορίζεται ως η ελάχιστη απόσταση μεταξύ δύο σημείων στόχων, στην οποία αυτοί να μπορούν να φαίνονται ως δύο ξεχωριστά σημεία και όχι ως ένα. Στην οπτική η διακριτική ικανότητα εξαρτάται και από το μήκος κύματος της φωτεινής πηγής.

Η αξονική διακριτική ικανότητα αντιπροσωπεύει την ελάχιστη απόσταση σε βάθος που πρέπει να χωρίζει δύο διαφορετικούς στόχους. Η αξονική διακριτική ικανότητα εξαρτάται αποκλειστικά από το μήκος του υπερηχητικού κύματος λ . Εφόσον το λ σχετίζεται με τη συχνότητα f με τη σχέση $\lambda = c/f$ παρατηρούμε ότι η αξονική διακριτική ικανότητα βελτιώνεται με μείωση του μήκους κύματος ή αντίστοιχα αύξηση της συχνότητας f .

Η πλευρική διακριτική ικανότητα αναφέρεται στην ελάχιστη απόσταση που πρέπει να χωρίζει δύο στόχους, στο κάθετο προς αυτό που διαδίδεται η δέσμη επίπεδο, ώστε αυτοί να γίνονται αντιληπτοί ξεχωριστά. Υπολείπεται σε ακρίβεια της αξονικής και εξαρτάται από τη διάμετρο της δέσμης. Για να είναι δυνατόν να διακριθούν στόχοι που βρίσκονται κοντά το εύρος της

υπερηχητικής δέσμης θα πρέπει αυτό να είναι σε λογικά πλαίσια μικρό. Αυτό είναι εφικτό όταν το πλάτος της κυματομορφή διατηρείται μικρό και η διάμετρος του μετατροπέα όσο πιο μεγάλη γίνεται.

Οι συχνότητες εκπομπής κυμαίνονται από 1 έως 10 MHz ανάλογα κάθε φορά με την εκάστοτε εφαρμογή. Έτσι για αρτηρίες που βρίσκονται κοντά στο δέρμα είναι μεγαλύτερη από 7.5 MHz, για αρτηρίες που βρίσκονται βαθύτερα το φάσμα συχνοτήτων κυμαίνεται μεταξύ 3 και 5 MHz.

Όταν επιλέγεται μια συχνότητα εκπομπής θα πρέπει να ληφθούν υπόψη τα παρακάτω :

- Η αξονική διακριτική ικανότητα είναι ανάλογη της συχνότητας του υπερηχητικού κύματος
- Η ένταση του σήματος εξαρτάται από την εξασθένηση του υπερήχου καθώς αυτός διαπερνά τον ιστό. Όσο πιο μεγάλη είναι η συχνότητα του υπερήχου τόσο πιο έντονη θα είναι η εξασθένηση του σήματος.

Ένα διαδιδόμενο κύμα όταν συναντήσει ένα μέσο με διαφορετικές μέσες τιμές ακουστικών ιδιοτήτων θα ανακλαστεί μερικώς, ενώ ένα μέρος του κύματος θα διαδοθεί μέσα στο δεύτερο μέσο, πιθανόν σε μια νέα διεύθυνση. Για κάθε μέσο διάδοσης υπάρχει ένα χαρακτηριστικό μέγεθος που ονομάζεται ακουστική εμπίδηση (acoustic impedance) Z , η οποία εξαρτάται από τις ακουστικές ιδιότητες του μέσου, όπως για παράδειγμα την πυκνότητα του ιστού και όχι από τη συχνότητα του ηχητικού σήματος. Όσο μεγαλύτερη είναι η διαφορά των ακουστικών εμπεδήσεων των δύο μέσων, τόσο μεγαλύτερη θα είναι η ανάκλαση. Ανομοιογένειες μικρού μεγέθους σε σχέση με το μήκος κύματος (π.χ τα κύτταρα του αίματος) ή γειτονικές ανομοιογένειες με διαφορετική ακουστική εμπίδηση, προκαλούν σκέδαση. Όσο πιο ισχυρή είναι η ανάκλαση που παρατηρείται σε κάποιες επιφάνειες, τόσο μικρότερο είναι το ποσοστό της υπερηχητικής ακτινοβολίας που θα μπορέσει να διεισδύσει μέσα στον ιστό. Το μεγαλύτερο μέρος της υπερηχητικής ενέργειας μετατρέπεται σε

κινητική ενέργεια μέσα στον ιστό. Αυτό βρίσκει ειδικότερα εφαρμογή σε υψηλότερες συχνότητες, όπου το ποσοστό της ενέργειας που απορροφάται αυξάνεται. Από την άλλη, λόγω του μικρότερου μήκους κύματος, οι υψίσυχνοι υπέρηχοι παρουσιάζουν μικρότερη διασπορά σε επιφάνειες, δηλαδή σε μέσα που βρίσκονται πολύ κοντά το ένα με το άλλο. Αυτό αυξάνει τοπικά τη διακριτική ικανότητα.

Ομογενείς περιοχές χαρακτηρίζονται από ηχητικά σήματα υψηλής ή μεσαίας έντασης, ενώ ετερογενείς περιοχές χαρακτηρίζονται από μικτά σήματα διαφόρων εντάσεων. Οι μαλακές αθηρωματικές πλάκες μπορεί να είναι ομογενείς ή ετερογενείς. Όμως, οι σκληρές πλάκες παράγουν ανακλάσεις υψηλής στάθμης, σκιές και συσχετίζονται με ινώδη υλικά και μικρού βαθμού αβεστοποίηση.⁵⁶

1.7 Σκοπός

Σύμφωνα με όλα τα προηγούμενα, αφ' ενός μεν είναι αποδεκτό ότι η υφή μιας αθηρωματικής πλάκας σχετίζεται άμεσα με την επικινδυνότητα αυτής, αφ' ετέρου υπάρχουν ενδείξεις ότι η μορφολογία των εικόνων που λαμβάνονται με χρήση υπερήχων από την αθηρωματική πλάκα στην καρωτίδα, μπορούν να συσχετισθούν με την υφή της πλάκας και κατά συνέπεια να έχουν αξία προγνωστική. Δηλαδή, είναι δυνατόν να αναγνωριστεί μια ομάδα ασθενών με κίνδυνο εγκεφαλικού ή καρδιακού επεισοδίου βάσει της εκτίμησης της κίνησης των εικόνων που λαμβάνονται από υπερηχογραφήματα (υψηλής ανάλυσης) της καρωτίδας⁴¹.

Σκοπό της συγκεκριμένης εργασίας αποτελεί η δημιουργία μιας εφαρμογής που θα σχετίζει την κίνηση της αθηρωματικής πλάκας με τη μορφολογία των εικόνων που λαμβάνονται με χρήση υπερήχων, ώστε να διευκολύνει το χαρακτηρισμό των αθηρωματικών πλακών σε συμπτωματικές και ασυμπτωματικές.

Ο συνδυασμός των χαρακτηριστικών που προκύπτουν από την ανάλυση της κίνησης της πλάκας αποτελούν ένδειξη ελαστικότητας της πλάκας, ιδιότητα που σχετίζεται άμεσα με την πιθανότητα εμφάνισης εγκεφαλικού ή καρδιακού επεισοδίου. Αξίζει να αναφερθεί ότι η μέτρηση της έκτασης της αθηρωμάτωσης γίνεται με εκτίμηση της ελαστικότητας των αγγείων. Η διεργασία αυτή μπορεί να γίνει σε πειραματόζωα σήμερα με απλή μεν σχετικά μέθοδο, που περιέχει όμως πολύπλοκη θεωρητική επεξεργασία και γραφικές παραστάσεις⁵⁸.

Κεφάλαιο 2°

ΑΝΑΛΥΣΗ ΚΙΝΗΣΗΣ

2.1 Επεξεργασία ιατρικών εικόνων

Η επεξεργασία ιατρικών εικόνων έχει εξελιχθεί σε μια ευρεία και σημαντική περιοχή εφαρμογών των τεχνικών επεξεργασίας ψηφιακής εικόνας. Τέτοιες τεχνικές μπορεί να εφαρμοστούν σε ένα ευρύ φάσμα εικόνων όπως ακτινογραφίες, υπέρηχοι, σπινθηρογραφήματα, θερμογραφικές εικόνες καθώς και εικόνες μικροσκοπίου.

Γενικότερα, οι τεχνικές επεξεργασίας ιατρικών εικόνων μπορούν να χωριστούν σε 3 κατηγορίες:

- α. *Τεχνικές υπολογισμού όγκου, σχήματος, μεγέθους* κτλ συγκεκριμένων οργάνων, συναρτήσει του χρόνου. Είσοδο αποτελεί μια ακολουθία εικόνων, ενώ σαν έξοδος λαμβάνεται μια ακολουθία αριθμητικών τιμών ή διανυσμάτων που αντιπροσωπεύουν το μετρούμενο μέγεθος
- β. *Απεικονιστικές τεχνικές*, στις οποίες είσοδο αποτελεί πάλι μια ακολουθία εικόνων, ενώ σαν έξοδος λαμβάνεται μια εικόνα στην οποία απεικονίζεται το μετρούμενο μέγεθος π.χ. η ταχύτητα του αίματος.
- γ. *Τεχνικές βελτίωσης εικόνας*. Η είσοδος αποτελείται από μια ακολουθία εικόνων, ενώ σαν έξοδος λαμβάνεται η ίδια ακολουθία με βελτιωμένα όμως συγκεκριμένα χαρακτηριστικά της όπως π.χ. χαμηλότερο θόρυβο, λιγότερη θόλωση κτλ

2.2 Τεχνικές εκτίμησης κίνησης

Ο άνθρωπος αλλά και όλα σχεδόν τα ζώα έχουν την ικανότητα να ξεχωρίζουν τα αντικείμενα, να διαπιστώνουν την κίνηση και να κινούνται στον τρισδιάστατο χώρο με χρήση της όρασής τους. Η ενσωμάτωση τέτοιων

δυνατοτήτων σε μηχανές είναι φαινομενικά κάτι το απλό, δεδομένης της ευρείας διαθεσιμότητας των μικροϋπολογιστών καθώς και ψηφιακών κινηματογραφικών μηχανών. Παρόλο που είναι εύκολο και σχετικά ανέξοδο να εξοπλιστεί ένας υπολογιστής με δυνατότητες όρασης, έχει αποδειχθεί εκπληκτικά δύσκολο να επιτευχθεί ικανότητα όρασης - ακόμη και πολύ περιορισμένης - σε μηχανές. Χωρίς ασφαλώς το τελευταίο να σημαίνει ότι δεν χρησιμοποιούνται διαφόρων ειδών συστήματα οπτικού ελέγχου και ανίχνευσης κίνησης σε πληθώρα εφαρμογών. Εντούτοις, η ευκολία με την οποία αντιλαμβάνεται ο άνθρωπος την κίνηση, σε αντιδιαστολή με τη δυσκολία του να δοθεί αυτή η ικανότητα σε μηχανές, έχει οδηγήσει στην εκπόνηση πολλών μελετών προκειμένου να γίνουν κατανοητές οι διαδικασίες της όρασης.

Οι τεχνικές εκτίμησης κίνησης μπορούν να διαχωριστούν σε τέσσερις κύριες κατηγορίες⁵⁷:

- Τεχνικές βαθμιδωτής μεταβολής (gradient)
- Τεχνικές επαναλαμβανόμενου pixel
- Τεχνικές block matching
- Τεχνικές στο πεδίο της συχνότητας

Οι τεχνικές βαθμιδωτής μεταβολής έχουν αναπτυχθεί για εφαρμογές ανάλυσης ακολουθιών εικόνων. Βρίσκουν την οπτική ροή και υπολογίζουν ένα πυκνό πεδίο διανυσμάτων κίνησης. Οι τεχνικές αυτές αναπτύσσονται αναλυτικά στο επόμενο κεφάλαιο.

Αμφότερες οι τεχνικές επαναλαμβανόμενου pixel και block matching έχουν αναπτυχθεί στο πλαίσιο της κωδικοποίησης ακολουθιών εικόνων⁵⁷. Όμως, οι πρώτες μπορούν να θεωρηθούν σαν υποσύνολο των τεχνικών βαθμιδωτής μεταβολής. Ακριβώς όμως λόγω της σημαντικότητάς τους, λαμβάνονται σαν ξεχωριστή κατηγορία. Οι τεχνικές block matching

βασίζονται στην ελαχιστοποίηση κάποιου μέτρου διαφοράς. Χρησιμοποιούνται ευρύτατα στις εφαρμογές κωδικοποίησης.

Τέλος, οι τεχνικές στο πεδίο της συχνότητας βασίζονται στις σχέσεις μεταξύ των μετασχηματισμών από το πεδίο του χρόνου, σε αυτό της συχνότητας. Στερούνται ευρείας χρήσης, ιδίως στο πεδίο της κωδικοποίησης ακολουθιών εικόνων.

Όλες οι τεχνικές βασίζονται στην υπόθεση ότι κάθε μεταβολή της φωτεινότητας ενός σημείου οφείλεται μόνο σε κάποιας μορφής κίνησης. Έτσι, εάν $\vec{r} = (x, y)^T$ είναι η θέση ενός pixel και τη χρονική στιγμή t έχει ένταση φωτεινότητας $g(\vec{r}, t)$ τότε η μετατόπισή του κατά το χρονικό διάστημα Δt θα είναι $\vec{d} = (dx, dy)^T$. Συνεπώς, η διαφορά για δύο διαδοχικά frames (Displaced Frame Difference) δίνεται από τη σχέση

$$DFD(\vec{r}, t, \vec{d}) = g(\vec{r}, t) - g(\vec{r} - \vec{d}, t - \Delta t) \quad (2.1)$$

Η επεξεργασία ακολουθιών εικόνων εμπλέκει επεξεργασία μεγάλου όγκου δεδομένων. Ευτυχώς, λόγω της συνεχώς αυξανόμενης υπολογιστικής ισχύος, είναι δυνατή η πραγματοποίηση αυτής της επεξεργασία εντός ανεκτών χρονικών διαστημάτων.

2.2.1 Τεχνικές επαναλαμβανόμενου pixel

Στις συγκεκριμένες τεχνικές⁵⁷, μέσω επαναληπτικών διαδικασιών, γίνεται προσπάθεια ελαχιστοποίησης του DFD στη σχέση 2.1. Η επανάληψη γίνεται συνήθως pixel-pixel και οδηγεί σε ένα πυκνό πεδίο διανυσμάτων κίνησης. Οι τεχνικές αυτές, είναι από τις πρώτες που αναπτύχθηκαν για εφαρμογές εκτίμησης κίνησης σε ακολουθίες εικόνων. Για το λόγο αυτό, είχαν σχεδιαστεί με γνώμονα το τότε διαθέσιμο hardware χαμηλής πολυπλοκότητας.

Η απόδοση τέτοιων τεχνικών εξαρτάται σημαντικά από τη μέθοδο διαδοχικού υπολογισμού του DFD. Επιπλέον, μεταθέτουν το αυξημένο υπολογιστικό κόστος στην πλευρά του αποκωδικοποιητή. Τέλος παρουσιάζουν δύο σημαντικά μειονεκτήματα. Πρώτον, η συνάρτηση σφάλματος - που προσπαθεί ο αλγόριθμος να ελαχιστοποιήσει - παρουσιάζει πολλά τοπικά ελάχιστα, με αποτέλεσμα συχνά να εγκλωβίζεται σε αυτά ο αλγόριθμος και να μη δίνει τη βέλτιστη λύση. Δεύτερον, δεν είναι δυνάτος ο επιτυχής χειρισμός μεγάλων μετατοπίσεων και ασυνέχειων.

2.2.2 Τεχνικές block matching

Στις μεθόδους που κάνουν χρήση της τεχνικής Block Matching⁵⁷, όλη η εικόνα χωρίζεται σε επιμέρους blocks. Όλα τα pixels που ανήκουν σε ένα block θεωρείται υφίστανται την ίδια μετατόπιση. Επομένως, στόχος τέτοιων τεχνικών είναι η εύρεση ενός διανύσματος $\vec{d}$ για κάθε block της εικόνας, το οποίο θα δείχνει τη μετατόπιση. Αρχικά η τεχνική αυτή εφαρμόστηκε για blocks του ενός pixel. Στη συνέχεια όμως, χρησιμοποιήθηκαν μεγαλύτερα blocks.

Για κάθε block του τρέχοντος frame, αναζητείται - μέσα σε μια κοντινή περιοχή του προηγούμενου frame - το αντίστοιχο block. Προφανώς επιλέγεται το block εκείνο που δίνει το μικρότερο σφάλμα για κάποια προκαθορισμένη συνάρτηση σφάλματος.

Οι συγκεκριμένες τεχνικές ελαχιστοποιούν απευθείας το DFD, με αποτέλεσμα να είναι πολύ αποδοτικές για κωδικοποίηση ακολουθιών εικόνων. Επιπλέον, ακριβώς λόγω της βασισμένης σε block φύσης τους, απαιτούν πολύ λίγες πληροφορίες σχετικά με τη γενική κίνηση. Έτσι, οι τεχνικές αυτές έχουν καταστεί ευρέως χρησιμοποιούμενες σε εφαρμογές, καθώς και από γνωστά standards όπως το MPEG-1,2 και το H.261.

2.2.3 Τεχνικές στο πεδίο της συχνότητας

Οι συγκεκριμένες τεχνικές στηρίζονται στη βασική ιδιότητα του μετασχηματισμού Fourier. Ότι δηλαδή, εάν $F(u,v)$ είναι ο μετασχηματισμός Fourier της $f(x,y)$, τότε για τη συνάρτηση $g(x,y) = f(x-\Delta x, y-\Delta y)$ ο αντίστοιχος μετασχηματισμός Fourier θα είναι $G(u,v) = F(u,v) e^{-j2\pi(u\Delta x + v\Delta y)}$. Επιπλέον, η διαφορά φάσης ανάμεσα σε δύο μετασχηματισμούς είναι

$$\Delta\phi(u,v) = \angle G(u,v) - \angle F(u,v) = -2\pi(u\Delta x + v\Delta y)$$

Έτσι, εάν υπολογισθεί η διαφορά φάσης $\Delta\phi$ για δύο ζεύγη (u,v) τότε μπορεί να βρεθούν και τα $\Delta x, \Delta y$.

2.3 Οπτική Ροή

Μια δημοφιλή μέθοδο προσδιορισμού ταχυτήτων από ακολουθίες εικόνων αποτελεί αυτή των διαφορών. Η μέθοδος των διαφορών, βασίζεται στην ιδέα ότι υπάρχει συσχέτιση, μεταξύ των χρονικών και των χωρικών διαφορών στην ένταση της φωτεινότητας της εικόνας. Σκοπό της μεθόδου αυτής αποτελεί ο υπολογισμός μιας προσέγγισης του δισδιάστατου πεδίου κίνησης. Προφανώς η δισδιάστατη αυτή κίνηση αποτελεί προβολή της τρισδιάστατης κίνησης των σημείων της μετρούμενης επιφάνειας πάνω στην επιφάνεια της (δισδιάστατης) εικόνας^{59,60}. Στην ιδανική περίπτωση, η προβολή του τρισδιάστατου πεδίου ταχύτητας στις δύο διαστάσεις της εικόνας θα μπορούσε να υπολογιστεί. Εντούτοις, κάτι τέτοιο δεν είναι απλώς δύσκολο να υπολογιστεί στην πράξη αλλά συνήθως είναι αδύνατον να επιτευχθεί τέλεια ακόμη και ο θεωρητικός υπολογισμός. Αυτή η δυσκολία εξηγείται από τους Horn και Schunk⁶¹ δίνοντας το παράδειγμα μιας περιστρεφόμενης σφαίρας χωρίς επιφανειακά σημάδια, η οποία υπό σταθερό φωτισμό, δεν προκαλεί διαφορές στη φωτεινότητα της εικόνας συναρτήσει του χρόνου παρόλο που υπάρχει κίνηση. Ένα άλλο παράδειγμα αποτελεί αυτό ενός στύλου που κινείται πίσω από ένα παράθυρο, μέσα από το οποίο φαίνεται μέρος μόνο του

στύλου. Η μόνη κίνηση που μπορεί να υπολογιστεί στην περίπτωση αυτή, είναι η κάθετη στο στύλο κίνηση.

2.3.1 Βασικές αρχές

Όπως αναφέρθηκε προηγουμένως, η μέθοδος της οπτικής ροής βασίζεται στην ιδέα ότι υπάρχει συσχέτιση μεταξύ των χρονικών και των χωρικών διαφορών. Έστω ότι $g(x,y,t)$ είναι η συνάρτηση της φωτεινότητας στο σημείο (x,y) τη χρονική στιγμή t . Για δύο διαδοχικά καρέ της ακολουθίας, τα οποία απέχουν χρονικά Δt , μπορεί να θεωρηθεί αντίστοιχα ότι και η οποιαδήποτε κίνηση θα είναι επίσης μικρή. Άρα προκύπτει η σχέση $g(x+\Delta x, y+\Delta y, t+\Delta t) = g(x,y,t)$. Αναπτύσσουμε σε σειρά Taylor την έκφραση του αριστερού μέρους - αγνοώντας τους όρους που είναι ανώτεροι της 1^η παραγώγου. Έτσι προκύπτει η σχέση

$$\begin{aligned} g(x,y,t) + \Delta x \frac{\partial g}{\partial x} + \Delta y \frac{\partial g}{\partial y} + \Delta t \frac{\partial g}{\partial t} &= g(x,y,t) \Leftrightarrow \frac{\partial g}{\partial x} \Delta x + \frac{\partial g}{\partial y} \Delta y + \frac{\partial g}{\partial t} \Delta t = 0 \Leftrightarrow \\ &\Leftrightarrow \frac{\partial g}{\partial x} V_x + \frac{\partial g}{\partial y} V_y + \frac{\partial g}{\partial t} = 0 \quad (2.2) \end{aligned}$$

όπου V_x και V_y είναι οι ταχύτητες ανάμεσα στις δύο διαδοχικές εικόνες προς την κατεύθυνση του άξονα x και y αντίστοιχα. Το σύνολο των διανυσμάτων V_x και V_y για μία εικόνα συνθέτει την "οπτική ροή" (optical flow) της εικόνας. Οι μερικές παράγωγοι που εμφανίζονται στην τελευταία εξίσωση μπορούν να υπολογιστούν από την ακολουθία των εικόνων. Έτσι για κάθε σημείο (x,y) προκύπτει μια εξίσωση με δύο αγνώστους V_x και V_y .

Η αοριστία αυτή, λόγω της ύπαρξης δύο αγνώστων σε μια εξίσωση, είναι γνωστή ως "aperture problem". Προκειμένου να λυθεί το συγκεκριμένο πρόβλημα, υιοθετούνται επιπρόσθετοι γενικοί περιορισμοί, οι οποίοι εξηγούνται στην επόμενη ενότητα.

2.3.1.1 Περιοριστικές παραδοχές

Προκειμένου να υπολογισθούν οι ταχύτητες από την εξίσωση 2.2, χρειάζεται να θεωρηθούν κατάλληλες περιοριστικές παραδοχές. Μερικές από τις πιο δημοφιλείς παραδοχές είναι:

- α. Η οπτική ροή είναι ομαλή και άρα γειτονικά σημεία έχουν παρεμφερείς ταχύτητες
- β. Η οπτική ροή είναι σταθερή για ένα μεγάλο τμήμα της εικόνας
- γ. Η οπτική ροή είναι αποτέλεσμα μιας περιορισμένης κίνησης, π.χ. επίπεδη κίνηση

Προφανώς, η παραδοχή (β) οδηγεί σε μια υβριδική κατάσταση ανάμεσα σε τεχνικές block matching και optical flow, αφού υπολογίζονται ταχύτητες για ένα block (και όχι για μεμονωμένα σημεία) αλλά με τεχνικές optical flow.

2.3.1.2 Περιοριστική παραδοχή Horn and Schunck

Η πιο προσφιλής παραδοχή είναι αυτή των Horn⁶¹ and Schunck^{61,62}. Δεχόμενοι ότι οι τιμές της κίνησης στα περισσότερα σημεία της εικόνας διακυμαίνονται ομαλά, μια περιοριστική παραδοχή είναι η ελαχιστοποίηση του σφάλματος που δίνεται από την ακόλουθη έκφραση :

$$E^2(x, y) = (\text{σφάλμα ανωτέρων όρων από την Taylor}) + \lambda^2 =$$

$$= \left(\frac{\partial g}{\partial x} V_x + \frac{\partial g}{\partial y} V_y + \frac{\partial g}{\partial t} \right)^2 + \lambda^2 \left[\left(\frac{\partial}{\partial x} V_x \right)^2 + \left(\frac{\partial}{\partial y} V_x \right)^2 + \left(\frac{\partial}{\partial x} V_y \right)^2 + \left(\frac{\partial}{\partial y} V_y \right)^2 \right]$$

όπου λ είναι μια σταθερά. Έτσι, αναζητούνται οι τιμές των V_x και V_y που ελαχιστοποιούν το R στην ακόλουθη έκφραση

$$R = \iint \left\{ \left(\frac{\partial g}{\partial x} V_x + \frac{\partial g}{\partial y} V_y + \frac{\partial g}{\partial t} \right)^2 + \lambda^2 \left[\left(\frac{\partial}{\partial x} V_x \right)^2 + \left(\frac{\partial}{\partial y} V_x \right)^2 + \left(\frac{\partial}{\partial x} V_y \right)^2 + \left(\frac{\partial}{\partial y} V_y \right)^2 \right] \right\} dx dy$$

Για να υπολογισθούν οι ζητούμενες τιμές των V_x και V_y , παραγωγίζεται η παραπάνω σχέση και τίθεται $\frac{\partial R}{\partial V_x} = \frac{\partial R}{\partial V_y} = 0$.

Θέτοντας

$$\left(\frac{\partial}{\partial x} V_x\right)^2 + \left(\frac{\partial}{\partial y} V_x\right)^2 = u - u_{ave}$$

$$\left(\frac{\partial}{\partial x} V_y\right)^2 + \left(\frac{\partial}{\partial y} V_y\right)^2 = v - v_{ave}$$

$$P = \left(\frac{\partial g}{\partial x} u_{ave} + \frac{\partial g}{\partial y} v_{ave} + \frac{\partial g}{\partial t}\right)^2$$

προκύπτουν οι τελικές εκφράσεις για τα u και v

$$D = \lambda^2 + \left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2$$

$$u = u_{ave} - \frac{\partial g}{\partial x} \frac{P}{D}$$

$$v = v_{ave} - \frac{\partial g}{\partial y} \frac{P}{D}$$

Οι παραπάνω σχέσεις λύνονται με τη μέθοδο Gauss-Seidel, με διαδοχικές δηλαδή επαναλήψεις μέχρι να βρεθεί κάποια τιμή που να ικανοποιεί κάποιο ελάχιστο όριο σφάλματος.

Εντούτοις, υπάρχουν δύο βασικά προβλήματα στην παραπάνω παραδοχή⁶³. Πρώτον, η βασική σχέση 2.2 δίνει "καλύτερα" αποτελέσματα για τις περιοχές της εικόνας στις οποίες υπάρχει μεγάλη μεταβολή της φωτεινότητας ως προς x ή/ και y . Αυτό γίνεται καλύτερα κατανοητό εάν στην 2.2 αντικατασταθούν τα V_x και V_y με το διάνυσμα $\vec{u} = (V_x, V_y)^T$ οπότε προκύπτει

η σχέση $\vec{u} = -\frac{\nabla g}{\nabla^2 g}$, όπου $\nabla g = (g_x, g_y)$. Για πολύ μικρές τιμές του ∇g

ουσιαστικά το $\vec{u}$ απειρίζεται. Το άλλο πρόβλημα είναι το ότι η οπτική ροή θα ομαλοποιηθεί στα σημεία ασυνέχειας, οδηγώντας σε ανακριβείς εκτιμήσεις της ροής.

2.3.2 Υπολογισμός διαφορών

Για την εκτίμηση της κίνησης, πρέπει να υπολογιστούν από την ακολουθία των εικόνων οι διάφορες μεταβολές, ως προς το χρόνο και ως προς τις κατευθύνσεις x και y . Οι μεταβολές αυτές είναι οι μερικές παράγωγοι της έντασης της φωτεινότητας για κάθε pixel ως προς t , x και y . Για την συνάρτηση της φωτεινότητας $g(x,y,t)$ δεν υπάρχει κάποια αναλυτική έκφραση για την τιμή της παρά μόνον τιμές της για διάφορες τιμές των (x,y,t) . Επομένως, για τον υπολογισμό των παραγώγων θα γίνει χρήση των διακριτών παραγώγων.

2.3.2.1 Υπολογισμός χωρικών διαφορών

Η διακριτή παράγωγος ενός σημείου ως προς x μπορεί να οριστεί με διαφόρους τρόπους. Έτσι για μια συνάρτηση $f(x)$ ισχύει:

$$\frac{df}{dx} = f(x) - f(x-1) \text{ , αριστερή παράγωγος}$$

$$\frac{df}{dx} = f(x+1) - f(x) \text{ , δεξιά παράγωγος}$$

$$\frac{df}{dx} = \frac{f(x+1) - f(x-1)}{2} \text{ , κεντρική παράγωγος}$$

Αντίστοιχα ισχύουν και για συναρτήσεις περισσότερων της μίας μεταβλητών. Στην πράξη, για τον υπολογισμό των διακριτών παραγώγων γίνεται χρήση της συνέλιξης. Έτσι, υπολογίζεται για κάθε σημείο μια ομαλοποιημένη τιμή της κεντρικής παραγώγου του. Με τον όρο ομαλοποιημένη, εννοείται ότι λαμβάνεται ο μέσος όρος των κεντρικών παραγώγων για μια μικρή περιοχή γύρω από το σημείο ενδιαφέροντος. Συνεπώς, ο υπολογισμός της χωρικής παραγώγου ανάγεται στην εφαρμογή μιας μάσκας συνέλιξης επάνω στην εικόνα που μελετάται. Υπάρχουν διάφορων τύπων τέτοιες μάσκες και για τις δύο κατευθύνσεις x και y .

- Roberts $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ (κατεύθυνση x) $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (κατεύθυνση y)

- Prewitt $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ (κατεύθυνση x) $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ (κατεύθυνση y)

- Sobel $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ (κατεύθυνση x) $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ (κατεύθυνση y)

Προφανώς για να προκύψει η τελική τιμή θα πρέπει το αποτέλεσμα της συνέλιξης να διαιρεθεί με αριθμό ανάλογο της περιοχής γύρω από το σημείο ενδιαφέροντος. Έτσι, για την περίπτωση της μάσκας Prewitt -της οποίας θα γίνει χρήση - η τιμή του αριθμού αυτού είναι το 6. Η τιμή προκύπτει βάσει του σκεπτικού ότι για κάθε σημείο λαμβάνεται η κεντρική παράγωγος άρα πρέπει να διαιρεθεί δια 2 ενώ συνολικά λαμβάνονται υπόψη 3 παράγωγοι.

2.3.2.2 Υπολογισμός χρονικών διαφορών

Ο υπολογισμός των χρονικών διαφορών είναι πολύ πιο απλός από αυτόν των χωρικών. Η διακριτή παράγωγος της συνάρτησης της φωτεινότητας $g(x,y,t)$ ως προς το χρόνο t δεν είναι άλλο από τη διαφορά $g(x,y,t+dt) - g(x,y,t)$, όπου dt ο χρόνος ανάμεσα σε δύο διαδοχικά frames της εν λόγω ακολουθίας. Για την ομαλοποίηση όμως των τιμών αυτών, είναι δυνατόν να εφαρμοσθεί μια συνελικτική μάσκα που να δίνει απλώς το μέσο όρο μιας περιοχής 9 σημείων γύρω από την περιοχή ενδιαφέροντος. Μια τέτοια μάσκα

είναι $\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

2.3.3 Υπολογισμός optical flow

Έχοντας υπολογιστεί οι παράγωγοι $\frac{\partial g}{\partial x}$, $\frac{\partial g}{\partial y}$ και $\frac{\partial g}{\partial t}$ - σύμφωνα με τα παραπάνω - είναι πια δυνατό να υπολογιστούν τα P και D της εξίσωσης 3.2. Αφού υπολογισθούν και τα P και D, αντικαθίστανται όλες οι αριθμητικές τιμές στην 3.2 και για κάθε επανάληψη προκύπτει και μια νέα τιμή για τα u και v. Για τον υπολογισμό των u_{ave} και v_{ave} , λαμβάνεται υπόψη μόνο μια μικρή περιοχή γύρω από το σημείο του ενδιαφέροντος. Δε λαμβάνεται υπόψη όλη η περιοχή στον υπολογισμό του μέσου όρου, διότι κάτι τέτοιο θα δημιουργούσε ουσιαστικά μια σταθερή συνιστώσα κίνησης για όλα τα σημεία που ενδιαφέρουν με μικρές μόνο μεταβολές από σημείο σε σημείο. Με την παραπάνω διαδικασία, εφόσον αυτή εφαρμοσθεί στο σύνολο των σημείων που ενδιαφέρουν, προκύπτουν τιμές των u και v για όλα τα σημεία.

2.3.4 Επιλογή σημείων

Η μέθοδος υπολογισμού της οπτικής ροής εφαρμόζεται ανάμεσα σε διαδοχικά frames. Δηλαδή, εν αντιθέσει με τις block matching διαδικασίες δεν υπάρχουν key frames. Επομένως, για κάθε ζεύγος διαδοχικών frames θα πρέπει να επαναπροσδιορίζονται τα σημεία ενδιαφέροντος. Ο επαναπροσδιορισμός μπορεί να γίνει αυτόματα ή χειροκίνητα. Ενδιαφέρον παρουσιάζει, προφανώς, ο αυτόματος επαναπροσδιορισμός. Έτσι, αφού υπολογιστεί η ταχύτητα για κάθε σημείο ανάμεσα σε δύο frames, υπολογίζεται η νέα θέση που έχει το σημείο στο δεύτερο frame. Αυτό γίνεται προσθέτοντας στις συντεταγμένες που είχε το σημείο στο πρώτο frame, τη μετατόπιση που υπολογίστηκε βάσει της μεθόδου οπτικής ροής. Η όλη διαδικασία επαναλαμβάνεται για όλα τα ζεύγη διαδοχικών frame και προφανώς για όλα τα σημεία.

2.4 Υλοποίηση αλγορίθμου - Εφαρμογή

Όλα τα παραπάνω, εφαρμόστηκαν για την υλοποίηση ενός αλγορίθμου υπολογισμού της οπτικής ροής σε ακολουθίες εικόνων. Για την εκτέλεση του αλγορίθμου χρειάζεται απλώς να οριστεί η περιοχή ενδιαφέροντος ROI (Region Of Interest). Ακολούθως, ο αλγόριθμος εφαρμόζεται επαναληπτικά για το σύνολο των εικόνων δίνοντας σαν έξοδο για κάθε ζεύγος εικόνων την οπτική ροή κατά τους άξονες x και y . Στη συνέχεια θα εξετασθεί η εφαρμογή βήμα-βήμα.

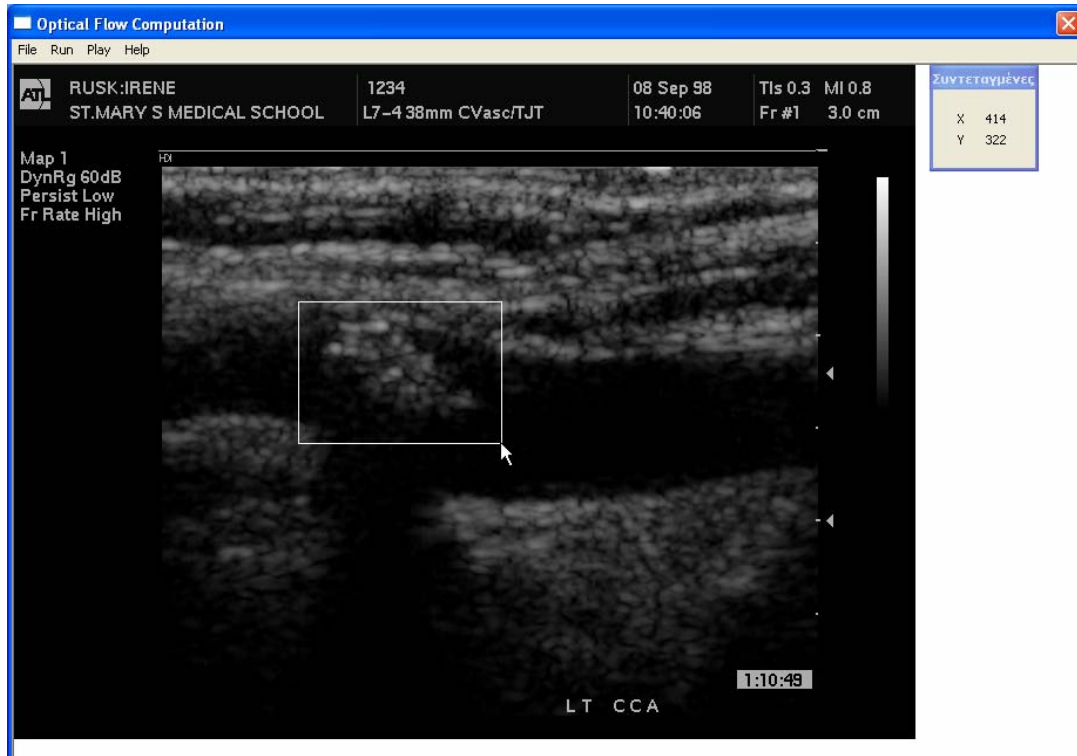
2.4.1 Επιλογή ROI

Σε πρώτη φάση, καλείται ο χρήστης να φορτώσει την πρώτη από την ακολουθία των εικόνων. Αυτό γίνεται μέσω της επιλογής FileOpen→Image. Αφού φορτωθεί η πρώτη εικόνα, ενημερώνεται ο χρήστης για το όνομα της αρχικής καθώς και της τελικής εικόνας της ακολουθίας. Να σημειωθεί ότι αναμένεται τα ονόματα των εικόνων της ακολουθίας να είναι της μορφής filename#.tif, όπου filename οποιοδήποτε όνομα αρχείου και # αριθμός που χαρακτηρίζει τη θέση του αρχείου μέσα στην ακολουθία.

Στη συνέχεια, μέσω της επιλογής Run→Select ROI, ο χρήστης με τη βοήθεια του mouse επιλέγει την περιοχή ενδιαφέροντος. Ουσιαστικά, κρατώντας πατημένο το αριστερό πλήκτρο του mouse επιλέγει μία ορθογώνια περιοχή πάνω στην εικόνα. Για να τελειώσει η διαδικασία επιλογής ROI ο χρήστης επιλέγει Run→Compute, οπότε εκτελείται ο αλγόριθμος υπολογισμού της οπτικής ροής για τις δύο κατευθύνσεις κίνησης x και y , για όλα τα pixels που περιλαμβάνονται από τα όρια της ROI.

Καθ' όλη τη διάρκεια της διαδικασίας επιλογής ROI εμφανίζεται ένα παράθυρο συντεταγμένων, στο οποίο απεικονίζονται οι συντεταγμένες του σημείου το οποίο "δείχνει" ο cursor του mouse. Ενώ μετά την επιλογή της

ROI, εμφανίζεται ένα επιπλέον παράθυρο που ενημερώνει το χρήστη για τις συντεταγμένες της επιλεγμένης ROI.

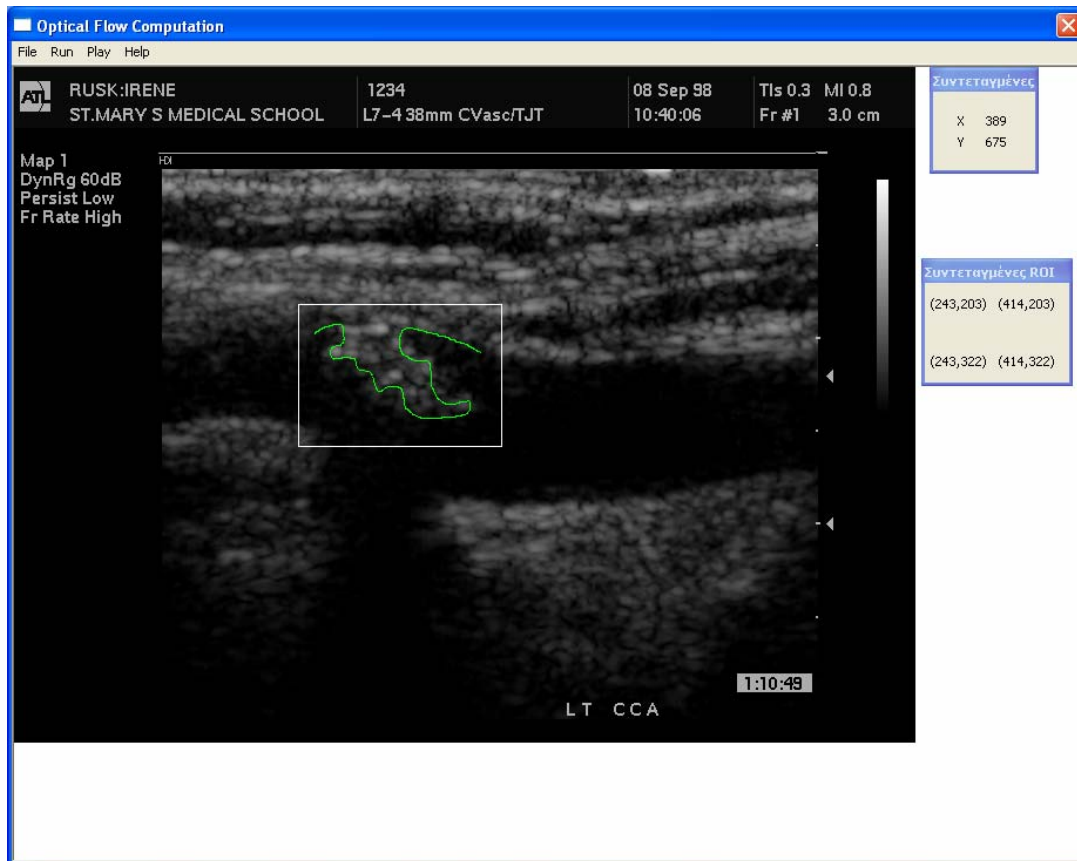


Εικόνα 1 Επιλογή ROI

2.4.2 Επιλογή σημείων ενδιαφέροντος

Επειδή μας ενδιαφέρει η κίνηση συγκεκριμένων σημείων και όχι ολόκληρης της ROI, δίνεται η δυνατότητα στον χρήστη να επιλέξει τα σημεία που τον ενδιαφέρουν - δηλαδή τα όρια της αθηρωματικής πλάκας που μελετάται. Επιλέγοντας Run→Mark Limits, εμφανίζεται στην οθόνη μια μεγεθυμένη απεικόνιση της ROI. Επάνω στην εικόνα αυτή, μπορεί ο χρήστης με τη βοήθεια του mouse να χρωματίσει τα σημεία που τον ενδιαφέρουν, απλώς κρατώντας πατημένο το αριστερό πλήκτρο του mouse και "ζωγραφίζοντας" free hand. Για να ολοκληρωθεί η επιλογή των σημείων, πρέπει να επιλεγεί Run→DONE Marking Limits.

Και σε αυτή την επιλογή, το παράθυρο των συντεταγμένων είναι ενεργό για να μπορεί ο χρήστης να γνωρίζει τις ακριβείς συντεταγμένες των σημείων που επιλέγει.

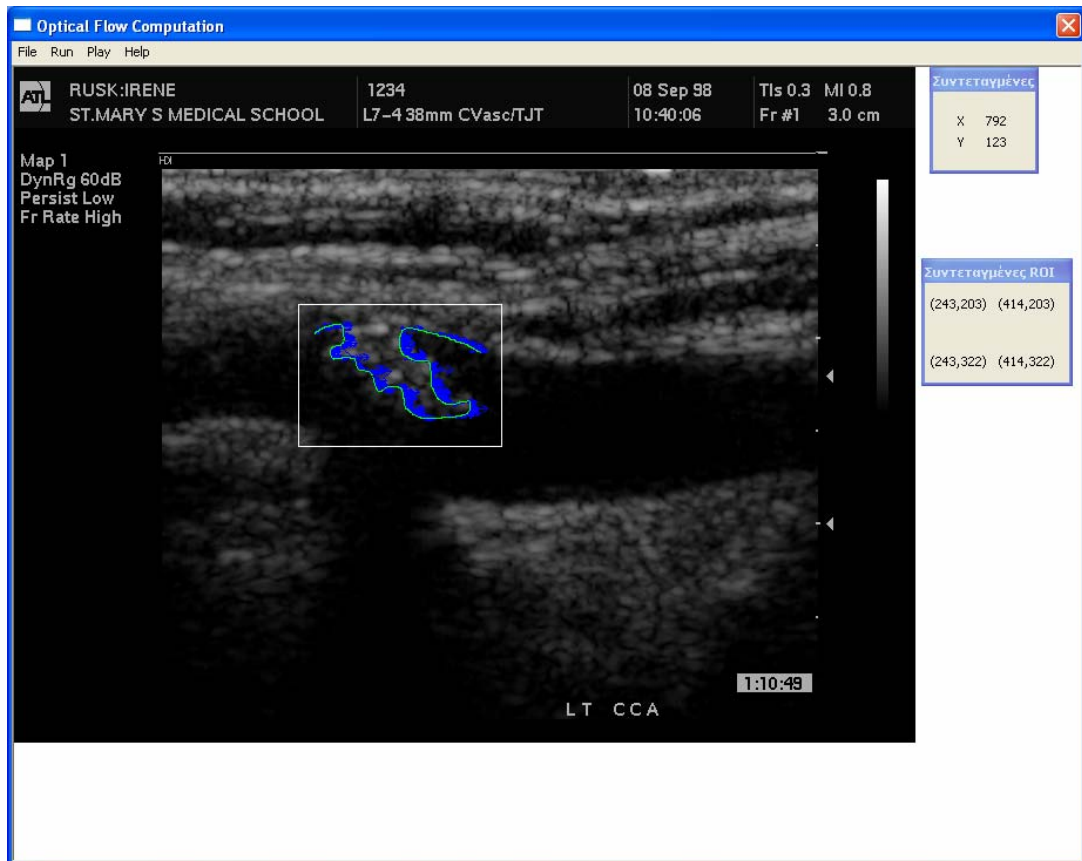


Εικόνα 2 - Επιλογή σημείων ενδιαφέροντος

2.4.3 Εμφάνιση πληροφοριών για τα σημεία ενδιαφέροντος

Αφού ολοκληρωθούν τα παραπάνω βήματα, μπορούν να εμφανιστούν στην οθόνη οπτικοποιημένες διάφορες πληροφορίες για τα σημεία ενδιαφέροντος. Έτσι με την επιλογή Run→Show Arrows→Vx εμφανίζονται πάνω στην εικόνα οι μέγιστες ταχύτητες - στην κατεύθυνση του άξονα x - που αναπτύχθηκαν καθ' όλη τη διάρκεια της ακολουθίας για κάθε pixel του σημείου ενδιαφέροντος. Αντίστοιχα, με τις Run→Show Arrows→Vy/Both μπορούμε να δούμε οπτικοποιημένες τις μέγιστες ταχύτητες στην κατεύθυνση του

άξονα γ ή και τις δύο ταχύτητες. Η οπτική αναπαράσταση γίνεται με χρήση βελών, τα οποία έχουν μέγεθος ανάλογο με το μέτρο της ταχύτητας και φορά ανάλογα με την φορά κίνησης.



Εικόνα 3 - Εμφάνιση πληροφοριών για τα σημεία ενδιαφέροντος

Επιπλέον με την επιλογή Run→Max Differences, εμφανίζονται πάνω στην οθόνη - πάλι με χρήση βελών - οι μέγιστες διαφορές ταχύτητας που αναπτύχθηκαν σε γειτονικά pixels. Δηλαδή, απεικονίζεται από ένα βέλος σε κάθε ζεύγος σημείων, τα οποία απέχουν μικρή απόσταση και για τα οποία σε κάποια χρονική στιγμή η κίνησή τους ήταν αντίθετης φοράς και η διαφορά των μέτρων τους μέγιστη.

Τέλος, παρέχεται η επιλογή Play→PlaySequence, με την οποία αναπαράγεται όλη η ακολουθία των εικόνων, υπό μορφή video. Εφόσον έχουν

επιλεγεί σημεία ενδιαφέροντος, τότε σε κάθε frame εμφανίζεται η θέση που προκύπτει για κάθε σημείο ενδιαφέροντος βάσει των αποτελεσμάτων της μεθόδου της οπτικής ροής. Δηλαδή, η αναπαράσταση αυτή αποτελεί και έναν τρόπο ελέγχου των αποτελεσμάτων της μεθόδου. Εφόσον η μέθοδος λειτουργεί σωστά, θα πρέπει τα σημεία ενδιαφέροντος να ακολουθούν τη γενικότερη κίνηση της εικόνας.

2.4.4 Αποθήκευση πληροφοριών για τα σημεία ενδιαφέροντος

Από την επιλογή File→Save Results δίνεται η δυνατότητα στον χρήστη να αποθηκεύσει όλες τις πληροφορίες που υπολογίστηκαν, σε μορφή αρχείου text. Στο αρχείο αποθηκεύονται οι συντεταγμένες της ROI καθώς και πληροφορίες για τη μέγιστη απόλυτη ταχύτητα όλων των σημείων ενδιαφέροντος καθώς και για την μέγιστη σχετική ταχύτητα όλων των σημείων ενδιαφέροντος. Επιπλέον, για κάθε σημείο ενδιαφέροντος δίνεται η μέγιστη ταχύτητα που αναπτύχθηκε και προς τους δύο άξονες κίνησης. Τέλος, αποθηκεύονται οι ταχύτητες κάθε σημείου ενδιαφέροντος για κάθε ζεύγος διαδοχικών frame καθώς και η νέα θέση στο δεύτερο frame του ζεύγους.

2.4.5 Αποθήκευση/ανάκτηση επιλογών ROI και σημείων ενδιαφέροντος

Προκειμένου να μπορεί ο χρήστης να ξανατρέξει το πρόγραμμα με ακριβώς τις ίδιες επιλογές υπάρχει το ζευγάρι εντολών File→Export Settings και File→Import Settings. Με την πρώτη, αποθηκεύονται σε εξωτερικό αρχείο όλες οι επιλογές σχετικά με την ακολουθία εικόνων, δηλαδή οι συντεταγμένες της ROI καθώς και όλων των σημείων ενδιαφέροντος. Αντίστροφα, με τη δεύτερη εντολή εισάγονται στο πρόγραμμα από αρχείο οι παραπάνω συντεταγμένες. Πρέπει να τονιστεί ότι εισάγονται και εξάγονται μόνο οι συντεταγμένες των σημείων στην αρχική εικόνα της

ακολουθίας και όχι πληροφορίες που προκύπτουν από την επεξεργασία. Επομένως, μετά την εισαγωγή εκτελείται εκ νέου η διαδικασία υπολογισμού της οπτικής ροής.

Κεφάλαιο 3^ο

ΑΣΘΕΝΕΙΣ - ΑΠΟΤΕΛΕΣΜΑΤΑ

3.1 Ασθενείς και καταγραφή εικόνων

Για την αξιολόγηση της αναπτυχθείσας μεθόδου χρησιμοποιήθηκαν 19 ακολουθίες εικόνων, οι οποίες ελήφθησαν από 17 ασθενείς που προσέφυγαν για υπερηχογράφημα των αρτηριών του λαιμού στο εργαστήριο Irvine του νοσοκομείου St Mary's στο Λονδίνο⁷⁴. Για δύο ασθενείς, εξετάσθηκαν και οι δύο αρτηρίες. Στην πλειονότητά τους, οι υπό εξέταση πλάκες βρίσκονταν στο εσωτερικό της αρτηρίας της καρωτίδας, πολύ κοντά ή ακριβώς στο σημείο του καρωτιδικού βολβού.

Η επιλογή των ασθενών έγινε με κριτήρια όπως η παραγόμενη εικόνα υπερήχων της πλάκας και ο βαθμός της στένωσης λόγω της ύπαρξης της πλάκας. Η διάμετρος της στένωσης στο σημείο των πλακών ήταν μεγαλύτερη από 50% (διακύμανση 50% - 90%, μέση τιμή 67,9%), έτσι ώστε οι πλάκες να προεξέχουν στη διατομή. Ο βαθμός της στένωσης, βάσει της μέτρησης της ταχύτητας στο σημείο των πλακών, υπολογίστηκε από τον χειριστή του οργάνου την στιγμή της εξέτασης. Το κριτήριο της κατ' ελάχιστο 50% εσωτερικής στένωσης της αρτηρίας προσδιορίστηκε σαν μέγιστη τιμή συστολικής ταχύτητας τουλάχιστον 125cm/sec.

Για κάθε ασθενή, καταγράφηκε μια ακολουθία εικόνων κατά την εγκάρσια κατεύθυνση. Χρησιμοποιήθηκε ένας υπερηχογράφος ATL (Advanced Technology Laboratory) Ultramark 4 Duplex scanner με έναν γραμμικής διάταξης μετατροπέα υψηλής ανάλυσης στα 7.5MHz. Οι ρυθμίσεις του οργάνου ορίστηκαν στην αρχή της εξέτασης και δεν τροποποιήθηκαν κατά τη διάρκεια της διαδικασίας (dynamic range: 60dB, persistence: low, 2D gray map: linear, and frame rate: high). Οι ακολουθίες των εικόνων ελήφθησαν

με ρυθμό ανανέωσης 25 frames/sec για 3 δευτερόλεπτα και επομένως περιλαμβάνουν 2-3 καρδιακούς κύκλους. Οι ασθενείς κρατούσαν την αναπνοή τους κατά τη διάρκεια της καταγραφής των εικόνων. Στη συνέχεια, οι εικόνες αποθηκεύτηκαν σε μαγνητικό οπτικό δίσκο και κατόπιν σε compact disk προκειμένου να μεταφερθούν σε ηλεκτρονικό υπολογιστή για περαιτέρω ανάλυση.

Ένα σύνολο 10 ακολουθιών προέρχεται από ασυμπτωματικούς ασθενείς και οι υπόλοιπες 9 από συμπτωματικούς. Στον πίνακα 1 περιγράφονται τα χαρακτηριστικά των ασθενών από τους οποίους έχουν ληφθεί οι εικόνες. Δεν υπάρχει στατιστικά σημαντική διαφορά μεταξύ των ηλικιών ($p=0.848$, t-test) και των βαθμών στένωσης ($p=0.929$, t-test) των δύο κατηγοριών ασθενών.

3.2 Αποτελέσματα

Για κάθε μία από τις παραπάνω ακολουθίες εκτελέστηκε το πρόγραμμα OFC με συγκεκριμένες επιλογές (settings) όσον αφορά στη ROI και στα σημεία ενδιαφέροντος. Από κάθε ακολουθία προέκυψε σαν έξοδος ένα αρχείο text στο οποίο περιέχεται η ταχύτητα κάθε σημείου ενδιαφέροντος για κάθε ζεύγος διαδοχικών frames καθώς και η νέα θέση στην οποία μετατοπίζεται μετά από κάθε κίνηση. Από όλες τις πληροφορίες, επιλέχθηκαν για αναπαράσταση σε πίνακα αυτές που αφορούν σε 3 μεγέθη:

- Στη μέγιστη κίνηση που αναπτύσσει κάποιο σημείο κατά τη διάρκεια της ακολουθίας.
- Στη μέγιστη σχετική διαφορά κίνησης ανάμεσα σε 2 κοντινά σημεία στο ίδιο frame - δηλαδή την ίδια χρονική στιγμή
- Στη μέγιστη διαφορά ταχύτητας ενός σημείου ενδιαφέροντος ανάμεσα σε 2 διαδοχικά frames - δηλαδή στη μεταβολή της ταχύτητας.

Τα 3 μεγέθη παρέχουν χαρακτηριστικές ποσοτικές πληροφορίες για την κίνηση της πλάκας. Ιδιαίτερα τα δύο τελευταία μεγέθη χαρακτηρίζουν τις

τάσεις που αναπτύσσονται ανάμεσα σε γειτονικά σημεία στο ίδιο frame και τις τάσεις που αναπτύσσονται στο ίδιο σημείο ενδιαφέροντος ανάμεσα σε 2 διαδοχικά frames, αντίστοιχα. Επιπλέον, μεγέθη όπως η μέγιστη ταχύτητα της επιφάνειας και η μέγιστη μεταβολή της ταχύτητας ενός σημείου μετρώνται και σε άλλες μελέτες⁴⁹ δίνοντας έτσι τη δυνατότητα να γίνουν συγκρίσεις αποτελεσμάτων

Από την εκτέλεση του προγράμματος προέκυψαν οι ακόλουθες πληροφορίες, οι οποίες παρουσιάζονται στον πίνακα 2. Όλες οι ταχύτητες είναι σε mm/sec ενώ ως θετική κίνηση στον άξονα x λαμβάνεται η κίνηση προς τα δεξιά και στον άξονα y η κίνηση προς τα κάτω.

Επιπλέον, παρουσιάζεται και η τιμή median grayscale, η οποία έχει ευρέως χρησιμοποιηθεί για την εκτίμηση της υφής της αθηρωματικής πλάκας στην καρωτίδα⁷⁴. Τέλος, εξετάζεται κατά πόσο υπάρχει στατιστικά σημαντική διαφορά μεταξύ των δύο κατηγοριών (t-test).

Πρέπει να σημειωθεί ότι η ακρίβεια ενός δεκαδικού ψηφίου που υπάρχει στον πίνακα 2 προκύπτει από τη μετατροπή των μονάδων μέτρησης. Πιο συγκεκριμένα, η ακρίβεια με την οποία γίνονται όλοι οι υπολογισμοί είναι της τάξης των 15 ψηφίων και τα αποτελέσματα των ταχυτήτων προκύπτουν αρχικά σε μονάδες pixels/frame. Στη συνέχεια πολλαπλασιάζονται με τον συντελεστή 25/178 που αντιστοιχεί στα 25 frames/sec frame rate της ακολουθίας και στα 178 pixels που αντιστοιχούν σε κάθε cm.

Πίνακας 1 - Οι 19 ακολουθίες εικόνων καθώς και τα χαρακτηριστικά τους που χρησιμοποιήθηκαν για την αξιολόγηση της αναπτυχθείσας μεθόδου.

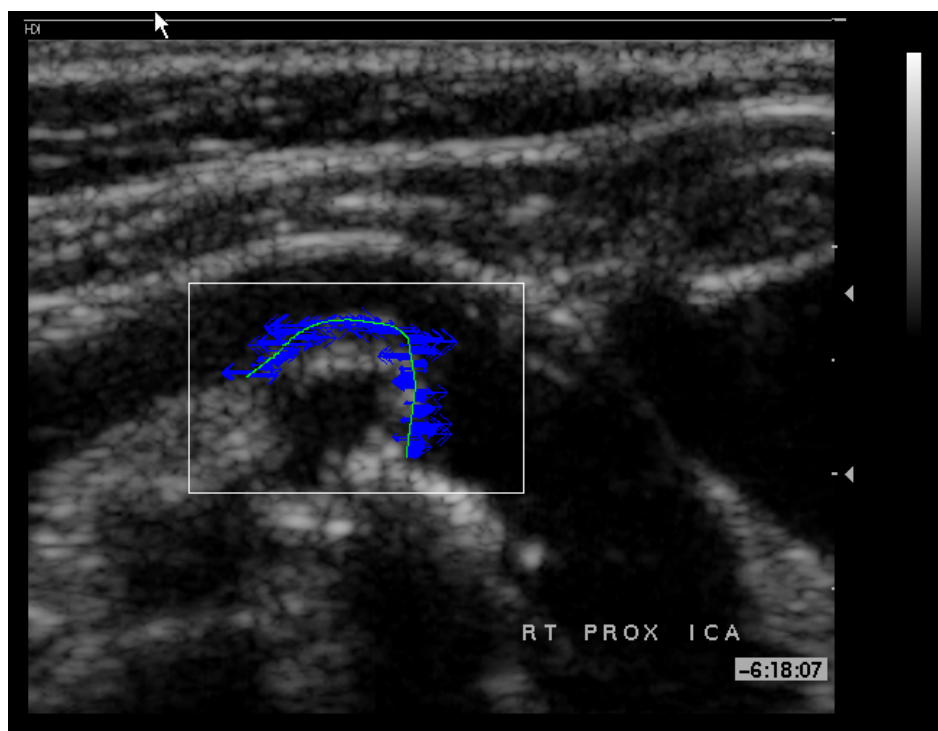
A/A	Ηλικία (έτη)	Βαθμός στένωσης (%)
<i>Ασυμπτωματικές πλάκες</i>		
A1	66	50
A2	66	65
A3	62	75
A4	70	55
A5	90	70
A6	63	90
A7	72	80
A8	59	80
A9	52	50
Μέση τιμή	<i>66.7 ± 10.6</i>	<i>68.3 ± 14.4</i>
<i>Συμπτωματικές πλάκες</i>		
S1	63	70
S2	71	90
S3	63	70
S4	50	90
S5	85	55
S6	74	55
S7	67	50
S8	74	50
S9	58	60
S10	73	85
Μέση τιμή	<i>67.8 ± 9.8</i>	<i>67.5 ± 16.0</i>
T-test		
p-value	0.848	0.929

Πίνακας 2 - Οι μέγιστες ταχύτητες - απόλυτες και σχετικές - καθώς και η μέγιστη διαφορά ταχύτητας για τις 19 ακολουθίες εικόνων.

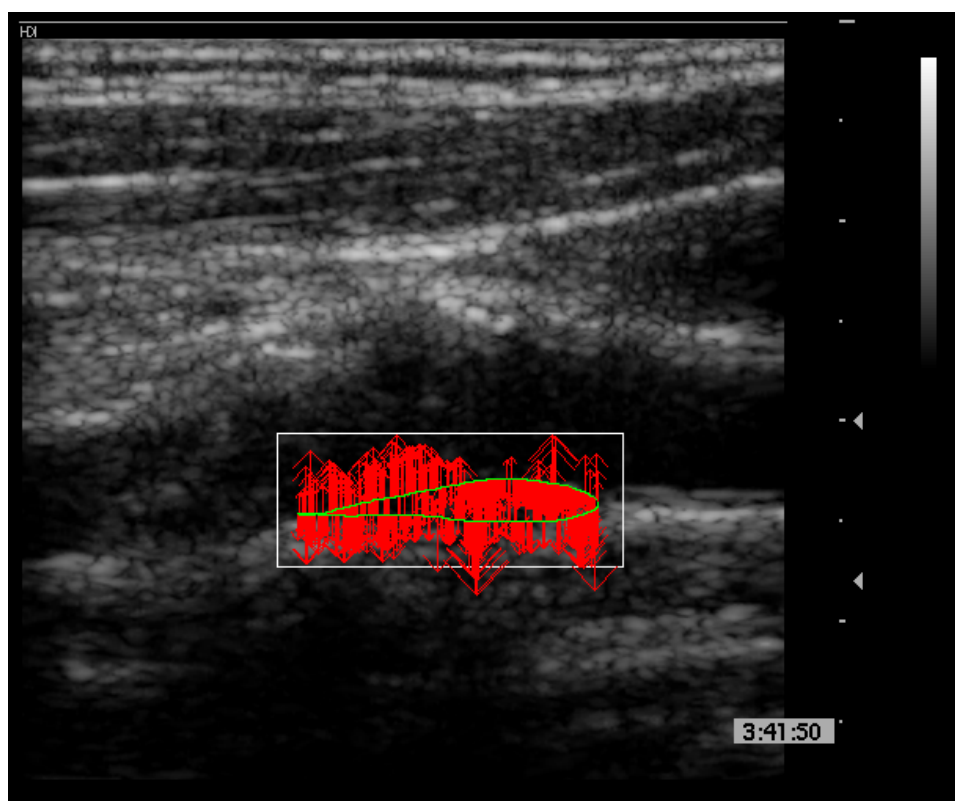
A/A	Μέγιστη ταχύτητα κίνησης (mm/sec)			Μέγιστη σχετική ταχύτητα (mm/sec)			Μέγιστη διαφορά ταχύτητας για το ίδιο σημείο (mm/sec)			Median grayscale
	κατά x	κατά y	συνολική	κατά x	κατά y	συνολική	κατά x	κατά y	συνολική	
A1	9.3	-8.7	12.7	7.9	9	12.0	10.1	9.5	13.9	14
A2	9.4	8.2	12.5	10.2	9.8	14.1	11.6	9.5	15.0	7
A3	-9.7	-8.4	12.8	10.3	7.3	12.6	10.7	9.8	14.5	6
A4	-3.2	2.7	4.2	6.9	4.7	8.3	3.5	3.4	4.9	23
A5	6.7	7.2	9.8	10	11.5	15.2	8	8.3	11.5	39
A6	-12.6	4.6	13.4	7.9	5.4	9.6	12.3	5.8	13.6	25
	-13.8	-12.8	18.8	7.9	8.2	11.4	14.4	12.9	19.3	
A7	-9.9	6.4	11.8	7.9	7.9	11.2	9.7	8.3	12.8	20
A8	13.3	-12.3	18.1	12	13.4	18.0	16.4	15.7	22.7	18
A9	-12.8	-13.9	18.9	7.8	9.1	12.0	14.8	18.1	23.4	47
Μέση τιμή		13.3±4.5		12.4±2.8			15.2±5.5			22.1 ±13.6
S1	5.3	4.5	7.0	7.9	9	12.0	4.8	4.5	6.6	22
S2	-5.6	-9.6	11.1	6.9	6.6	9.5	7.22	10.8	13.0	18
S3	-10.8	9.1	14.1	7.9	9	12.0	11.9	8.6	14.7	23
S4	7.9	8.8	11.8	6.9	8.9	11.3	9.9	9.6	13.8	20
S5	10.8	-8.7	13.9	8.9	7.3	11.5	12	8.9	14.9	24
S6	-11.6	12.8	17.3	10	11.8	15.5	12.7	12.7	18.0	18
S7	11	9.4	14.5	6.3	7.6	9.9	11.1	11.4	15.9	55
S8	8.13	10.1	13.0	7.87	10.2	12.9	9.31	11.7	15.0	28
S9	-9.3	-7.9	12.2	9	8.9	12.7	9	8.8	12.6	40
S10	-14.5	10.8	18.1	12.3	12.3	17.4	14.8	11.8	18.9	25
Μέση τιμή		13.3±3.2		12.5±2.4			14.3±3.4			27.3±11.6
T-test										
p-Value		0.871		0.865			0.751			

Τα παραπάνω δεδομένα προκύπτουν από τις μέγιστες τιμές που υπολογίζονται μετά από κάθε εκτέλεση της εφαρμογής - για συγκεκριμένη ακολουθία εικόνων. Οι τιμές της κάθε ακολουθίας προέκυψαν για συγκεκριμένη επιλογή σημείων ενδιαφέροντος, πάχους κατ' ελάχιστη τιμή 1 pixel. Έτσι, για ελαφρώς τροποποιημένη επιλογή σημείων ενδιαφέροντος οι μέγιστες τιμές θα ήταν διαφορετικές. Για το λόγο αυτό, οι μέγιστες τιμές αν και είναι χαρακτηριστικά μεγέθη της κίνησης της πλάκας, εντούτοις δεν δίνουν πληροφορίες για τη συνολική κίνηση της πλάκας. Για την ποιοτική εκτίμηση της κίνησης, οι καλύτερες πληροφορίες προκύπτουν από τα βέλη που εμφανίζονται στην οθόνη κατά την εκτέλεση του προγράμματος και δείχνουν την κατεύθυνση και το μέτρο της κίνησης κάθε σημείου ενδιαφέροντος. Δυστυχώς όμως, οι πληροφορίες αυτές δε μπορούν να παρασταθούν σε πίνακες, αφού είναι υπερβολικά πολλές. Εντούτοις, στο παράρτημα Β παρουσιάζονται οι εικόνες με τα βέλη όπως ακριβώς εμφανίζονται στην οθόνη του υπολογιστή.

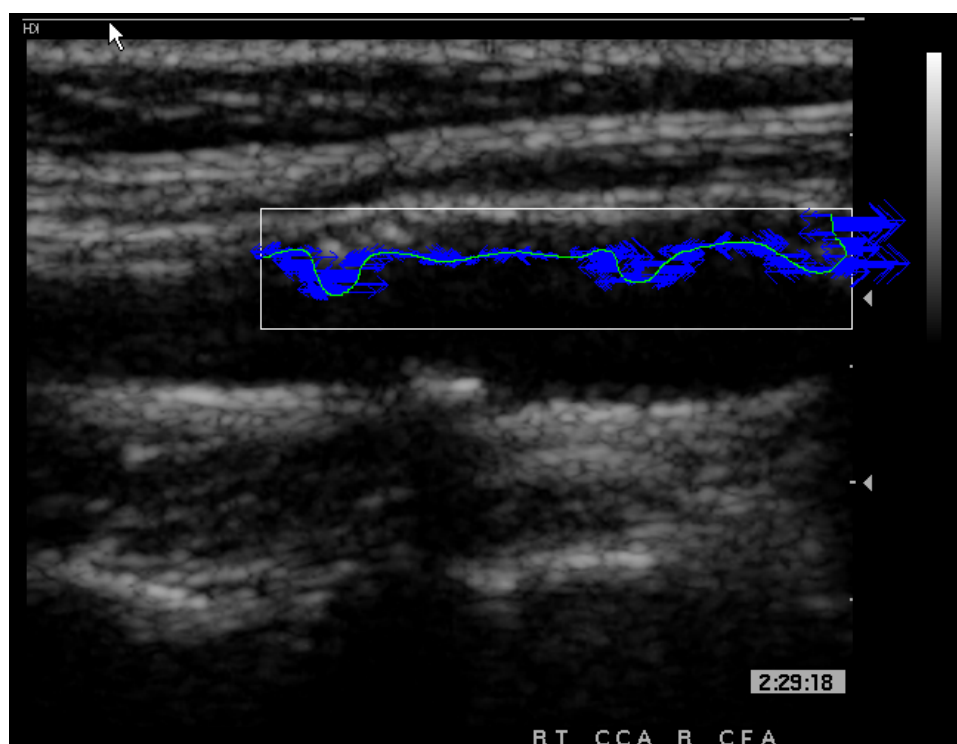
Στη συνέχεια παρατίθενται δύο εικόνες από κάθε κατηγορία αθηρωματικών πλακών. Επάνω στις εικόνες εμφανίζεται η μέγιστη ταχύτητα ως προς τους άξονες x και y που αναπτύσσει κάθε σημείο ενδιαφέροντος καθόλη τη διάρκεια της ακολουθίας.



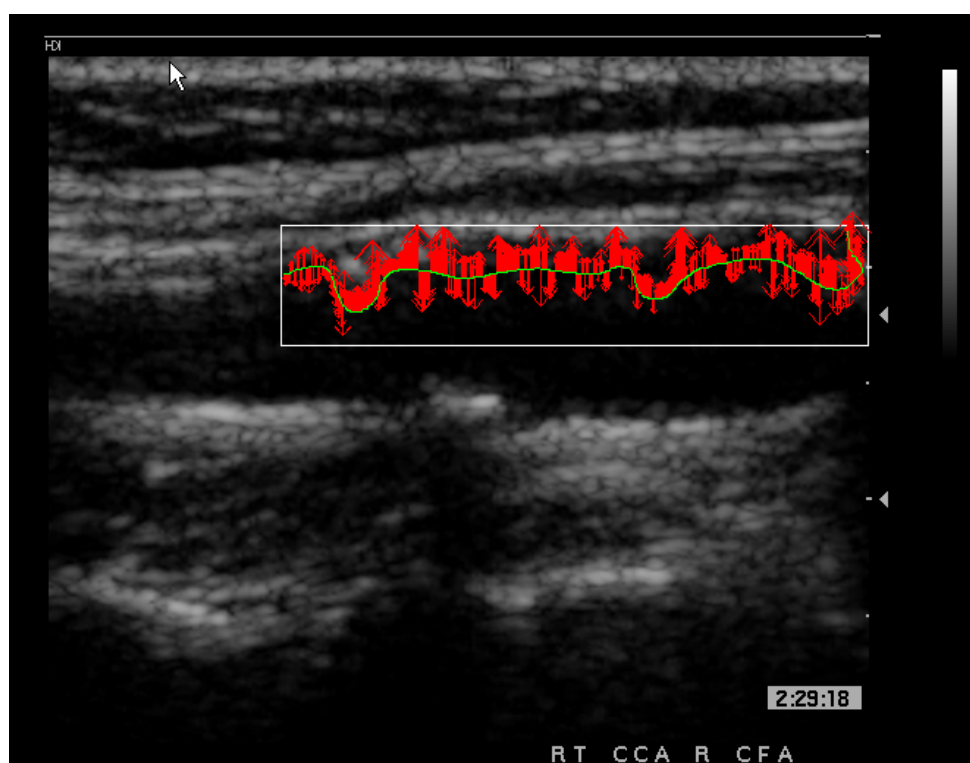
Εικόνα 4 - Κίνηση ασυμπτωματικής πλάκας στη διεύθυνση x



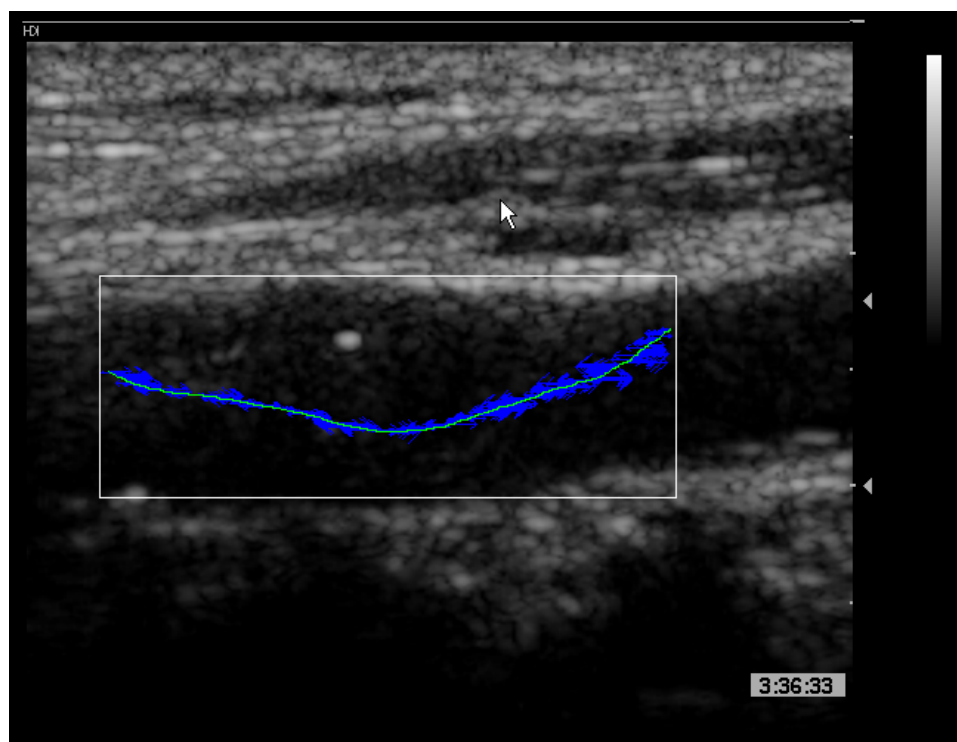
Εικόνα 5 - Κίνηση ασυμπτωματικής πλάκας στη διεύθυνση y



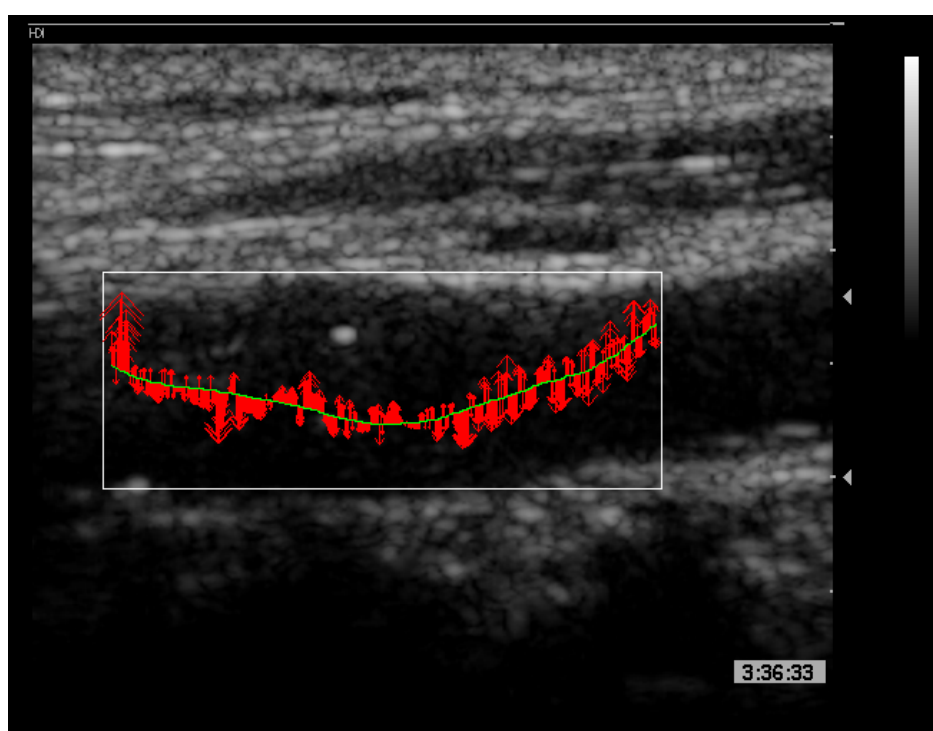
Εικόνα 6 - Κίνηση ασυμπτωματικής πλάκας στη διεύθυνση x



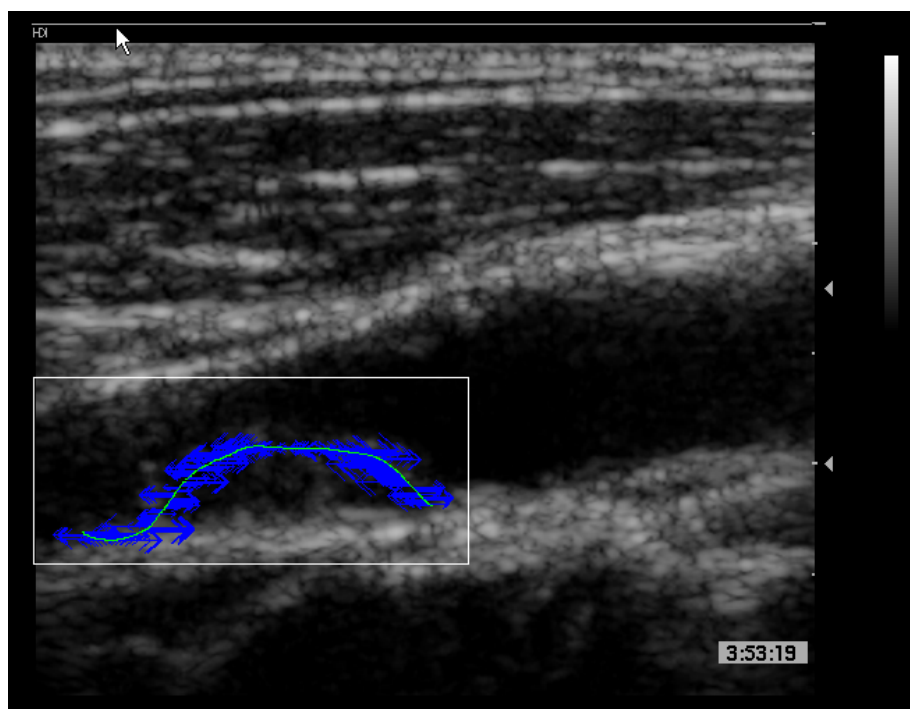
Εικόνα 7 - Κίνηση ασυμπτωματικής πλάκας στη διεύθυνση y



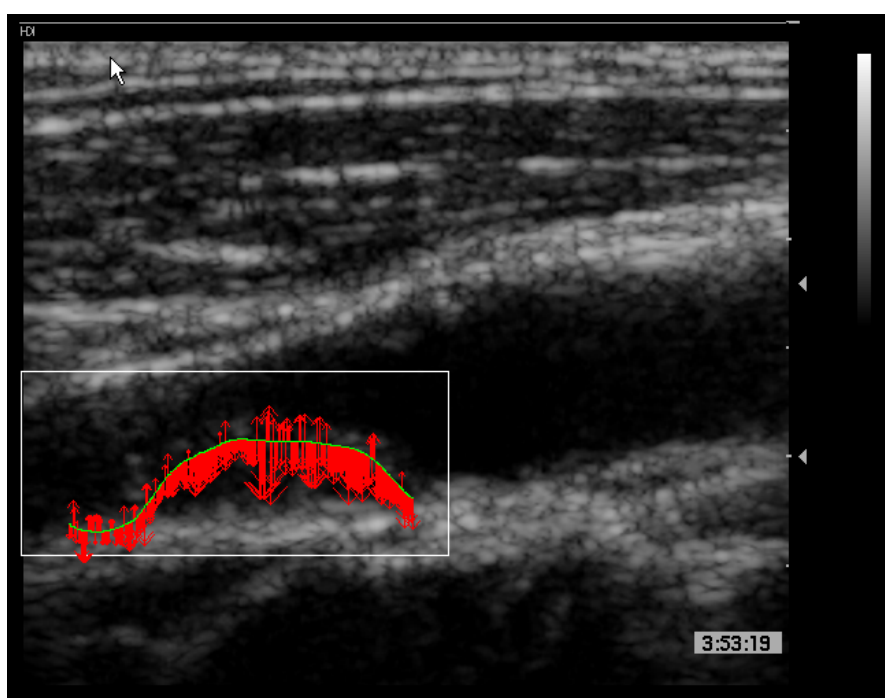
Εικόνα 8 - Κίνηση συμπτωματικής πλάκας στη διεύθυνση x



Εικόνα 9- Κίνηση συμπτωματικής πλάκας στη διεύθυνση y



Εικόνα 10- Κίνηση συμπτωματικής πλάκας στη διεύθυνση x



Εικόνα 11 - Κίνηση συμπτωματικής πλάκας στη διεύθυνση y

Κεφάλαιο 4^ο

ΣΥΖΗΤΗΣΗ

4.1 Αξιολόγηση της αναπτυχθείσας μεθόδου

Προκειμένου να καταστεί δυνατή η αξιολόγηση των αποτελεσμάτων που δίνει η αναπτυχθείσα μέθοδος, στην επιλογή Play→Play Sequence για κάθε frame της ακολουθίας που αναπαράγεται εμφανίζονται και τα σημεία ενδιαφέροντος που αντιστοιχούν στο τρέχον frame. Δηλαδή εμφανίζεται η θέση για κάθε σημείο όπως προκύπτει από την υπολογιζόμενη μετατόπιση. Ο συγκεκριμένος τρόπος εκτίμησης της μεθόδου είναι ποιοτικός και όχι ποσοτικός. Όμως, επιτρέπει να γίνει αντιληπτό κατά πόσο η κίνηση εκτιμάται σωστά. Εάν δηλαδή τα σημεία ενδιαφέροντος ακολουθούν την κίνηση των φυσικών σημείων.

Πράγματι, βάσει των 19 ακολουθιών που μελετήθηκαν, η γενική εκτίμηση είναι ότι η μέθοδος αυτή για τις περισσότερες από τις ακολουθίες εκτιμά σωστά την κίνηση. Προφανώς εάν δεν ακολουθείται σωστά η κίνηση τότε δεν έχουν και νόημα τα ποσοτικά στοιχεία της μεθόδου που προκύπτουν.

Συγκριτικά με άλλες μελέτες⁴⁹ στις οποίες έχουν μελετηθεί αντίστοιχες ταχύτητες για τα σημεία ενδιαφέροντος, παρατηρούμε ότι η τάξη μεγέθους συμφωνεί μεν αλλά είναι οι ταχύτητες είναι περίπου διπλάσιες. Οι τιμές για τα 2 μεγέθη ακολουθούν κανονική κατανομή και επομένως ενδείκνυται η χρήση t-test για μη συσχετιζόμενα δείγματα. Η δοκιμασία δείχνει ότι οι τιμές των μεγεθών για τις δύο μεθόδους διαφέρουν στατιστικώς σημαντικά $p=0.000$.

Ωστόσο, αξίζει να αναζητηθούν στη συνέχεια οι παράγοντες που οδηγούν στην αστοχία της μεθόδου σε ορισμένες περιπτώσεις.

4.2 Αδυναμίες της αναπτυχθείσας μεθόδου

Όπως έχει ήδη αναφερθεί, κατά την εκτίμηση της κίνησης με τη μέθοδο της οπτικής ροής, δεν γίνεται αναζήτηση του κάθε pixel στο επόμενο frame αλλά με βάση τις διάφορες μεταβολές φωτεινότητας εκτιμάται μια αναμενόμενη μετατόπιση. Επομένως η όλη εκτίμηση της μετατόπισης ενός pixel για το τρέχον frame, εξαρτάται από τη συνολική εκτίμηση για όλα τα προηγούμενα frames.

Πράγματι, κυριότερη αδυναμία της μεθόδου αποτελεί η παρακολούθηση του "πραγματικού" σημείου ενδιαφέροντος. Πιο συγκεκριμένα, επιλέγεται στην πρώτη εικόνα της ακολουθίας ένα σημείο ενδιαφέροντος. Στη συνέχεια και για κάθε διαδοχικό ζευγάρι εικόνων υπολογίζεται μια νέα θέση για το σημείο αυτό, βάσει των αποτελεσμάτων της μεθόδου οπτικής ροής. Δυστυχώς, εάν για κάποιο λόγο δεν εντοπιστεί σωστά η νέα θέση του σημείου, η μέθοδος συνεχίζει κανονικά να υπολογίζει ταχύτητες και επόμενες θέσεις αλλά για κάποιο άλλο σημείο και όχι το πραγματικό σημείο ενδιαφέροντος. Δηλαδή εάν χαθεί το πραγματικό σημείο τότε είναι αδύνατον να ξαναβρεθεί.

Κυριότερος λόγος αστοχίας στον σωστό υπολογισμό της νέα θέσης ενός σημείου είναι η διακριτή φύση της μεθόδου. Δηλαδή κάθε αποτέλεσμα πρέπει να έχει ακέραιη τιμή και όχι δεκαδική. Έτσι κάθε φορά που υπολογίζεται η μετατόπιση ενός σημείου υπεισέρχεται ένας παράγοντας σφάλματος λόγω ακριβώς της χρήσης μόνο ακεραίων τιμών. Όσο μεγαλύτερη είναι η ακολουθία των εικόνων, δηλαδή όσο περισσότεροι είναι οι υπολογισμοί μετατόπισης, τόσο πιο μεγάλο είναι το συνολικό σφάλμα.

Ακριβώς για να ελαττωθεί όσο το δυνατόν περισσότερο το σφάλμα λόγω δεκαδικών μερών, για κάθε μετατόπιση λαμβάνονται υπόψη όλες οι προηγούμενες ταχύτητες του σημείου με ακρίβεια αρκετών δεκαδικών ψηφίων. Δηλαδή, για κάθε frame η νέα θέση του σημείου υπολογίζεται από

την αρχική θέση του σημείου με πρόσθεση όλων των μετατοπίσεων μέχρι το τρέχον frame, με χρήση κατά την πρόσθεση όλων των δεκαδικών ψηφίων. Έτσι εάν για παράδειγμα η υπολογιζόμενη μετατόπιση είναι 0.5 για 2 διαδοχικά ζευγάρια frames τότε η μετατόπιση που θα θεωρηθεί στο τέταρτο κατά σειρά frame θα είναι 1.

Ένα άλλο θέμα που τίθεται είναι αυτό της μέγιστης μετατόπισης. Η μέθοδος στηρίζεται στην παραδοχή ότι το σημείο μετακινείται πολύ λίγο με αποτέλεσμα να ισχύει η σχέση $g(x+\Delta x, y+\Delta y, t+\Delta t)=g(x, y, t)$. Εάν όμως τα Δx και Δy που υπολογίζονται είναι μεγάλα τότε η παραδοχή αυτή παύει να ισχύει. Δηλαδή τα Δx και Δy πρέπει να έχουν τέτοια τιμή ώστε να μπορεί να θεωρηθεί η μετατόπιση σημειακή. Έχει αναφερθεί⁶⁴ ότι για μετατόπιση μέχρι 3 pixels/frame μπορεί να χρησιμοποιηθεί η μέθοδος οπτικής ροής, εφόσον θεωρηθεί καταχρηστικά σημειακή η μετατόπιση. Μεγαλύτερες όμως μετατοπίσεις θα πρέπει να αγνοούνται ή να ορίζονται ίσες με κάποιο προκαθορισμένο ανώτατο όριο όπως αυτό των 3 pixels.

Τέλος, η όλη μέθοδος στηρίζεται στην παραδοχή ότι η φωτεινότητα ενός φυσικού σημείου παραμένει η ίδια σε όλη τη διάρκεια της ακολουθίας. Κάτι τέτοιο όμως είναι πολύ γενικό και μάλλον αμφίβολο να ισχύει απόλυτα σε εικόνες όπως αυτές των υπερήχων, στις οποίες η διακριτική ικανότητα της εικόνας είναι πολύ περιορισμένη. Συνεπώς, η ποιότητα των εικόνων υπερήχων επηρεάζει την ποιότητα των εξαγόμενων αποτελεσμάτων. Βέβαια, καθώς η ποιότητα των συσκευών υπερήχων βελτιώνεται, γίνεται εφικτή η ολοένα καλύτερη ποιότητα εικόνων και άρα καλύτερη απόδοση των τεχνικών εκτίμησης κίνησης.

Παρατηρήθηκε ότι για πολύ φωτεινές εικόνες η εκτίμηση της κίνησης δεν είναι καθόλου επιτυχημένη σε σύγκριση με τις άλλες λιγότερο φωτεινές εικόνες. Αυτό συμφωνεί με ό,τι έχει αναφερθεί στην θεωρητική τεκμηρίωση στο κεφάλαιο 2. Δηλαδή ότι η βασική σχέση 2.2 δίνει καλύτερα αποτελέσματα

για τις περιοχές της εικόνας στις οποίες υπάρχει μεγάλη μεταβολή της φωτεινότητας ως προς x ή/ και y . Έτσι, στις φωτεινές περιοχές, η τιμή της φωτεινότητας είναι κοντά στο 0 αφού στην αναπαράσταση των χρωμάτων στις grayscale εικόνες το άσπρο χρώμα έχει τιμή 0 ενώ το μαύρο 255. Το πρόβλημα αυτό οδηγεί στον υπολογισμό πολύ μεγάλων μετατοπίσεων, κάτι που με τη σειρά του οδηγεί στο να χαθεί το πραγματικό σημείο ενδιαφέροντος, όπως αναφέρθηκε και προηγούμενα.

Μια λύση για το παραπάνω πρόβλημα μπορεί να αναζητηθεί προς την κατεύθυνση της εφαρμογής κατάλληλου φίλτρου. Συγκεκριμένα, αποδοτικό θα ήταν ένα φίλτρο που θα έκανε εντονότερες τις μεταβολές της φωτεινότητας και θα φιλτράριζε τις μεγάλες εντάσεις φωτεινότητας. Ασφαλώς, όλα τα παραπάνω θα πρέπει να επιτυγχάνονται από το φίλτρο χωρίς να χάνονται σημαντικές πληροφορίες για την εκτίμηση της κίνησης από την εικόνα.

4.3 Προτάσεις για περαιτέρω βελτίωση της μεθόδου

Προκειμένου να αντιμετωπισθούν οι διάφορες δυσκολίες της μεθόδου πρέπει να αναζητηθούν κατάλληλες λύσεις.

Για το πρόβλημα της παρακολούθησης του "πραγματικού" σημείου ενδιαφέροντος η πιο αποδοτική λύση από πλευράς εκτίμησης κίνησης είναι η εκτέλεση του αλγορίθμου μόνο για δύο διαδοχικά frames. Βέβαια κάτι τέτοιο δεν έχει πρακτικό νόημα αφού ο χρήστης θα πρέπει συνεχώς να επιλέγει τα όρια της πλάκας για κάθε frame. Εντούτοις, μια μέση λύση θα ήταν ο αλγόριθμος να εκτελείται για μικρό αριθμό frames ενώ για μεγαλύτερο αριθμό κάπου ενδιάμεσα να καλείται ο χρήστης να επιλέξει τα όρια της πλάκας. Ασφαλώς, θα μπορούσε να ενσωματωθεί στον αλγόριθμο κάποια μέθοδος υπολογισμού των ορίων της πλάκας. Έτσι μετά από κάποιο αριθμό frames, αυτόματα να εκτελείται ο αλγόριθμος εντοπισμού των ορίων της

πλάκας, προκειμένου να συνεχιστεί η εκτίμηση της κίνησης για τα "πραγματικά" σημεία ενδιαφέροντος.

Ένας άλλος τρόπος βελτίωσης της απόδοσης της αναπτυχθείσας μεθόδου θα ήταν μέσω της τρισδιάστατης ανακατασκευής των πλακών, κάτι το οποίο θα οδηγούσε στη εξαγωγή τρισδιάστατων χαρακτηριστικών κίνησης που θα βοηθούσαν στον χαρακτηρισμό της πλάκας⁶⁵. Τέτοιες τρισδιάστατες πληροφορίες, οι οποίες δεν λαμβάνονται υπόψη στη δισδιάστατη θεώρηση, παρέχουν καλύτερη απεικόνιση της δομής της πλάκας και συνεισφέρουν στην κατεύθυνση της πιο επιτυχούς διάγνωσης. Συγκεκριμένα, οι τεχνικές εκτίμησης κίνησης δεν λαμβάνουν υπόψη πολλούς σημαντικούς παράγοντες, ο σπουδαιότερος των οποίων είναι παραμόρφωση και αλλοίωση των πλακών. Προφανώς, με τη χρήση τρισδιάστατων μοντέλων κίνησης η ισχύς της παραδοχής αναβαθμίζεται.

4.4 Σχολιασμός αποτελεσμάτων

Οι προηγούμενες προσπάθειες για ταυτοποίηση των αθηρωματικών πλακών με αποκλειστική χρήση εικόνων υπερήχων δεν στέφθηκαν με επιτυχία⁶⁶⁻⁷⁰. Η αδυναμία χαρακτηρισμού των πλακών βάσει μόνο της μορφολογίας τους τονίζεται από την επίσης αδυναμία παθολογικών ερευνών να ταυτοποιήσουν διαφορές στη σύνθεση ή στην επιφανειακή δομή των πλακών που λαμβάνονται από τις επεμβατικές μεθόδους⁷¹⁻⁷³. Προφανώς, οι μέχρι στιγμής αποτυχίες παρέχουν ισχυρό κίνητρο για αναζήτηση και νέων μεθόδων χαρακτηρισμού των αθηρωματικών πλακών.

Σκοπός της παρούσας εργασίας ήταν δημιουργία κατάλληλου εργαλείου υποβοήθησης των ειδικών, για τον χαρακτηρισμό της αθηρωματικής πλάκας. Στην μορφή που έχει αυτή τη στιγμή η εφαρμογή, αν και υπολογίζονται πληθώρα στοιχείων και ταχυτήτων, μόνον ένας περιορισμένος αριθμός εμφανίζεται στον τελικό χρήστη. Αυτό συμβαίνει διότι οι πληροφορίες που

προκύπτουν είναι τόσες πολλές που αν εμφανίζονταν όλες στον χρήστη δεν θα μπορούσε να τις αξιοποιήσει λόγω του μεγάλου όγκου. Σε μελλοντική έκδοση, θα μπορούσε να ενσωματωθεί στατιστική επεξεργασία αυτού του τεράστιου όγκου δεδομένων προκειμένου να βοηθήσει στη λήψη αποφάσεων.

Επιπλέον, για τον υπολογισμό των μετατοπίσεων λαμβάνονται υπόψη μόνο τα συγκεκριμένα σημεία ενδιαφέροντος που επέλεξε ο χρήστης. Δηλαδή τα όρια της πλάκας που καθορίστηκαν από τον χρήστη έχουν ελάχιστο πάχος μόλις ένα pixel. Κάτι τέτοιο δεν είναι λάθος αν έχει γίνει σωστή επιλογή ορίων από τον χρήστη. Όμως επειδή υπεισέρχεται και ο υποκειμενικός παράγοντας κατά την επιλογή των ορίων της πλάκας, θα ήταν επιθυμητό να αυξηθεί το ελάχιστο πάχος σε αρκετά pixels. Με τον τρόπο αυτό η εκτίμηση για κάθε σημείο θα γινόταν συναρτήσει των γειτονικών του. Πιο συγκεκριμένα, και στην παρούσα μέθοδο η ταχύτητα κάθε σημείου εξαρτάται από αυτή των γειτονικών του αλλά με τον ίδιο συντελεστή βαρύτητας. Αντιθέτως, εάν το πάχος της πλάκας ληφθεί μεγαλύτερο τότε τα γειτονικά σημεία ενδιαφέροντος θα έχουν μεγαλύτερο συντελεστή βαρύτητας.

Τέλος, περαιτέρω πηγές πληροφοριών όπως κλινικά δεδομένα, παράγοντες κινδύνου και παράγοντες σχετικοί με την πρόκληση εγκεφαλικού επεισοδίου θα μπορούσαν να ενσωματωθούν σε ένα μελλοντικό σύστημα διάγνωσης. Μάλιστα, σε ένα τέτοιο σύστημα θα ήταν αυτοματοποιημένη η όλη διαδικασία της αναγνώρισης της θέσης της πλάκας στην εικόνα καθώς και ο χαρακτηρισμός της. Με τον τρόπο αυτό προφανώς θα απαλειφθούν τα σφάλματα του χρήστη στον καθορισμό της θέσης και του μεγέθους της πλάκας.

4.5 Συμπέρασμα

Μπορεί, εκ πρώτης όψεως, να μην κατέστη εφικτή -όπως και σε προηγούμενες προσπάθειες- η διαφοροποίηση των αποτελεσμάτων μεταξύ

των συμπτωματικών και ασυμπτωματικών πλακών, με τη χρήση μορφολογικών κριτηρίων όπως η δομή της επιφανείας της πλάκας. Εντούτοις, τα αποτελέσματα θα πρέπει να μελετηθούν στατιστικά προτού να αξιολογηθεί η επιτυχία της μεθόδου. Η συγκεκριμένη αναπτυχθείσα μέθοδος είναι σε θέση να δώσει πληθώρα πληροφοριών για τον τρόπο κίνησης κάθε σημείου ενδιαφέροντος. Οι πληροφορίες αυτές μπορούν να συνδυαστούν για την υποβοήθηση της ιατρικής διάγνωσης.

Όλα τα ανωτέρω επιβεβαιώνουν ότι επιτεύχθηκε ο σκοπός της μελέτης, δηλαδή η ανάπτυξη ενός εργαλείου που να μετρά τις κινήσεις της αθηρωματικής πλάκας προκειμένου να γίνει εφικτός ο συσχετισμός της υψή της πλάκας με τη μορφολογία των εικόνων που λαμβάνονται με χρήση υπερήχων. Με τον τρόπο αυτό διευκολύνει το χαρακτηρισμό των αθηρωματικών πλακών σε συμπτωματικές και ασυμπτωματικές που είχαν τεθεί σαν αρχικοί στόχοι. Βέβαια η αναπτυχθείσα μέθοδος θα ήταν πιο πλήρης αν περιλαμβάνετο και ο χαρακτηρισμός των αποτελεσμάτων, θέμα όμως που θα μπορούσε να αποτελέσει αντικείμενο άλλων μελετών.

Μάλιστα, εφόσον καταστεί εφικτός ο χαρακτηρισμός μιας ομάδας ασθενών υψηλής επικινδυνότητας με βάση μόνο τα χαρακτηριστικά μιας ακολουθίας εικόνων υπερήχων υψηλής ανάλυσης, οι ασθενείς της ομάδας αυτής θα επωφεληθούν των αποτελεσμάτων μιας χειρουργικής επέμβασης. Επιπλέον οι ασθενείς της άλλης ομάδας θα γλιτώσουν την μη απαραίτητη επέμβαση. Προφανώς τα όποια αποτελέσματα θα πρέπει να επαληθευθούν με περισσότερες εικόνες.

Κεφάλαιο 5°

ΚΩΔΙΚΑΣ

5.1 Γενικά

Η εφαρμογή OFC έχει γραφεί σε γλώσσα C++ και πιο συγκεκριμένα έχει αναπτυχθεί στο περιβάλλον **MS Visual Studio .NET**. Η όλη εφαρμογή αποτελείται από 10 συνολικά αρχεία τόσο .cpp όσο και .h. Από τα αρχεία αυτά, ιδιαίτερο ενδιαφέρον παρουσιάζουν μόνο τα *ChildView.cpp/h*, αφού τα υπόλοιπα είναι βοηθητικά για τις διάφορες λειτουργίες που απαιτούνται να υποστηρίζονται λόγω του γραφικού περιβάλλοντος των Windows. Επιπλέον, κάποια περιέχουν τις χρησιμοποιούμενες κλάσεις. Χρησιμοποιήθηκαν μόνο κλάσεις που βρίσκονται στην MFC.

Όλα τα αρχεία βρίσκονται στο CD - στο φάκελο source - ενώ το listing τους βρίσκεται στο παράρτημα Α.

Ακολούθως θα εξηγηθούν τα βασικά χαρακτηριστικά τους.

5.2 *MainFrame.cpp/h*

Στα δύο αυτά αρχεία, ορίζεται η μορφή του κεντρικού παραθύρου του προγράμματος. Δηλαδή το αρχικό μέγεθος, το στυλ καθώς και τα μενού. Κατά τα άλλα δεν διαδραματίζει άλλο ρόλο.

5.3 *CImage.cpp/h*

Στα δύο αυτά αρχεία, ορίζεται η κλάση *CImage* της MFC, η οποία χρησιμοποιείται στο πρόγραμμα για τον χειρισμό των εικόνων tiff.

5.4 stdafx.cpp/h

Τα δύο αυτά αρχεία έχουν προστεθεί αυτόματα από το .NET για χρησιμοποιηθούν από τον compiler.

5.5 Resource.h

Στο αρχείο αυτό ορίζονται οι διάφορες σταθερές που χρησιμοποιούνται στο πρόγραμμα.

5.6 GdiPlus.h

Το αρχείο αυτό χρησιμοποιείται επίσης για την επεξεργασία των γραφικών.

5.7 ChildView.cpp/h

Στα δύο αυτά αρχεία βρίσκεται ουσιαστικά όλο το πρόγραμμα. Στο ChildView.h ορίζονται όλες οι μεταβλητές καθώς και οι συναρτήσεις που θα χρησιμοποιηθούν από την κλάση. Η ανάλυση θα επικεντρωθεί στις σημαντικότερες συναρτήσεις του ChildView.cpp. Σημειώνεται ότι πάνω στον κώδικα υπάρχουν σχόλια για να βοηθήσουν όποιον διαβάζει τον κώδικα.

void OnFileOpenImage(void)

Η συνάρτηση εκτελείται όταν ο χρήστης επιλέξει από το μενού File→Open Image. Ανοίγει ένα dialog Open File... στο οποίο ο χρήστης επιλέγει την πρώτη εικόνα της ακολουθίας, η οποία πρέπει να είναι της μορφής filename#.tif, όπου # οποιοσδήποτε αριθμός. Η συνάρτηση βρίσκει τα επόμενα frames της ακολουθίας και αποθηκεύει στην μεταβλητή *fname* την πλήρη διαδρομή του πρώτου αρχείου χωρίς τον αριθμό και την κατάληξη tif. Επίσης αποθηκεύει στην μεταβλητή *NumFiles* το πλήθος των frames

προς επεξεργασία και στην μεταβλητή *digs* τον αριθμό που αντιστοιχεί στην πρώτη εικόνα.

void OnSelectROI (void)

Όταν γίνει η επιλογή *Run→Select ROI*, τότε εκτελείται η συγκεκριμένη συνάρτηση μέσω της οποίας γίνεται αληθής η λογική μεταβλητή *selectROI*. Η μεταβλητή αυτή επιτρέπει την επιλογή της ROI από τον χρήστη με τη βοήθεια του mouse. Όσο είναι ψευδής αγνοούνται οι κινήσεις του mouse. Επιπλέον, εμφανίζεται ένα παράθυρο που απεικονίζει τις συντεταγμένες του σημείου στο οποίο δείχνει το mouse.

void Compute (void)

Η συνάρτηση αυτή αποτελεί την καρδιά της μεθόδου. Αφού έχει καθοριστεί η ROI, ο χρήστης μπορεί να επιλέξει *Run→Compute* οπότε εκτελείται η συγκεκριμένη συνάρτηση. Η συνάρτηση ανοίγει μια-μια της εικόνες και εφαρμόζει τη μέθοδο οπτικής ροής. Χρησιμοποιούνται οι βοηθητικές συναρτήσεις *NextFrame (CImage &image, int frame)* και *FirstFrame(CImage &image)* οι οποίες αναλύονται αμέσως μετά και ουσιαστικά υπολογίζουν τα διανύσματα κίνησης. Επιπλέον γίνεται η ομαλοποίηση των ταχυτήτων μέσω της *NormVelocities(void)*.

Τέλος, γίνονται δευτερεύουσες λειτουργίες όπως η εμφάνιση progress bar.

void FirstFrame(CImage &image)

Εδώ απλώς διαβάζονται οι τιμές φωτεινότητας για όλα τα pixels που περιέχονται στην ROI - στο πρώτο από τα 2 διαδοχικά frames.

void NextFrame (CImage &image, int frame)

Σε πρώτη φάση διαβάζονται οι τιμές φωτεινότητας για όλα τα pixels που περιέχονται στην ROI- στο δεύτερο από τα 2 frames.

Στη συνέχεια υπολογίζονται τα $\frac{\partial g}{\partial x}$, $\frac{\partial g}{\partial y}$ και $\frac{\partial g}{\partial t}$ για τα δύο αυτά frames. Ο

υπολογισμός γίνεται με συνέλιξη του πίνακα των φωτεινотήτων με την κατάλληλη μάσκα PreWitt για κάθε μερική παράγωγο. Προφανώς τα σημεία της περιμέτρου τα οποία δεν έχουν 9 γειτονικά pixels εξετάζονται χωριστά από όλα τα υπόλοιπα.

Έχοντας υπολογιστεί όλες οι απαιτούμενοι μερικές παράγωγοι καλείται η συνάρτηση *HornSchunk(int frame)*.

int HornSchunk(int frame)

Στη συνάρτηση αυτή υλοποιείται ο υπολογισμός των ταχυτήτων βάσει της περιοριστικής παραδοχής των Horn και Schunk.

Έτσι, αρχικά υπολογίζονται τα u_{ave} και v_{ave} για τα 9 γειτονικά σημεία κάθε pixel - εκτός από τα σημεία τις περιμέτρου για τα οποία υπολογίζονται πάλι τα ίδια μεγέθη αλλά για λιγότερα γειτονικά σημεία. Στη συνέχεια υπολογίζονται τα u^k και v^k .

Η όλη διαδικασία επαναλαμβάνεται μέχρι η μέση τετραγωνική μεταβολή και των δύο ταχυτήτων να είναι μικρότερη από 10^{-5} ή να γίνουν 1000 επαναλήψεις.

void NormVelocities(void)

Με τη συνάρτηση αυτή, γίνεται μια προσπάθεια για ομαλοποίηση των τιμών των ταχυτήτων που υπολογίστηκαν. Έτσι, για τις ταχύτητες που έχουν υπολογιστεί για κάθε frame λαμβάνεται ο μέσος όρος των 9 γειτονικών pixels. Γίνεται αυτή η επιπλέον ομαλοποίηση (έχει ήδη γίνει με τη χρήση των

μασκών PreWitt) αφού όπως φάνηκε από το αποτέλεσμα έτσι η κίνηση είναι πιο ομαλή και η μέθοδος αποδίδει καλύτερα.

void OnRunMarkLimits(void)

Η συνάρτηση εκτελείται όταν επιλεγεί Run→Mark Limits από τον χρήστη. Εδώ εμφανίζεται μια μεγεθυμένη εικόνα της ROI προκειμένου να μπορέσει ο χρήστης να επιλέξει τα σημεία ενδιαφέροντος με το mouse. Πάλι, η λογική μεταβλητή *MarkLimits* γίνεται αληθής προκειμένου να "ενεργοποιηθεί" το mouse.

void OnRunDonemarkinglimits (void)

Επιλέγοντας ο χρήστης Run→DONE Marking Limits εκτελείται η συγκεκριμένη συνάρτηση, η οποία είναι η δεύτερη πιο σημαντική για τη μέθοδό μας.

Αρχικά υπολογίζονται τα σημεία ενδιαφέροντος που επέλεξε ο χρήστης. Επειδή η επιλογή έγινε σε μεγέθυνση, οι συντεταγμένες μετατρέπονται ώστε να αντιστοιχούν στις πραγματικές διαστάσεις της εικόνας.

Ακολουθως, εκτελούνται διαδοχικά οι συναρτήσεις *CompPoints(void)*, *MaxVelocity(void)*, *MaxMaxVelocity(void)*, *SortPoints(void)* και *MaxSuccessiveDiff(void)* μέσω των οποίων υπολογίζονται οι πληροφορίες για την κίνηση των σημείων ενδιαφέροντος.

void CompPoints(void)

Η συνάρτηση αυτή υπολογίζει τη νέα θέση για κάθε pixel - σημείο ενδιαφέροντος στο επόμενο frame. Για κάθε σημείο, προσθέτει στην αρχική θέση του σημείου όλες τις μετατοπίσεις που έχουν υπολογιστεί, με ακρίβεια αρκετών δεκαδικών ψηφίων, προκειμένου να βρεθεί η νέα θέση.

Η όλη διαδικασία επαναλαμβάνεται για όλα τα σημεία ενδιαφέροντος και για όλα τα ζεύγη διαδοχικών frames

void MaxVelocity(void)

Στη συνάρτηση αυτή υπολογίζεται η μέγιστη ταχύτητα που αναπτύσσει κάθε σημείο ενδιαφέροντος στο σύνολο της ακολουθίας καθώς και σε πιο frame γίνεται η ταχύτητα μέγιστη. Προφανώς, γίνεται διαφορετική μέτρηση για την ταχύτητα κατά την διεύθυνση του άξονα x και διαφορετική για την αντίστοιχη κατά y.

void MaxMaxVelocity(void)

Στη συνάρτηση αυτή υπολογίζεται η μέγιστη ταχύτητα - πάλι κατά x και y- που αναπτύσσουν όλα τα σημεία ενδιαφέροντος στο σύνολο της ακολουθίας. Δηλαδή, ουσιαστικά βρίσκεται το μέγιστο από τις τιμές που προέκυψαν από την προηγούμενη συνάρτηση *MaxVelocity(void)*.

void SortPoints(void)

Μέσω της συνάρτησης αυτής, υπολογίζονται οι μέγιστες διαφορές ταχύτητας που αναπτύσσουν γειτονικά σημεία ενδιαφέροντος. Ο έλεγχος γίνεται τόσο ως προς x όσο και ως προς y.

void MaxSuccessiveDiff(void)

Σκοπός της συνάρτησης είναι ο υπολογισμός των μέγιστων διαφορών στην ταχύτητα που αναπτύσσει ένα σημείο ενδιαφέροντος σε δύο διαδοχικά frames. Οπότε, αποθηκεύονται οι μέγιστες διαφορές στους δύο άξονες καθώς και η χρονική στιγμή στην οποία συμβαίνουν αυτές οι διαφορές.

void ShowArrows(bool V)

Η συνάρτηση απεικονίζει τις μέγιστες ταχύτητες που ανέπτυξε κάθε σημείο ενδιαφέροντος, πάνω στην οθόνη με τη μορφή βέλους. Το βέλος έχει φορά και μέγεθος ανάλογο με την ταχύτητα που αναπτύχθηκε. Ανάλογα με την τιμή της λογικής μεταβλητής *V* εμφανίζονται οι μέγιστες ταχύτητες στο άξονα x ή στον άξονα y.

Την συγκεκριμένη συνάρτηση χρησιμοποιούν και οι συναρτήσεις *OnShowarrowsVx(void)*, *void OnShowarrowsVy(void)* και *void OnShowarrowsBoth(void)* που απεικονίζουν την ταχύτητα στον άξονα x, στον άξονα y και στους δύο άξονες, αντίστοιχα.

void OnRunMaxdifferences(void)

Η συγκεκριμένη συνάρτηση εκτελείται όταν ο χρήστης επιλέξει *Run→Max Differences*. Μέσω της *DrawSortPoints(void)* απεικονίζονται πάλι με τη μορφή βελών, οι μέγιστες διαφορές ταχύτητας που αναπτύχθηκαν ανάμεσα σε γειτονικά σημεία.

void OnPlayPlaysequence(void)

Όταν ο χρήστης επιλέξει *Play→Play Sequence*, τότε εκτελείται η συγκεκριμένη συνάρτηση. Μέσω αυτής, εμφανίζεται στην οθόνη όλη η ακολουθία των εικόνων σαν video.

void OnMenu(void)

Η συνάρτηση εκτελείται όταν ο χρήστης επιλέξει *File→Import Settings*. Μέσω αυτής είναι δυνατόν να εισαχθούν στο πρόγραμμα συντεταγμένες ROI και σημείων ενδιαφέροντος από προηγούμενες εκτελέσεις, εφόσον έχουν αποθηκευτεί σε εξωτερικό αρχείο, μορφή text.

void OnFileExport(void)

Αντίστροφα από την προηγούμενη συνάρτηση, η συγκεκριμένη εκτελείται όταν ο χρήστης επιλέξει File→Export Settings και οδηγεί στην αποθήκευση σε εξωτερικό αρχείο των επιλεγμένων σημείων ενδιαφέροντος και της ROI.

void OnImageInfo(void)

Από την επιλογή File→Save Results, εκτελείται η συγκεκριμένη συνάρτηση. Ουσιαστικά παραπέμπει στην *OutPutFile(void)* μέσω της οποίας αποθηκεύονται όλα τα αποτελέσματα σε ένα αρχείο text. Επιπλέον, αποθηκεύονται και τα settings σε ξεχωριστό αρχείο.

void MarkROI(void)

Η συνάρτηση αυτή δεν είναι άμεσα προσπελάσιμη από τον χρήστη αλλά εκτελείται όποτε φορτώνεται μια νέα εικόνα. Σκοπός της είναι να δημιουργεί το άσπρο περίγραμμα της ROI.

void OnMouseMove(UINT nFlags, CPoint point)

Η συγκεκριμένη συνάρτηση εκτελείται όποτε κουνιέται το mouse. Βέβαια, όσο δεν είναι αληθής κάποια από τις λογικές μεταβλητές *SelectROI* ή *MarkLimits* τότε δεν κάνει τίποτε άλλο από το να στέλνει τις συντεταγμένες του mouse στο παράθυρο των συντεταγμένων.

Τέλος υπάρχουν και οι συναρτήσεις *OnLButtonUp(UINT nFlags, CPoint point)* και *OnLButtonDown(UINT nFlags, CPoint point)* που λειτουργούν συνεπικουρικά με την *OnMouseMove(UINT nFlags, CPoint point)* για την επιλογή της ROI και των σημείων ενδιαφέροντος.

Όλες οι παραπάνω συναρτήσεις παρουσιάζονται στη συνέχεια στο παράρτημα Α.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. American Heart Association. Heart and Stroke Statistical Update. 2001.
2. Keaney, J. F. Jr. Atherosclerosis: from lesion formation to plaque activation and endothelial dysfunction. *Mol Aspects Med* 2000; **21**: 99-166.
3. Dawber, T. R. Measuring the risk of coronary heart disease in adult population groups II. Coronary heart disease in the Framingham study. *Am J Public Health* 1957; **47**: 4-25.
4. Kannel, W. B., McGee, D. L. Diabetes and cardiovascular disease. The Framingham study. *JAMA* 1979; **241**: 2035-2038.
5. Stry, H. C. Evolution and progression of atherosclerotic lesions in coronary arteries of children and young adults. *Arteriosclerosis* 1989; **9**: I19-I32.
6. English, J. P., Willius, F. A., and Berkse, N. J. Tobacco and coronary disease. *JAMA* 1940; **115**: 1327-1329.
7. Doll, R. and Hill, A. B. Lung cancer and other causes of death in relation to smoking: a second report on the mortality of British doctors. *Br Med J* 1956; **2**: 1071-1081.
8. Hammond, E. C., Horn, D. Smoking and death rates: report on forty-four months of follow-up of 187,783 men. 2. Death rates by cause. *J Am Med Assoc* 1958; **166**: 1294-1308.
9. US department of Health and Human Services. Reducing the health consequences of smoking: 44 years of progress. *A report of the Surgeon General. DHSS CDC*, 1989; **89**: 8411-8419.
10. Gaziano, J. M. Epidemiology of risk factor reduction. *In: Vascular Medicine*. Loscalzo, J., Creagher, M., Dzau, V.(Eds), Little Brown, Boston. pp 569-586, 1996.
11. MacMahon, S., Peto, R., Cutler, J., Collins, R., Sorlie, P., Neaton, J. Blood pressure, stroke, and coronary heart disease. Part 1, Prolonged differences in blood pressure: prospective observational studies corrected for the regression dilution bias. *Lancet* 1990; **335**: 765-774.
12. Collins, R., Peto, R., MacMahon, S., Hebert, P., Fiebach, N. H., Eberlein, K. A. Blood pressure, stroke, and coronary heart disease. Part 2, Short-term reductions in blood pressure: overview of randomised drug trials in their epidemiological context. *Lancet* 1990; **335**: 827-838.
13. Bierman, E. L. George Lyman Duff Memorial Lecture. Atherogenesis in diabetes. *Arterioscler Thromb* 1992; **12**: 647-656.

14. Nishigaki, I. Hagihara, M., Tsunekawa, H., Maseki, M., Yagi, K. Lipid peroxide levels of serum lipoprotein fractions of diabetic patients. *Biochem Med* 1981; **25**: 373-378.
15. Rayer, P. *Arch Gen de Med* (Paris) 1823; **1**: 313.
16. Virchow, R. *Wien Med Wochenschr* 1856; **6**: 825.
17. Saphir, O. and Gore, I. *Arch Pathol* 1950; **49**: 418.
18. Joris, I., Majno, G. Cell-to-cell herniae in the arterial wall. I. The pathogenesis of vacuoles in the normal media. *Am J Pathol* 1977; **87**: 375-398.
19. Rokitansky, C. A manual of pathological anatomy. Sydenham Society. 1852.
20. Virchow, R. Cellular pathology: Lecture VIII. Blood and lymph. *CA Cancer J Clin* 1975; **25**: 93-97.
21. Ross, R., Glomset, J, Harker, L. Response to injury and atherogenesis. *Am J Pathol* 1977; **86**: 675-684.
22. Ross, R., Glomset, J. A. Atherosclerosis and the arterial smooth muscle cell: Proliferation of smooth muscle is a key event in the genesis of the lesions of atherosclerosis. *Science* 1973; **180**: 1332-1339.
23. Consigny, P. M. Pathogenesis of atherosclerosis. *AJR Am J Roentgenol* 1995; **164**: 553-558.
24. Jonasson, L., Holm, J., Skalli, O., Bondjers, G., Hansson, G. K. Regional accumulations of T cells, macrophages, and smooth muscle cells in the human atherosclerotic plaque. *Arteriosclerosis* 1986; **6**: 131-138.
25. Ross, R., Glomset, J. A. The pathogenesis of atherosclerosis (first of two parts). *N Engl J Med* 1976; **295**: 369-377.
26. Ross, R. Atherosclerosis--an inflammatory disease. *N Engl J Med* 1999; **340**: 115-126.
27. Demopoulos, C. A., Karantonis, H. C., Antonopoulou, S. Platelet activating factor - a molecular link between atherosclerosis theories *Eur. J. Lipid Sci. Technol.* 2003; **105**: 705-716
28. American Diabetes association. Consensus Statement: Role of cardiovascular risk factors in prevention and treatment of macrovascular disease in diabetes. *Diabetes Care* 1993; **16**: 72-78.
29. Schwartz, C. J., Valente, A. J., Sprague, E. A., Kelley, J. L., Cayatte, A. J., Rozek, M. M. Pathogenesis of the atherosclerotic lesion. Implications for diabetes mellitus. *Diabetes Care* 1992; **15**: 1156-1167.
30. Laakso, M. Hyperglycemia and cardiovascular disease in type 2 diabetes. *Diabetes* 1999; **48**: 937-942.
31. Kuhn, N., Heydeck, D., Hugou, I., and Gniwotta, C. The diabetes control and complication trial. *N Engl J Med* 1990; **329**: 80-85.

32. Fioretto, P., Steffes, M. W., Sutherland, D. E., Goetz, F. C., Mauer, M. Reversal of lesions of diabetic nephropathy after pancreas transplantation. *N Engl J Med* 1998; **339**: 69-75.
33. Maillard, L. C. Action des adices amines sur les sucres: Formation des melanoidines par voice methodique. *C R Acad Sci* 1919; **154**: 66-68.
34. Nishikawa, T., Edelstein, D., Du, X. L., Yamagishi, S., Matsumura, T., Kaneda, Y. Normalizing mitochondrial superoxide production blocks three pathways of hyperglycaemic damage. *Nature* 2000; **404**: 787-790.
35. Baynes, J. W., Thorpe, S. R. Role of oxidative stress in diabetic complications: a new perspective on an old paradigm. *Diabetes* 1999; **48**: 1-9.
36. Yoshida, K., Hirokawa, J., Tagami, S., Kawakami, Y., Urata, Y., Kondo, T. Weakened cellular scavenging activity against oxidative stress in diabetes mellitus: regulation of glutathione synthesis and efflux. *Diabetologia* 1995; **38**: 201-210.
37. Aronson, D., Rayfield, E. J. How hyperglycemia promotes atherosclerosis: molecular mechanisms. *Cardiovasc Diabetol* 2002; **1**: 1-7.
38. Sano, T., Umeda, F., Hashimoto, T., Nawata, H., Utsumi, H. Oxidative stress measurement by in vivo electron spin resonance spectroscopy in rats with streptozotocin-induced diabetes. *Diabetologia* 1998; **41**: 1355-1360.
39. Chen, M. S., Hutchinson, M. L., Pecoraro, R. E., Lee, W. Y., Labbe, R. F. Hyperglycemia induced intracellular depletion of ascorbic acid in human mononuclear leukocytes. *Diabetes* 1983; **32**: 1078-1081.
40. Nicolaides, A. Asymptomatic carotid stenosis and the risk of stroke (The ACSRS study): Identification of high-risk group. *Cerebrovascular Ischaemia Investigation and Management*. 1996 ; **36**: 435-441
41. Christodoulou, C.I., Pattichis, C.S., Pantziaris, M., Nicolaides, A. Texture-Based Classification of Atherosclerotic Carotid Plaques. *IEEE Transactions On Medical Imaging*. 2003; **22**: 7-12.
42. Geroulakos, G., Domjan, J., Nicolaides, A., Stevens, J., Labropoulos, N., Ramaswami, G., Belcaro, G. Ultrasonic carotid artery plaque structure and the risk of cerebral infraction on computed tomography. *J. Vasc. Surg.* 1994; **20**: 263-266.
43. El-Baroughouty, N., Geroulakos, G., Nicolaides, A., Androulakis, A., Bahal, V. Computer assisted carotid plaque characterization. *Eur. J. Vasc. Endovasc. Surg.* 1995; **9**: 548-557.

44. Iannuzzi, A., Wilcosky, T., Mercury, M., Rubba, P., Bryan, F., Bond, G. Ultrasonographic correlates of carotid atherosclerosis in transient ischemic attack and stroke. *Stroke*. 1995; **26**: 614-619.
45. Elatrozy, T., Nicolaides, A., Tegos, T., Griffin, M. The objective characterization of ultrasonic carotid plaque features. *Eur. J. Vasc. Endovasc. Surg.* 1998; **16**: 223-230.
46. Wilhjelm, J.E., Gronholdtm L.M., Wiebe, B., Jespersen, S.K., Hansen, L.K., Sillesen, H. Quantitative analysis of ultrasound B-mode images of carotid atherosclerotic plaque: Correlation with visual classification and histological examination. *IEEE Transactions On Medical Imaging*. 1998; **17**: 910-922.
47. Polak, J., Schemanski, L., O'Leary, D., Lefkowitz, D., Price, T., Savage, P., Brand, W., Reld, C. Hypoechoic plaque at US of the carotid artery: An independent risk factor for incident stroke in adults aged 65 years or older. *Radiology*. 1998; **208**: 3:649-654.
48. Lusby, R.J. et al. Transient ischemic attacks: The static and dynamic morphology of the carotid artery bifurcation. *Br. J. Surg.* 1982; **69** : s41-s44.
49. Meairs, S., Hennerici, M. Four-Dimensional Ultrasonographic Characterization of Plaque Surface Motion in Patients With Symptomatic and Asymptomatic Carotid Artery Stenosis. *Stroke*. 1999; **30** : 1807-1813.
50. Richardson, P.D., Davies, M.J., Born, G.V. Influence of plaque configuration and stress distribution on fissuring of coronary atherosclerotic plaques. *Lancet*. 1989; **2**: 941-944.
51. Loree, H.M., Kamm, R.D., Atkinson, C.M., Lee, R.T. Turbulent pressure fluctuations on surface of model vascular stenoses. *Am J Physiol*. 1991; **261**: H644-H650.
52. Cheng, G.G., Loree, H.M., Kamm, R.D., Fishbein, M.C., Lee, R.T. Distribution of circumferential stress in ruptured and stable atherosclerotic lesions: a structural analysis with histopathological correlation. *Circulation*. 1993; **97**: 1179-1187.
53. Lee, R.T., Scoen, F.J., Loree, H.M., Lark, M.W., Libby, P. Circumferential stress and matrix metalloproteinase 1 in human coronary atherosclerosis: implications for plaque rupture. *Arterioscler Thromb Vasc Biol*. 1996; **16**: 1070-1073.
54. Richardson, P.D. Physical factors affecting the atherosclerotic plaque. *In: Diagnostic Vascular Ultrasound*. Labs, K.H., Fitzgerald, D.E., Neuerburg - Heusler, D, (Eds.) London, England, Edward Arnold. pp 33-41, 1992.

55. Meairs, S., Hennerici, M. Cerebrovascular ultrasound. *In: Cerebrovascular Disease: Pathology, Diagnosis, and Management.* Ginsberg, M., Bogousslavsky, J., (Eds.) New York, NY. Blackwell. pp 1318-1336, 1998.
56. Chan, K.L. Two approaches to motion analysis of the ultrasound image sequence of carotid atheromatous plaque. *Ultrasonics*. 1993; **31**: 117-123.
57. Dufaux, F., Moscheni, F. Motion estimation techniques for digital TV: A review and a new contribution. *Proceedings of the IEEE*. 1995; **86**: 858-876.
58. Sokolis, D.P., Boudoulas, H., Karayannacos, P.E. Assessment of the aortic stress-strain relation in uniaxial tension. *J. Biomech*. 2002; **35**: 1213-1223.
59. Horn, B.K.P. Robot Vision. MIT Press, Cambridge, 1986.
60. Verri, A., Poggio, T. Against quantitative optical flow. *In: Proc. IEEE ICCV*, pp 171-180, 1987.
61. Horn, B.K.P., Schunck, B.G. Determining optical flow. *Artificial Intelligence*. 1981; **17**: 185-203.
61. Schunck, B.G. Image flow: Fundamentals and future research. *In: Proc. Of IEEE Conf. On Pattern Recognition and Image Processings*, pp 560-571, 1985.
63. Smith, S.M. Feature Based Image Sequence Understanding. *Phil.D. thesis*. Robotics Research Group, Department of Engineering Science, Oxford University, 1992.
64. Barron, J.L., Fleet, D.J., Beauchemin, S.S. Performance of Optical Flow Techniques. *IJCV*. 1994; **21**: pp.43-77
65. Kovalev, V., Petrou, M. Texture analysis in three dimensions as a cue to medical diagnosis. *In: Handbook of Medical Imaging - Processing and Analysis*. Bankman, I. N. (Ed.) New York, Academic Press, pp 231-247, 2000.
66. Leen, E.J., Feeley, T.M., Colgan, M.P., O'Malley, M.K., Moore, D.J., Hourihane, D.O., Shanik, G.D. "Haemorrhagic" carotid plaque does not contain haemorrhage. *Eur J Vase Surg*. 1990; **4**: 123-128.
67. Lennihan, L., Kupsky, W.J., Mohr, J.P., Hauser, W.A., Correll, J.W. Quest DO Lack of association between carotid plaque hematoma and ischemic cerebral symptoms. *Stroke*. 1987; **18**: 879-881.
68. Comerota, A.J., Katz, M.L, White, J.V., Grosh, J.D. The preoperative diagnosis of the ulcerated carotid atheroma. *J Vase Surg*. 1990; **11**: 505-510.

69. Robinson, M.L., Sacks, D., Perlmutter, G.S., Marinelli, D.L. Diagnostic criteria for carotid duplex sonography. *AJR Am J Roentgenol.* 1988; 151: 1045-1049.
70. Widder, B., Paulat, K., Hackspache, J., Hamann, H., Hutschenreiter, S., Kreutzer, C., Ott, F., Vollmar, J. Morphological characterization of carotid artery stenoses by ultrasound duplex scanning. *Ultrasound Med Biol* 1990; 16: 349-354.
71. Svindland, A., Torvik, A. Atherosclerotic carotid disease in asymptomatic individuals: an histological study of 53 cases. *Acta Neurol Scand* 1988; 78: 506-517.
72. Glagov, S., Bassiouny, H.S., Giddens, D.P., Zarins, C.K. Intimal thickening morphogenesis, functional significance and detection. *J Vase Invest.* 1995; 1: 1-14.
73. Hatsukami, T.S., Ferguson, M.S., Beach, K.W., Gordon, D., Detmer, P.R., Burns, D.H., Alpers, C., Strandness, D.E. Jr. Carotid plaque morphology and clinical events. *Stroke.* 1997; 28: 95-100.
74. Golemati, S., Tegos, T.J., Sassano, A., Nikita, K.S., Nicolaides, A.N. Echogenicity of B-mode ultrasound images of the carotid artery - Work-in-progress. *J Ultrasound Med.* 2004; **23**:659-669.

Παράρτημα Α

Στο συγκεκριμένο παράρτημα παρουσιάζεται ο κώδικας της εφαρμογής OFC. Κάθε αρχείο παρουσιάζεται χωριστά.

ChildView.cpp

```
#include "stdafx.h"
#include "CImage.h"
#include "ChildView.h"
#include ".\childview.h"
#include <Windows.h>
#include <stdio.h>

#using <mcorlib.dll>
using namespace System;
using namespace System::IO;
using namespace System::Text;

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// CChildView

CChildView::CChildView()
{
    down=false;
    selectROI=false;
    MarkLimits=false;
    CoordIlg=false;
    RoiSet=false;

    OKCompute=false;
    Computed=false; // Έχει τελειώσει το Compute()

    PreWittX[0][0] = -1;
    PreWittX[1][0] = 0;
    PreWittX[2][0] = 1;
    PreWittX[0][1] = -1;
    PreWittX[1][1] = 0;
    PreWittX[2][1] = 1;
    PreWittX[0][2] = -1;
    PreWittX[1][2] = 0;
    PreWittX[2][2] = 1;

    PreWittY[0][0] = -1;
    PreWittY[1][0] = -1;
    PreWittY[2][0] = -1;
    PreWittY[0][1] = 0;
    PreWittY[1][1] = 0;
    PreWittY[2][1] = 0;
    PreWittY[0][2] = 1;
    PreWittY[1][2] = 1;
```

```
PreWittY[2][2] = 1;

PreWittT[0][0] = 1;
PreWittT[1][0] = 1;
PreWittT[2][0] = 1;
PreWittT[0][1] = 1;
PreWittT[1][1] = 1;
PreWittT[2][1] = 1;
PreWittT[0][2] = 1;
PreWittT[1][2] = 1;
PreWittT[2][2] = 1;

}

CChildView::~CChildView()
{
}

BEGIN_MESSAGE_MAP(CChildView, CWnd)
    ON_WM_PAINT()
    ON_COMMAND(ID_FILE_OPENIMAGE, OnFileOpenimage)
    ON_COMMAND(ID_FILE_IMAGEINFO, OnImageInfo)
    ON_COMMAND(ID_FILE_SELECTROI, OnSelectRoi)
    ON_COMMAND(ID_PLAY_PLAYSEQUENCE, OnPlayPlaysequence)
    ON_COMMAND(ID_RUN_COMPUTE, OnRunCompute)
    ON_COMMAND(ID_RUN_MARKLIMITS, OnRunMarklimits)
    ON_COMMAND(ID_RUN_DONEMARKINGLIMITS, OnRunDonemarkinglimits)
    ON_COMMAND(ID_SHOWARROWS_VX, OnShowarrowsVx)
    ON_COMMAND(ID_SHOWARROWS_VY, OnShowarrowsVy)
    ON_COMMAND(ID_FILE_EXPORT, OnFileExport)
    ON_COMMAND(ID_Menu, OnMenu)
    ON_COMMAND(ID_SHOWARROWS_BOTH, OnShowarrowsBoth)
    ON_COMMAND(ID_RUN_MAXDIFFERENCES, OnRunMaxdifferences)

    ON_UPDATE_COMMAND_UI(ID_FILE_IMAGEINFO, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_FILE_SELECTROI, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_PLAY_PLAYSEQUENCE, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_RUN_COMPUTE, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_RUN_MARKLIMITS, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_RUN_DONEMARKINGLIMITS, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_SHOWARROWS_VX, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_SHOWARROWS_VY, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_FILE_EXPORT, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_Menu, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_SHOWARROWS_BOTH, OnUpdateImage)
    ON_UPDATE_COMMAND_UI(ID_RUN_MAXDIFFERENCES, OnUpdateImage)

    ON_WM_MOUSEMOVE()
    ON_WM_LBUTTONDOWN()
    ON_WM_LBUTTONUP()

END_MESSAGE_MAP()

// CChildView message handlers
```

```

BOOL CChildView::PreCreateWindow(CREATESTRUCT& cs)
{
    if (!CWnd::PreCreateWindow(cs))
        return FALSE;

    cs.dwExStyle |= WS_EX_CLIENTEDGE;

    cs.style &= ~WS_BORDER;

    cs.lpszClass = AfxRegisterWndClass(CS_HREDRAW|CS_VREDRAW|CS_DBLCLKS,
        ::LoadCursor(NULL, IDC_ARROW),
        reinterpret_cast<HBRUSH>(COLOR_WINDOW+1), NULL);

    return TRUE;
}

void CChildView::OnPaint()
{
    int n;
    if (!CoordDlg)
        { //Ανοίγει το παράθυρο
            // pDialog initialized to NULL in the constructor of CMyWnd class

            pDialog= new CDialog();
            // Check if new succeeded and we got a valid pointer to a dialog object
            if(pDialog != NULL)
            {
                BOOL ret = pDialog->Create(IDD_ROI,this);
                if(!ret) //Create failed.
                    AfxMessageBox("Error creating Dialog");
                pDialog->ShowWindow(SW_HIDE);
                CoordDlg=true;
            }
            else
                AfxMessageBox("Error Creating Dialog Object");

            pCoor= new CDialog();
            // Check if new succeeded and we got a valid pointer to a dialog object
            if(pCoor!= NULL)
            {
                BOOL ret = pCoor->Create(IDD_COOR,this);
                if(!ret) //Create failed.
                    AfxMessageBox("Error creating Dialog");
                pCoor->ShowWindow(SW_HIDE);
            }
            else
                AfxMessageBox("Error Creating Dialog Object");

        }
    CPaintDC dc(this); // device context for painting
    if (!imgOriginal.IsNull())
    {
        imgOriginal.StretchBlt(dc,0,0,imgOriginal.GetWidth(),imgOriginal.GetHeight(),SRCCOPY);
        for (n=0;n<LimitPointCounter;n++)
            dc.SetPixel(Vpt[n].point[0],RGB(0,255,0));
    }
}

```

```

    }

    if (MarkLimits)
    {

        CClientDC dc(this);
        CRect rectWindowSize;
        GetClientRect(rectWindowSize);
        imgOriginal.StretchBlt(dc,0,100,(XCOORD+ROISIZEX)*ZoomX,(YCOORD+ROISIZEY)*ZoomY,XCOORD,YCOORD,XCOORD+ROISIZEX,YCOORD+ROISIZEY,SRCCOPY);

    }
}

void CChildView::OnFileOpenimage(void)
{
    // TODO: Add your command handler code here
    MarkLimits=false;
    LimitPointCounter=0;
    CString file_types =
    "TIFF Files (*.tiff;*.tif)|*.tif;*.tiff;*.tif;||";
    CFileDialog dlg(TRUE,
        NULL,
        NULL,
        OFN_HIDEREADONLY | OFN_EXPLORER,
        (LPCTSTR)file_types,NULL);
    HRESULT hResult;
    dlg.m_ofn.nFilterIndex = m_nFilterLoad;
    hResult = (int)dlg.DoModal();
    if(hResult != IDOK) {
        return;
    }
    m_nFilterLoad = dlg.m_ofn.nFilterIndex;
    CString temp,filename1; // filename απλά το όνομα χωρίς το path
    filename=dlg.GetPathName();
    filename1=dlg.GetFileName();

    NumFiles=0;
    digs=0;
    filename=filename.Left(filename.GetLength()-4);
    filename1=filename1.Left(filename1.GetLength()-4);
    char bf[30];
    int i;
    int MXLength=filename1.GetLength();
    for (i=1;i<=MXLength;i++)
    {
        temp=filename1.Right(i);
        temp=temp.Left(1);
        if (CheckNum(temp)) digs++;
    }
    if (digs==0)
    {
        MessageBox("Το όνομα του αρχείου πρέπει να έχει στο τέλος αριθμούς","ERROR!");
        OnFileOpenimage();
    }

    temp=filename1.Right(digs);
    filename=filename.Left(filename.GetLength()-digs);

```

```

        digs=atoi(temp); // αύξων αριθμός εικόνας
        int nextdig;
        nextdig=digs;

        do //υπολόγισε μέχρι ποιον αριθμό φθάνει η αρίθμηση των εικόνων
        {
            NumFiles++;
            sprintf(bf,"%i.tif",++nextdig);
            temp=fname+bf;
        } while (File::Exists(temp));

        nextdig--; //αριθμός τελικής εικόνας
        if (nextdig==digs)
        {
            MessageBox("Πρέπει να υπάρχουν τουλάχιστον 2 αρχεία!!", "ERROR!");
            OnFileOpenimage();
        }

        sprintf(bf,"%i.tif",digs);
        temp=fname+bf;
        imgOriginal.Destroy();
        imgOriginal.Load(temp);
        Invalidate();
        UpdateWindow();

        char buf[300];
        sprintf(bf,"%i.tif",nextdig);
        sprintf(buf,"Το αρχικό frame είναι το %s ενώ το τελικό είναι το %s",temp,fname+bf);
        MessageBox(buf,"INFO!");
        UpdateWindow();
        width=imgOriginal.GetWidth();
        height=imgOriginal.GetHeight();
        OKCompute=false;
    }

void CChildView::OnImageInfo()
{
    OutPutFile();
    Invalidate();
    UpdateWindow();
}

void CChildView::OnSelectRoi()
{
    if (!selectROI)
    {
        selectROI=true;
        LimitPointCounter=0;
        pDialog->ShowWindow(SW_SHOW);
    }
}

void CChildView::OnUpdateImage(CCcmdUI *pCmdUI)
{
    pCmdUI->Enable(!imgOriginal.IsNull());
}

void CChildView::Compute(void)
{

```

```

int i=0;
char buffer[100];
CString temp;
pProg = new CDialog;

//Ανοίγει το παράθυρο
if(pProg != NULL)
{
    BOOL ret = pProg ->Create(IDD_PROG,this);
    if(!ret) //Create failed.
        AfxMessageBox("Error creating Dialog");
        pProg ->ShowWindow(SW_SHOW);
}
else
    AfxMessageBox("Error Creating Dialog Object");
    CStatic* pInfo;
    sprintf(buffer,"0");
    pInfo = (CStatic*) pProg ->GetDlgItem(IDC_PROG);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    CProgressCtrl myCtrl;

    // Create a child progress control.
    myCtrl.Create(WM_CHILD|WS_VISIBLE|PBS_SMOOTH|WS_BORDER,
    CRect(15,15,210,40), pProg, IDD_PROG);

    myCtrl.SetRange(0,NumFiles);
    myCtrl.SetStep(1);

    //1o frame
    sprintf(buffer,"%i.tif",digs);
    temp=fname+buffer;
    imgOriginal.Destroy();
    imgOriginal.Load(temp);

    MarkROI();
    FirstFrame(imgOriginal);
    sprintf(buffer,"%i",myCtrl.GetPos()*100/NumFiles);

    //τύπωσε το %
    pInfo = (CStatic*) pProg ->GetDlgItem(IDC_PROG);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    //τύπωσε το όνομα του αρχείου
    sprintf(buffer,"%s",temp);
    pInfo = (CStatic*) pProg ->GetDlgItem(IDC_FNAME);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    pProg ->UpdateWindow();

    for (i=0;i<NumFiles;i++)
    {
        myCtrl.StepIt();
        sprintf(buffer,"%i",myCtrl.GetPos()*100/NumFiles);

        //τύπωσε το %
        pInfo = (CStatic*) pProg ->GetDlgItem(IDC_PROG);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
        sprintf(buffer,"%s",temp);

        //τύπωσε το όνομα του αρχείου

```

```

        pInfo = (CStatic*) pProg ->GetDlgItem(IDC_FNAME);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
        pProg ->UpdateWindow();

        sprintf(buffer,"%i.tif",digs+i);
        temp=fname+buffer;
        imgOriginal.Destroy();
        imgOriginal.Load(temp);

        NextFrame(imgOriginal,i); // Υπολογισμός ταχύτητας

        FirstFrame(imgOriginal); // Λήψη ως frame αναφοράς για την επόμενη ταχύτητα
    }

    //Ξαναφόρτωσε το 1ο frame ώστε να εμφανιστεί εαν γίνει UpdateWindow()
    sprintf(buffer,"%i.tif",digs);
    temp=fname+buffer;
    imgOriginal.Destroy();
    imgOriginal.Load(temp);
    MarkROI();
    pProg ->DestroyWindow();
    delete pProg ;

    NormVelocities();
    MessageBox("DONE","!");
}

void CChildView::FirstFrame(CImage &image)
{
    CImage* m_pImage = &image;
    int i,j,temp;

    for (i=XCOORD;i<XCOORD+ROISIZE_X;i++)
        for (j=YCOORD;j<YCOORD+ROISIZE_Y;j++)
        {
            if ((temp=m_pImage->GetPixel(i,j))==CLR_INVALID)

            {MessageBox("CLR_INVALID","CLR_INVALID First Image");
              break;
            }

            df[i-XCOORD][j-YCOORD] = GetRValue(temp);
        }
    }
}

void CChildView::MarkROI(void)
{
    // εαν έχουμε φτάσει στα άκρα
    if (XCOORD+ROISIZE_X>=width-1) ROISIZE_X=width-XCOORD-2;
    if (YCOORD+ROISIZE_Y>=height-1) ROISIZE_Y=height-YCOORD-2;
    int i,j;

    i=XCOORD-1;
    for (j=YCOORD-1;j<YCOORD+ROISIZE_Y+1;j++)
    {
        imgOriginal.SetPixelRGB(i,j,255,255,255);
    }
}

```

```

        i=XCOOR+ROISIZEX+1;
        for (j=YCOOR-1;j<=YCOOR+ROISIZEY+1;j++)
        {
            imgOriginal.SetPixelRGB(i,j,255,255,255);
        }
        j=YCOOR-1;
        for (i=XCOOR-1;i<XCOOR+ROISIZEX+1;i++)
        {
            imgOriginal.SetPixelRGB(i,j,255,255,255);
        }
        j=YCOOR+ROISIZEY+1;
        for (i=XCOOR-1;i<=XCOOR+ROISIZEX+1;i++)
        {
            imgOriginal.SetPixelRGB(i,j,255,255,255);
        }
    }

void CChildView::NextFrame(CImage &image,int frame)
{
    CImage* m_pImage = &image;
    int i,j,temp;
    for (i=XCOOR;i<XCOOR+ROISIZEX;i++)
        for (j=YCOOR;j<YCOOR+ROISIZEY;j++)
        {

            {if ((temp=m_pImage->GetPixel(i,j))==CLR_INVALID)

                {MessageBox("CLR_INVALID","CLR_INVALID Next Image");
                    break;

                }

            }

            dtemp[i-XCOOR][j-YCOOR] = GetRValue(temp);
            dt[i-XCOOR][j-YCOOR] = dtemp[i-XCOOR][j-YCOOR] -df[i-XCOOR][j-YCOOR] ;
            //προσθέτω τα 2 Frames
            dtemp[i-XCOOR][j-YCOOR]+=df[i-XCOOR][j-YCOOR] ;
        };

    //Επεξεργασία
    //Υπολογισμός θf/θx και θf/θy
    int m,n;
    for(i=0;i<ROISIZEX;i++)
        for(j=0;j<ROISIZEY;j++)
        {
            dy[i][j]=0.0;
            dx[i][j]=0.0;
            df[i][j]=0.0;
        }

    //Τα σημεία της περιμέτρου - συνέλιξη

    i=XCOOR;//πάνω πλευρά
    for(j=YCOOR;j<YCOOR+ROISIZEY;j++)
        for (m=-1;m<=1;m++)
            for (n=-1;n<=1;n++)
            {
                if ((i-XCOOR+m>=0)&(i-
XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
                {

```

```

        dy[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-
YCOOR+n]*PreWittY[1+m][1+n];
        dx[i-XCOOR][j-YCOOR]+=dtemp[i-
XCOOR+m][j-YCOOR+n]*PreWittX[1+m][1+n];
        df[i-XCOOR][j-YCOOR]+=dt[i-
XCOOR+m][j-YCOOR+n]*PreWittT[1+m][1+n];
    }
}

i=XCOOR+ROISIZEX-1;//κάτω πλευρά
for(j=YCOOR;j<YCOOR+ROISIZEY;j++)
    for (m=-1;m<=1;m++)
        for (n=-1;n<=1;n++)
        {
if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
        {
dy[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittY[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittX[1+m][1+n];
df[i-XCOOR][j-YCOOR]+=dt[i-XCOOR+m][j-YCOOR+n]*PreWittT[1+m][1+n];
        }
        }

j=YCOOR;//αριστερή πλευρά
for(i=XCOOR;i<XCOOR+ROISIZEX;i++)
    for (m=-1;m<=1;m++)
        for (n=-1;n<=1;n++)
        {
if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
        {
dy[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittY[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittX[1+m][1+n];
df[i-XCOOR][j-YCOOR]+=dt[i-XCOOR+m][j-YCOOR+n]*PreWittT[1+m][1+n];
        }
        }

j=YCOOR+ROISIZEY-1;//δεξιά πλευρά
for(i=XCOOR;i<XCOOR+ROISIZEX;i++)
    for (m=-1;m<=1;m++)
        for (n=-1;n<=1;n++)
        {
if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
        {
dy[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittY[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittX[1+m][1+n];
df[i-XCOOR][j-YCOOR]+=dt[i-XCOOR+m][j-YCOOR+n]*PreWittT[1+m][1+n];
        }
        }

//Τα κεντρικά σημεία εκτός από την περίμετρο - συνέλιξη
for(i=XCOOR+1;i<XCOOR+ROISIZEX-1;i++)
    for(j=YCOOR+1;j<YCOOR+ROISIZEY-1;j++)
        for (m=-1;m<=1;m++)
            for (n=-1;n<=1;n++)
            {
dy[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittY[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=dtemp[i-XCOOR+m][j-YCOOR+n]*PreWittX[1+m][1+n];

```

```

df[i-XCOOR][j-YCOOR]+=dt[i-XCOOR+m][j-YCOOR+n]*PreWittT[1+m][1+n];

        }
    for(i=0;i<ROISIZEX;i++)
        for(j=0;j<ROISIZEY;j++)
        {
            dy[i][j]/=12.0;
            dx[i][j]/=12.0;
            df[i][j]/=9.0;
        }

    HornSchunk(frame);

}
int CheckNum(CString inp)
{
    if ((inp.Compare("0"))==0) return 1 ;
    if ((inp.Compare("1"))==0) return 1 ;
    if ((inp.Compare("2"))==0) return 1 ;
    if ((inp.Compare("3"))==0) return 1 ;
    if ((inp.Compare("4"))==0) return 1 ;
    if ((inp.Compare("5"))==0) return 1 ;
    if ((inp.Compare("6"))==0) return 1 ;
    if ((inp.Compare("7"))==0) return 1 ;
    if ((inp.Compare("8"))==0) return 1 ;
    if ((inp.Compare("9"))==0) return 1 ;
    return 0;
}

void CChildView::OnPlayPlaysequence()
{
    // TODO: Add your command handler code here
    int i,n;
    char bf[30];
    CClientDC dc(this);

    for (i=digs;i<digs+NumFiles;i++)
    {
        sprintf(bf,"%i.tif",i);
        imgOriginal.Destroy();
        imgOriginal.Load(fname+bf);
        MarkROI();
        imgOriginal.StretchBlt(dc,0,0,width,height,SRCCOPY);
        for (n=0;n<LimitPointCounter;n++)
            dc.SetPixel(Vpt[n].point[i-digs],RGB(0,255,0));
        Sleep(40); //40ms = 1/(25 frames/sec)
    }

    sprintf(bf,"%i.tif",digs);
    imgOriginal.Destroy();
    imgOriginal.Load(fname+bf);
    MarkROI();

    pDialog->ShowWindow(SW_HIDE);
    pCoor->ShowWindow(SW_HIDE);
    imgOriginal.StretchBlt(dc,0,0,width,height,SRCCOPY);
    for (n=0;n<LimitPointCounter;n++)
        dc.SetPixel(Vpt[n].point[0],RGB(0,255,0));
    pDialog->ShowWindow(SW_SHOW);
}

```

```

        pCoor->ShowWindow(SW_SHOW);
    }

void CChildView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    char bf[20];
    sprintf(bf,"%i.tif",digs);
    CString temp=fname+bf;
    CStatic* pInfo;
    char buffer[10];
    pInfo = (CStatic*) pDialog->GetDlgItem(IDC_XCOOR);
    if (MarkLimits) sprintf(buffer,"%i",XCOOR+(point.x-point.x%ZoomX)/ZoomX);
    //Μετατροπή για τιμή zoomed
    else sprintf(buffer,"%i",point.x);

    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    pInfo = (CStatic*) pDialog->GetDlgItem(IDC_YCOOR);
    if (MarkLimits) sprintf(buffer,"%i",YCOOR+((point.y-100)-(point.y-100)%ZoomY)/ZoomY) ;
    //Μετατροπή για τιμή zoomed
    else sprintf(buffer,"%i",point.y);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    if ((selectROI)&(!RoiSet)&(CoorDlg)&(!down))
    {
        CStatic* pInfo;
        char buffer[10];
        pInfo = (CStatic*) pDialog->GetDlgItem(IDC_XCOOR);
        sprintf(buffer,"%i",point.x);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
        pInfo = (CStatic*) pDialog->GetDlgItem(IDC_YCOOR);
        sprintf(buffer,"%i",point.y);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    }
    if (((nFlags & MK_LBUTTON) == MK_LBUTTON)&(selectROI)&(down)&(point!=oldpoint))
    {
        oldpoint=point;

        imgOriginal.Destroy();
        imgOriginal.Load(temp);

        XCOOR=min(start.x,point.x);
        YCOOR=min(start.y,point.y);
        ROISIZEX=abs(point.x-start.x);
        ROISIZEY=abs(point.y-start.y);

        MarkROI();
        imgOriginal.StretchBlt(dc,0,0,width,height,SRCCOPY);
        CStatic* pInfo;
        char buffer[10];
        pInfo = (CStatic*) pDialog->GetDlgItem(IDC_XCOOR);
        sprintf(buffer,"%i",point.x);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
        pInfo = (CStatic*) pDialog->GetDlgItem(IDC_YCOOR);
        sprintf(buffer,"%i",point.y);
        pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    }

    if ((MarkLimits)&((nFlags & MK_LBUTTON) == MK_LBUTTON))
    {

```

```

        if
        ((point.x<(ROISIZEX)*ZoomX)&(point.y<100+ROISIZEY*ZoomY)&(point.y>100))
        {
            int k,i,m,j,MidY,b,l;
            k=0;
            for (i=0;i<=LimitPointCounter;i++)
                if (Vpt[i].point[0]==point) k++;

            if (k==0)          //Νέο Σημείο
            {

                Vpt[LimitPointCounter].point[0]=point;
                if (LimitPointCounter++>LimitPointCounterMAXSIZE)
                {
                    MessageBox("Δεν μπορείτε να διαλέξετε άλλο σημείο!1","!");
                    goto out;
                }

                CPen lPen(PS_SOLID, 1, RGB(0, 255, 0));
                dc.SelectObject(&lPen);
                for (m=-2;m<3;m++)
                {

                    dc.MoveTo(MSPrevX+m,MSPrevY+m);

                    dc.LineTo(point.x+m,point.y+m);

                }

                if (MSPrevX!=point.x)
                {

                    l=(MSPrevY-point.y)/(MSPrevX-point.x);
                    b=MSPrevY-l*MSPrevX;

                    int a1=min(MSPrevX,point.x);
                    int a2=max(MSPrevX,point.x);

                    for (j=a1;j<=a2;j++)
                    {
                        MidY=l*j+b;
                        k=0;
                        for (i=0;i<=LimitPointCounter;i++)
                            if ((Vpt[i].point[0].x==j)&(Vpt[0].point[0].y==MidY)) k++;
                        if (k==0)          //Νέο Σημείο
                        {
                            Vpt[LimitPointCounter].point[0].x=j;

                            Vpt[LimitPointCounter].point[0].y=MidY;

                            if
                            (LimitPointCounter++>LimitPointCounterMAXSIZE)
                            {
                                MessageBox("Δεν μπορείτε να
                                διαλέξετε άλλο σημείο!2","!");
                                goto out;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

else
{
    int a1=min(MSPrevY,point.y);
    int a2=max(MSPrevY,point.y);
    for (j=a1;j<=a2;j++)
    {
        k=0;
        for (i=0;i<=LimitPointCounter;i++)
        if ((Vpt[i].point[0].x==point.x)&(Vpt[i].point[0].y==j)) k++;
        if (k==0) //Νέο Σημείο
        {
            Vpt[LimitPointCounter].point[0].x=point.x;
            Vpt[LimitPointCounter].point[0].y=j;
            if (LimitPointCounter++>LimitPointCounterMAXSIZE)
            {
                MessageBox("Δεν μπορείτε να διαλέξετε άλλο σημείο!3","!");
                goto out;
            }
        }
    }

    MSPrevX=point.x;
    MSPrevY=point.y;
}

}

out:
    CWnd::OnMouseMove(nFlags, point);
}

void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    //Για το MarkRoi στη μεγέθυνση
    MSPrevX=point.x;
    MSPrevY=point.y;

    if ((!down)&(selectROI))
    {
        start=point;
        down=true;
    }

    CWnd::OnLButtonDown(nFlags, point);
}

void CChildView::OnLButtonUp(UINT nFlags, CPoint point)
{
    if ((down)&(selectROI))
    {
        XCOOR=min(start.x,point.x);
        YCOOR=min(start.y,point.y);
        ROISIZEX=abs(point.x-start.x);
        ROISIZEY=abs(point.y-start.y);

        if ((ROISIZEX<MAX_ROISIZE)&(ROISIZEY<MAX_ROISIZE))
        {

```

```

        OKCompute=true;
        RoiSet=true;
        selectROI=false;
    }

    else
    {
        OKCompute=false;
        selectROI=true;
        MessageBox("ROISIZE > MAX ROISIZE", "!");
        CClientDC dc(this);
        char bf[20];
        CString temp;

        //Ξαναφόρτωσε το 1ο frame ώστε να εμφανιστέ εαν γίνει UpdateWindow()
        sprintf(bf,"%i.tif",digs);
        temp=fname+bf;
        imgOriginal.Destroy();
        imgOriginal.Load(temp);
        imgOriginal.StretchBlt(dc,0,0,width,height,SRCCOPY);
        pDialog->ShowWindow(SW_HIDE);
        pDialog->ShowWindow(SW_SHOWNOACTIVATE);
    }
    down=false;
    CStatic* pInfo;
    char buffer[30];
    pInfo = (CStatic*) pDialog->GetDlgItem(IDC_XCOORD);
    sprintf(buffer,"%i",point.x);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    pInfo = (CStatic*) pDialog->GetDlgItem(IDC_YCOORD);
    sprintf(buffer,"%i",point.y);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
    pDialog->ShowWindow(SW_SHOW);
    pCoor->ShowWindow(SW_SHOW);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_TopLeft);
    sprintf(buffer,"%i,%i",XCOORD,YCOORD);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_TopRight);
    sprintf(buffer,"%i,%i",XCOORD+ROISIZE_X,YCOORD);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_BottomLeft);
    sprintf(buffer,"%i,%i",XCOORD,YCOORD+ROISIZE_Y);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_BottomRight);
    sprintf(buffer,"%i,%i",XCOORD+ROISIZE_X,YCOORD+ROISIZE_Y);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);
}

CWnd::OnLButtonUp(nFlags, point);
}

void CChildView::OnRunCompute()
{
    // TODO: Add your command handler code here
    if (OKCompute)
    {
        selectROI=false;
    }
}

```

```

        OKCompute=false;
        MarkROI();
        Compute();
        Computed=true;
    }

    else MessageBox("NOT ready for compute - Select ROI First!", "!");
}

int CChildView::HornSchunk(int frame)
{
    int i,j;
    double VxAV, VyAV, PD, OldVx, OldVy, Err;
    int m,n;
    int mMin,mMax,nMin,nMax;

    int k=0;

    while (k<1000)
    {
        Err=0.0;
        for (i=0;i<ROISIZEX-1;i++)
            for (j=0;j<ROI SIZEY-1;j++)
            {
                //Υπολογισμός μέσου όρου μόνο από τα γειτονικά Pixels
                VxAV=0.0;
                VyAV=0.0;

                mMin=max(i-1,0);
                mMax=min(i+2,ROISIZEX);

                nMin=max(j-1,0);
                nMax=min(j+2,ROI SIZEY);

                for (m=mMin;m<mMax;m++)
                    for (n=nMin;n<nMax;n++)
                    {
                        VxAV+=Vx[m][n][frame];
                        VyAV+=Vy[m][n][frame];
                    }
                VxAV=(double) VxAV/((mMax-mMin)*(nMax-nMin));
                VyAV=(double) VyAV/((mMax-mMin)*(nMax-nMin));

                //LAMDA = 1
                PD=(dx[i][j]*VxAV + dy[i][j]*VyAV +df[i][j])/(1.0+(dx[i][j]*dx[i][j]) + (dy[i][j]*dy[i][j]));

                OldVx=Vx[i][j][frame];
                OldVy=Vy[i][j][frame];
                Vx[i][j][frame]=VxAV-dx[i][j]*PD;
                Vy[i][j][frame]=VyAV-dy[i][j]*PD;

                Err+=(OldVx-Vx[i][j][frame])*(OldVx-Vx[i][j][frame])+(OldVy-Vy[i][j][frame])*(OldVy-
                Vy[i][j][frame]);

                if (Err<0.00001) return 0;
            }

        k++;
    }

    return 0;
}

```

```

}

void CChildView::OnRunMarklimits()
{
    if (Computed)
    {
        CClientDC dc(this);
        imgOriginal.StretchBlt(dc,0,100,(XCOOR+ROISIZEX)*ZoomX,(YCOOR+ROISIZEY)*ZoomY,XCOOR,YCOOR,XCOOR+ROISIZEX,YCOOR+ROISIZEY,SRCCOPY);
        dc.Rectangle(0,0,width,100); //άσπρο "κάλυμα"

        MarkLimits=true;
        LimitPointCounter=0;

    }
    else MessageBox("Διάλεξε ROI και κάνε Compute πρώτα","!");
}

void CChildView::OnRunDonemarkinglimits()
{ if (MarkLimits)
{
    MarkLimits=false;
    int k,i,j;
    CPoint PointTemp[LimitPointCounterMAXSIZE];
    CClientDC dc(this);
    Invalidate();
    UpdateWindow();

    imgOriginal.StretchBlt(dc,0,0,width,height,SRCCOPY);

    for (i=0;i<LimitPointCounter;i++)//Μετατροπή από συντεταγμένες Zoom σε κανονικές
    {
        Vpt[i].point[0].y=(Vpt[i].point[0].y-100)/ZoomY+YCOOR;
        Vpt[i].point[0].x=(Vpt[i].point[0].x/ZoomX+XCOOR);
    }

    int PointCounterTemp=0; //Απαλοιφή "διπλών" σημείων
    for (i=0;i<LimitPointCounter;i++)
    {
        k=0;
        for (j=0;j<PointCounterTemp;j++)
            if (Vpt[i].point[0]==PointTemp[j]) k++;
        if (k==0)
        {
            PointTemp[PointCounterTemp]=Vpt[i].point[0];
            PointCounterTemp++;
        }
    }
    LimitPointCounter=PointCounterTemp;
    for (i=0;i<LimitPointCounter;i++)
    {
        Vpt[i].point[0]=PointTemp[i];
    }

    for (i=0;i<LimitPointCounter;i++)
        dc.SetPixel(Vpt[i].point[0],RGB(0,255,0));
}
}

```

```

CompPoints();
MaxVelocity();
MaxMaxVelocity();
SortPoints();
MaxSuccessiveDiff();

for (i=0;i<LimitPointCounter;i++)
{
    VptNorm[i].point[0]=Vpt[i].point[0];
    VptNorm[i].VmaxX=Vpt[i].VmaxX/NormValue ;
    VptNorm[i].VmaxY=Vpt[i].VmaxY/NormValue ;
}
}
else MessageBox("Επιλέξτε τα όρια με το \"Mark Limits\" και μετα επιλεξτε το DONE","!");
}

void CChildView::ShowArrows(bool V)
{
    //V==0 -> Vx
    //V==1 -> Vy
    CClientDC dc(this);

    int x0,y0,ArrowLength,ArrowNose,ArrowCone,xOld,yOld,i;

    //Βέλη
    for (i=0;i<LimitPointCounter;i++)
    {
        x0=VptNorm[i].point[0].x;
        y0=VptNorm[i].point[0].y;
        xOld=-50;
        yOld=-50;

        if ((abs(x0-xOld)>50)&V)
        {
            //Vy
            ArrowLength=-40*VptNorm[i].VmaxY;
            ArrowNose=0.7*ArrowLength;
            ArrowCone=0.3*ArrowLength;

            CPen lpen1(PS_SOLID, 1, RGB(255,0,0));
            dc.SelectObject(&lpen1);

            dc.MoveTo(x0,y0);
            dc.LineTo(x0,y0-ArrowLength);

            dc.LineTo(x0-ArrowCone,y0-ArrowNose);

            dc.MoveTo(x0,y0-ArrowLength);
            dc.LineTo(x0+ArrowCone,y0-ArrowNose);
            xOld=x0;
        }

        if ((abs(y0-yOld)>50)&(!V))
        {
            //Vx
            ArrowLength=-40*VptNorm[i].VmaxX;
            ArrowNose=0.7*ArrowLength;
            ArrowCone=0.3*ArrowLength;

```

```

        CPen lpen2(PS_SOLID, 1, RGB(0,0,255));
        dc.SelectObject(&lpen2);

        dc.MoveTo(x0,y0);
        dc.LineTo(x0-ArrowLength,y0);

        dc.LineTo(x0-ArrowNose,y0-ArrowCone);

        dc.MoveTo(x0-ArrowLength,y0);
        dc.LineTo(x0-ArrowNose,y0+ArrowCone);

        yOld=y0;
    }
}
}
void CChildView::CompPoints(void)//Υπολογισμός της νέας θέσης κάθε Pixel
{
    int i,k,n,x,y;//,x1,x0,y1,y0;
    double Sx,Sy;
    for (n=0;n<LimitPointCounter;n++) // n== α/α σημείων
    for (k=1;k<NumFiles;k++) //κ== ο αριθμός του frame
    {
        Sx=0;
        Sy=0;
        for (i=0;i<k;i++) //i== τα frame πριν το k
        {
            x=Vpt[n].point[i].x-XCOORD;
            y=Vpt[n].point[i].y-YCOORD;

            Sx+=Vx[x][y][i];
            Sy+=Vy[x][y][i];

        }
        Vpt[n].point[k].x=Vpt[n].point[0].x+Sx; //+(int)Sx;
        Vpt[n].point[k].y=Vpt[n].point[0].y+Sy; //+(int)Sy;
    }
}
void CChildView::OutPutFile(void)
{
    char buffer[100];
    int i,k,n,j;
    HRESULT hResult1;
    CString file_types =
    "Text Files(*.txt)*.txt";

    CFileDialog dlg(FALSE,NULL,NULL,OFN_HIDEREADONLY |
    OFN_OVERWRITEPROMPT | OFN_EXPLORER,(LPCTSTR)file_types,NULL);

    dlg.m_ofn.nFilterIndex = m_nFilterLoad;
    hResult1 = (int)dlg.DoModal();
    if (hResult1 != IDOK) {
        return;
    }

    // Add the appropriate extension if the user didn't type one

    CString strExtension,strFileName;
    strFileName = dlg.m_ofn.lpstrFile;

```

```

        // add the file extension if the user didn't supply one
        if (dlg.m_ofn.nFileExtension == 0)
        {
            strExtension = "txt";
            strFileName = strFileName + '.' + strExtension;
        }
        if (FAILED(hResult1)) {
            CString fmt;
            fmt.Format("Output text save failed:\n%x - %s", hResult1,
_com_error(hResult1).ErrorMessage());
            ::AfxMessageBox(fmt);
            return;
        }

        Invalidate();
        UpdateWindow();
        FileStream* fs = File::Create(strFileName);

        sprintf (buffer,"Filename %sXX.tif\n\n",fname);
        Byte info[] = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROI COOR: TopLeft:(%i,%i)\n",XCOOR,YCOOR);
        info= (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROI COOR: TopRight:(%i,%i)\n",XCOOR+ROISIZEX,YCOOR);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROI COOR: BottomLeft:(%i,%i)\n",XCOOR,YCOOR+ROISIZEY);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROI COOR:
BottomRight:(%i,%i)\n",XCOOR+ROISIZEX,YCOOR+ROISIZEY);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"\n");
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf(buffer,"Max Vx is %.8g at point (%i,%i) at frame %i (time %g
sec)\n",MaxMax.Vx*FrameRate/ImageDensity,MaxMax.x.x,MaxMax.x.y,MaxMax.framex,MaxMax.f
ramex*0.04);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf(buffer,"Max Vy is %.8g at point (%i,%i) at frame %i (time %g
sec)\n",MaxMax.Vy*FrameRate/ImageDensity,MaxMax.y.x,MaxMax.y.y,MaxMax.framey,MaxMax.f
ramex*0.04);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf(buffer,"\n");
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        for (i=0;i<LimitPointCounter;i+=SEARCHAREA)

```

```

        {
            if (difax[i/SEARCHAREA-1].V!=0)
            {
                sprintf(buffer,"Max relative Vx is %.8g between points
(%%i,%%i) and (%%i,%%i)",difax[i/SEARCHAREA-1].V*FrameRate/ImageDensity,difax[i/SEARCHAREA-
1].point1.x,difax[i/SEARCHAREA-1].point1.y,difax[i/SEARCHAREA-
1].point2.x,difax[i/SEARCHAREA-1].point2.y);
                info = (new UTF8Encoding(true))->GetBytes(buffer);
                fs->Write(info, 0, info->Length);

                sprintf(buffer,"with velocities %.8g and %.8g at frame %%i
(time %%g sec)\n",difax[i/SEARCHAREA-1].V1*FrameRate/ImageDensity,difax[i/SEARCHAREA-
1].V2*FrameRate/ImageDensity,difax[i/SEARCHAREA-1].frame,difax[i/SEARCHAREA-
1].frame*0.04);
                info = (new UTF8Encoding(true))->GetBytes(buffer);
                fs->Write(info, 0, info->Length);
            }
            if (dify[i/SEARCHAREA-1].V!=0)
            {
                sprintf(buffer,"Max relative Vy is %.8g between points
(%%i,%%i) and (%%i,%%i)",dify[i/SEARCHAREA-1].V*FrameRate/ImageDensity,dify[i/SEARCHAREA-
1].point1.x,dify[i/SEARCHAREA-1].point1.y,dify[i/SEARCHAREA-
1].point2.x,dify[i/SEARCHAREA-1].point2.y);
                info = (new UTF8Encoding(true))->GetBytes(buffer);
                fs->Write(info, 0, info->Length);

                sprintf(buffer,"with velocities %.8g and %.8g at frame %%i
(time %%g sec)\n",dify[i/SEARCHAREA-1].V1*FrameRate/ImageDensity,dify[i/SEARCHAREA-
1].V2*FrameRate/ImageDensity,dify[i/SEARCHAREA-1].frame,dify[i/SEARCHAREA-
1].frame*0.04);
                info = (new UTF8Encoding(true))->GetBytes(buffer);
                fs->Write(info, 0, info->Length);
            }
        }

        sprintf(buffer,"\n");
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf(buffer,"Max Successive Vx is %.8g at point (%%i,%%i) in frame
%%i\n",SuccessiveMaxX.V*FrameRate/ImageDensity,SuccessiveMaxX.point.x,SuccessiveMaxX.point.
y,SuccessiveMaxX.frame);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf(buffer,"Max Successive Vy is %.8g at point (%%i,%%i) in frame
%%i\n",SuccessiveMaxY.V*FrameRate/ImageDensity,SuccessiveMaxY.point.x,SuccessiveMaxY.point.
y,SuccessiveMaxY.frame);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        for (n=0;n<LimitPointCounter;n++)
        {
            sprintf (buffer,"Points of Interest(%%i,%%i) me VxMAX:%.8g kai
VyMAX:%.8g\n",Vpt[n].point[0].x,Vpt[n].point[0].y,Vpt[n].VmaxX*FrameRate/ImageDensity,Vpt[n]
.VmaxY*FrameRate/ImageDensity);
            info = (new UTF8Encoding(true))->GetBytes(buffer);
            fs->Write(info, 0, info->Length);
        }
    }

```

```

        sprintf (buffer,"x\ty\tVx \tVy \n");// \tV \ttheta \n");
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        for (n=0;n<LimitPointCounter;n++)
        {
            for (k=0;k<NumFiles;k++)

                i=Vpt[n].point[k].x-XCOOR;
                j=Vpt[n].point[k].y-YCOOR;

                //σε cm/sec
                sprintf
(buffer,"%i\t%i\t%.8g\t%.8g\n",i+XCOOR,j+YCOOR,Vx[i][j][k]*FrameRate/ImageDensity,Vy[i][j][k]
]*FrameRate/ImageDensity);
                info = (new UTF8Encoding(true))-
>GetBytes(buffer);
                fs->Write(info, 0, info->Length);
            }
            sprintf (buffer,"\n");
            info = (new UTF8Encoding(true))->GetBytes(buffer);
            fs->Write(info, 0, info->Length);
        }

        fs->Close();
        CString SetOutFile =strFileName;
        SetOutFile=SetOutFile.Left(SetOutFile.GetLength()-4);
        SetOutFile=SetOutFile+"_settings.txt";

        //from here
        fs = File::Create(SetOutFile);

        sprintf (buffer,"XCOOR,YCOOR: (%i,%i)\n",XCOOR,YCOOR);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROISIZEX,ROISIZEY: (%i,%i)\n",ROISIZEX,ROISIZEY);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"LimitPointCounter: %i\n",LimitPointCounter);
        info= (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        for (i=0;i<LimitPointCounter;i++)
        {
            sprintf (buffer,"VptPoint:
(%i,%i)\n",Vpt[i].point[0].x,Vpt[i].point[0].y);
            info= (new UTF8Encoding(true))->GetBytes(buffer);
            fs->Write(info, 0, info->Length);
        }
        sprintf(buffer,"Τα settings αποθηκεύτηκαν στο %s",SetOutFile);
        MessageBox(buffer,"!");

        fs->Close();

    }
    void CChildView::OnShowarrowsVx()
    {

```

```

        // TODO: Add your command handler code here
        Invalidate();
        UpdateWindow();
        ShowArrows(0);
        int n;
        CClientDC dc(this);
        for (n=0;n<LimitPointCounter;n++)
        dc.SetPixel(Vpt[n].point[0],RGB(0,255,0));
    }

void CChildView::OnShowarrowsVy()
{
    // TODO: Add your command handler code here
    Invalidate();
    UpdateWindow();
    ShowArrows(1);
    CClientDC dc(this);
    int n;
    for (n=0;n<LimitPointCounter;n++)
    dc.SetPixel(Vpt[n].point[0],RGB(0,255,0));
}

void CChildView::OnShowarrowsBoth()
{
    // TODO: Add your command handler code here
    Invalidate();
    UpdateWindow();
    ShowArrows(1);
    ShowArrows(0);
    int n;
    CClientDC dc(this);
    for (n=0;n<LimitPointCounter;n++)
    dc.SetPixel(Vpt[n].point[0],RGB(0,255,0));
}

int CChildView::MaxVelocity(void)
{
    int i,k,m,n;
    double MaxValueX,MinValueX,MaxValueY,MinValueY;
    for (i=0;i<LimitPointCounter;i++)
    {
        MaxValueX=0;
        MinValueX=1000;
        MaxValueY=0;
        MinValueY=1000;
        for (k=0;k<NumFiles;k++)
        {
            m=Vpt[i].point[k].x-XCOORD;
            n=Vpt[i].point[k].y-YCOORD;

            MaxValueX=max(Vx[m][n][k],MaxValueX);
            MinValueX=min(Vx[m][n][k],MinValueX);
            MaxValueY=max(Vy[m][n][k],MaxValueY);
            MinValueY=min(Vy[m][n][k],MinValueY);
        }

        if (abs(MinValueX)>MaxValueX) Vpt[i].VmaxX=MinValueX;
        else Vpt[i].VmaxX=MaxValueX;

        if (abs(MinValueY)>MaxValueY) Vpt[i].VmaxY=MinValueY;
    }
}

```

```

        else Vpt[i].VmaxY=MaxValueY;

        for (k=0;k<NumFiles;k++)
        {
            m=Vpt[i].point[k].x-XCOOR;
            n=Vpt[i].point[k].y-YCOOR;
            if (Vpt[i].VmaxX==Vx[m][n][k]) Vpt[i].framex=k;
            if (Vpt[i].VmaxY==Vy[m][n][k]) Vpt[i].framey=k;
        }
    }
    return 0;
}

void CChildView::MaxMaxVelocity(void)
{
    int i;
    double MaxValueX,MinValueX,MaxValueY,MinValueY;
    MaxValueX=0;
    MinValueX=1000;
    MaxValueY=0;
    MinValueY=1000;

    for (i=0;i<LimitPointCounter;i++)
    {
        MaxValueX=max(Vpt[i].VmaxX,MaxValueX);
        MinValueX=min(Vpt[i].VmaxX,MinValueX);
        MaxValueY=max(Vpt[i].VmaxY,MaxValueY);
        MinValueY=min(Vpt[i].VmaxY,MinValueY);

        if (abs(MinValueX)>MaxValueX) MaxMax.Vx=MinValueX;
        else MaxMax.Vx=MaxValueX;

        if (abs(MinValueY)>MaxValueY) MaxMax.Vy=MinValueY;
        else MaxMax.Vy=MaxValueY;
    }
    for (i=0;i<LimitPointCounter;i++)
    {
        if (MaxValueX==Vpt[i].VmaxX)
        {
            MaxMax.framex=Vpt[i].framex;
            MaxMax.x=Vpt[i].point[0];
        }
        if (MaxValueY==Vpt[i].VmaxY)
        {
            MaxMax.framey=Vpt[i].framey;
            MaxMax.y=Vpt[i].point[0];
        }
    }
}

void CChildView::OnFileExport()
{
    int i;
    HRESULT hResult1;
    CString file_types =
    "Text Files(*.txt)*.txt";

    CFileDialog dlg(FALSE,NULL,NULL,OFN_HIDEREADONLY |
    OFN_OVERWRITEPROMPT | OFN_EXPLORER,(LPCTSTR)file_types,NULL);

```

```

        dlg.m_ofn.nFilterIndex = m_nFilterLoad;
        hResult1 = (int)dlg.DoModal();
        if (hResult1 != IDOK) {
            return;
        }

        CString strExtension,strFileName;

        strFileName = dlg.m_ofn.lpstrFile;

        // add the file extension if the user didn't supply one
        if (dlg.m_ofn.nFileExtension == 0)
        {
            strExtension = "txt";
            strFileName = strFileName + '.' + strExtension;
        }
        if (FAILED(hResult1)) {
            CString fmt;
            fmt.Format("Output text save failed:\n%x - %s", hResult1,
_com_error(hResult1).ErrorMessage());
            ::AfxMessageBox(fmt);
            return;
        }

        Invalidate();
        UpdateWindow();

        FileStream* fs = File::Create(strFileName);

        char buffer[80];

        sprintf (buffer,"XCOORD,YCOORD: (%i,%i)\n",XCOORD,YCOORD);
        Byte info[] = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"ROISIZEX,ROISIZEY: (%i,%i)\n",ROISIZEX,ROISIZEY);
        info = (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        sprintf (buffer,"LimitPointCounter: %i\n",LimitPointCounter);
        info= (new UTF8Encoding(true))->GetBytes(buffer);
        fs->Write(info, 0, info->Length);

        for (i=0;i<LimitPointCounter;i++)
        {
            sprintf (buffer,"VptPoint: (%i,%i)\n",Vpt[i].point[0].x,Vpt[i].point[0].y);
            info= (new UTF8Encoding(true))->GetBytes(buffer);
            fs->Write(info, 0, info->Length);
        }

        fs->Close();
        MessageBox("DONE","!");
    }

void CChildView::OnMenu()
{
    FILE *stream;

    int i,i_temp,k,n;
    Invalidate();

```

```

UpdateWindow();
char buffer[100],temp[10];;
    CString file_types =
"TXT Files(*.txt)|*.txt;||";
    CFileDialog dlg(TRUE,
        NULL,
        NULL,
        OFN_HIDEREADONLY | OFN_EXPLORER,
        (LPCTSTR)file_types,NULL);

    HRESULT hResult;

    dlg.m_ofn.nFilterIndex = m_nFilterLoad;
    hResult = (int)dlg.DoModal();
    if(hResult != IDOK) {
        return;
    }

    CString strFileName=dlg.GetPathName();

stream=fopen(strFileName,"r");

    //Διάβασε τα XCOOR YCOOR
    //XCOOR
    fgets(buffer,100,stream);
    i=0;
    while (buffer[i]!='(') i++;
    i++;
    i_temp=i;
    while (buffer[i]!=')') i++;

    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];
    temp[k++]='\n';
    XCOOR=atoi(temp);
    //YCOOR
    i_temp=++i;
    while (buffer[i]!=')') i++;
    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];
    temp[k++]='\n';
    YCOOR=atoi(temp);

    //Διάβασε τα ROISIZEX ROISIZEY
    //ROISIZEX
    fgets(buffer,100,stream);
    i=0;
    while (buffer[i]!='(') i++;
    i++;
    i_temp=i;
    while (buffer[i]!=')') i++;
    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];
    temp[k++]='\n';
    ROISIZEX=atoi(temp);
    //ROISIZEY
    i_temp=++i;
    while (buffer[i]!=')') i++;
    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];

```

```

temp[k++]='\n';
ROISIZEY=atoi(temp);

//Διάβασε το LimitPointCounter
fgets(buffer,100,stream);
i=0;
while (buffer[i]!=' ') i++;
i++;
i_temp=i;
while (buffer[i]!='\n') i++;
for (k=0;k<i-i_temp;k++)
    temp[k]=buffer[i_temp+k];
temp[k++]='\n';
LimitPointCounter=atoi(temp);

//Διάβασε τα σημεία
for (n=0;n<LimitPointCounter;n++)
{
    fgets(buffer,100,stream);
    //X
    i=0;
    while (buffer[i]!='(') i++;
    i++;
    i_temp=i;
    while (buffer[i]!=' ') i++;
    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];
    temp[k++]='\n';
    Vpt[n].point[0].x=atoi(temp);
    //Y
    i_temp=++i;
    while (buffer[i]!=')') i++;
    for (k=0;k<i-i_temp;k++)
        temp[k]=buffer[i_temp+k];
    temp[k++]='\n';
    Vpt[n].point[0].y=atoi(temp);
}

OKCompute=true;
RoiSet=true;
selectROI=false;
fclose(stream);

//Εμφάνισε τις συνταταγμένες
Invalidate();
UpdateWindow();
{
    CStatic* pInfo;
    char buffer[30];
    pCoor->ShowWindow(SW_SHOW);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_TopLeft);
    sprintf(buffer,"%i,%i",XCOORD,YCOORD);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_TopRight);
    sprintf(buffer,"%i,%i",XCOORD+ROISIZE_X,YCOORD);
    pInfo->SendMessage(WM_SETTEXT,0,(LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_BottomLeft);

```

```

    sprintf(buffer, "(%i,%i)", XCOORD, YCOORD+ROISIZEY);
    pInfo->SendMessage(WM_SETTEXT, 0, (LPARAM)buffer);

    pInfo = (CStatic*) pCoor->GetDlgItem(IDC_BottomRight);
    sprintf(buffer, "(%i,%i)", XCOORD+ROISIZEX, YCOORD+ROISIZEY);
    pInfo->SendMessage(WM_SETTEXT, 0, (LPARAM)buffer);

}
char bf[5];
sprintf(bf, "%i.tif", digs);
imgOriginal.Destroy();
imgOriginal.Load(fname+bf);
MarkROI();
Invalidate();
UpdateWindow();
CClientDC dc(this);

Compute();
Computed=true;
CompPoints();
MaxVelocity();
MaxMaxVelocity();
SortPoints();
MaxSuccessiveDiff();

for (n=0; n<LimitPointCounter; n++)
    dc.SetPixel(Vpt[n].point[0], RGB(0,255,0));

for (i=0; i<LimitPointCounter; i++)
{
    VptNorm[i].point[0]=Vpt[i].point[0];
    VptNorm[i].VmaxX=Vpt[i].VmaxX/NormValue ;
    VptNorm[i].VmaxY=Vpt[i].VmaxY/NormValue ;
}
}

void CChildView::SortPoints(void)
{
    int i,j,m,n,k,P1x,P1y,P2x,P2y;
    CPoint swap;
    double d;
    for(i=SEARCHAREA; i<LimitPointCounter; i+=SEARCHAREA)
    {
        difx[i/SEARCHAREA-1].V=0;
        dify[i/SEARCHAREA-1].V=0;
    }

    //Ταξινόμηση των σημείων κατά x
    for (i=1; i<LimitPointCounter; i++)
        for (j=LimitPointCounter-1; j>i; j--)
            if (Vpt[j-1].point[0].x>Vpt[j].point[0].x)
            {
                swap=Vpt[j].point[0];
                Vpt[j].point[0]=Vpt[j-1].point[0];
                Vpt[j-1].point[0]=swap;
            }

    CompPoints();

    //Μέγιστες διαφορές κατά x

```

```

for(i=SEARCHAREA;i<LimitPointCounter;i+=SEARCHAREA)
    for (m=i-SEARCHAREA;m<i;m++)
        for (n=i-SEARCHAREA;n<i;n++)
            for (k=0;k<NumFiles;k++)
            {
                P1x=Vpt[m].point[k].x-XCOORD;
                P1y=Vpt[m].point[k].y-YCOORD;
                P2x=Vpt[n].point[k].x-XCOORD;
                P2y=Vpt[n].point[k].y-YCOORD;
                d=abs(Vx[P1x][P1y][k]-Vx[P2x][P2y][k]);
                if (d>difx[i/SEARCHAREA-1].V)
                {
                    difx[i/SEARCHAREA-1].V=d;
                    difx[i/SEARCHAREA-1].point1=Vpt[m].point[0];
                    difx[i/SEARCHAREA-1].point2=Vpt[n].point[0];
                    difx[i/SEARCHAREA-1].V1=Vx[P1x][P1y][k];
                    difx[i/SEARCHAREA-1].V2=Vx[P2x][P2y][k];
                    difx[i/SEARCHAREA-1].frame=k;
                }
            }
    }

//Ταξινόμηση των σημείων κατά y
for (i=1;i<LimitPointCounter;i++)
    for (j=LimitPointCounter-1;j>i;j--)
        if (Vpt[j-1].point[0].y>Vpt[j].point[0].y)
        {
            swap=Vpt[j].point[0];
            Vpt[j].point[0]=Vpt[j-1].point[0];
            Vpt[j-1].point[0]=swap;
        }

CompPoints();

//Μέγιστες διαφορές κατά y
for(i=SEARCHAREA;i<LimitPointCounter;i+=SEARCHAREA)
    for (m=i-SEARCHAREA;m<i;m++)
        for (n=i-SEARCHAREA;n<i;n++)
            for (k=0;k<NumFiles;k++)
            {
                P1x=Vpt[m].point[k].x-XCOORD;
                P1y=Vpt[m].point[k].y-YCOORD;
                P2x=Vpt[n].point[k].x-XCOORD;
                P2y=Vpt[n].point[k].y-YCOORD;
                d=abs(Vy[P1x][P1y][k]-Vy[P2x][P2y][k]);
                if (d>dify[i/SEARCHAREA-1].V)
                {
                    dify[i/SEARCHAREA-1].V=d;
                    dify[i/SEARCHAREA-1].point1=Vpt[m].point[0];
                    dify[i/SEARCHAREA-1].point2=Vpt[n].point[0];
                    dify[i/SEARCHAREA-1].V1=Vy[P1x][P1y][k];
                    dify[i/SEARCHAREA-1].V2=Vy[P2x][P2y][k];
                    dify[i/SEARCHAREA-1].frame=k;
                }
            }
    }

}

void CChildView::OnRunMaxdifferences()
{
    Invalidate();
}

```

```

        UpdateWindow();
        SortPoints();
        DrawSortPoints();
    }

void CChildView::NormVelocities() //Σκοπός είναι να ομαλοποιήσω τις ταχύτητες

{
    int m,n,k,i,j;
    for (k=0;k<NumFiles;k++)
    {

        for(i=0;i<ROISIZEX;i++)
            for(j=0;j<ROISIZEY;j++)
            {
                dy[i][j]=0.0;
                dx[i][j]=0.0;
            }

        //Τα σημεία της περιμέτρου - συνέλιξη

        i=XCOOR;//πάνω πλευρά
        for(j=YCOOR;j<YCOOR+ROISIZEY;j++)
            for (m=-1;m<=1;m++)
                for (n=-1;n<=1;n++)
                {
                    if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
                    {
                        dy[i-XCOOR][j-YCOOR]+=Vy[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                        dx[i-XCOOR][j-YCOOR]+=Vx[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                    }
                }

        i=XCOOR+ROISIZEX-1;//κάτω πλευρά
        for(j=YCOOR;j<YCOOR+ROISIZEY;j++)
            for (m=-1;m<=1;m++)
                for (n=-1;n<=1;n++)
                {
                    if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
                    {
                        dy[i-XCOOR][j-YCOOR]+=Vy[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                        dx[i-XCOOR][j-YCOOR]+=Vx[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                    }
                }

        j=YCOOR;//αριστερή πλευρά
        for(i=XCOOR;i<XCOOR+ROISIZEX;i++)
            for (m=-1;m<=1;m++)
                for (n=-1;n<=1;n++)
                {
                    if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
                    {
                        dy[i-XCOOR][j-YCOOR]+=Vy[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                        dx[i-XCOOR][j-YCOOR]+=Vx[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
                    }
                }

        j=YCOOR+ROISIZEY-1;//δεξιά πλευρά
        for(i=XCOOR;i<XCOOR+ROISIZEX;i++)
    }
}

```

```

        for (m=-1;m<=1;m++)
            for (n=-1;n<=1;n++)
            {
if ((i-XCOOR+m>=0)&(i-XCOOR+m<ROISIZEX)&(j-YCOOR+n>=0)&(j-YCOOR+n<ROISIZEY))
            {
dy[i-XCOOR][j-YCOOR]+=Vy[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=Vx[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
            }
        }

//Τα κεντρικά σημεία εκτός από την περίμετρο - συνέλιξη

for(i=XCOOR+1;i<XCOOR+ROISIZEX-1;i++)
    for(j=YCOOR+1;j<YCOOR+ROISIZEY-1;j++)
        for (m=-1;m<=1;m++)
            for (n=-1;n<=1;n++)
            {

dy[i-XCOOR][j-YCOOR]+=Vy[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];
dx[i-XCOOR][j-YCOOR]+=Vx[i-XCOOR+m][j-YCOOR+n][k]*PreWittT[1+m][1+n];

            }

        for(i=0;i<ROISIZEX;i++)
            for(j=0;j<ROISIZEY;j++)
            {

                Vy[i][j][k]=dy[i][j]/9.0;
                Vx[i][j][k]=dx[i][j]/9.0;

            }
    }
}
void CChildView::MaxSuccessiveDiff(void)
{
    int i,k,x0,y0,x1,y1;
    double d_x,d_y;
    SuccessiveMaxX.V=-10.0;
    SuccessiveMaxY.V=-10.0;
    for(i=0;i<LimitPointCounter;i++)
        for(k=0;k<NumFiles-1;k++)
        {
            x0=Vpt[i].point[k].x;
            y0=Vpt[i].point[k].y;
            x1=Vpt[i].point[k+1].x;
            y1=Vpt[i].point[k+1].y;
            d_x=abs(Vx[x0][y0][k]-Vx[x1][y1][k+1]);
            d_y=abs(Vy[x0][y0][k]-Vy[x1][y1][k+1]);

            if (d_x>SuccessiveMaxX.V)
            {
                SuccessiveMaxX.V=d_x;
                SuccessiveMaxX.point=Vpt[i].point[0];
                SuccessiveMaxX.frame=k;
            }
            if (d_y>SuccessiveMaxY.V)
            {
                SuccessiveMaxY.V=d_y;
                SuccessiveMaxY.point=Vpt[i].point[0];

```

```

SuccessiveMaxY.frame=k;
    }
}
}
void CChildView::DrawSortPoints(void)
{
    int i;
    CClientDC dc(this);
    int x0,y0,ArrowLength,ArrowNose,ArrowCone;
    int quantum=LimitPointCounter/SEARCHAREA;
    for(i=SEARCHAREA;i<LimitPointCounter;i+=SEARCHAREA)
    {

        //Vy 1ο σημείο
        x0=dify[i].point1.x;
        y0=dify[i].point1.y;
        ArrowLength=-40*dify[i].V1/NormValue;
        ArrowNose=0.7*ArrowLength;
        ArrowCone=0.3*ArrowLength;

        CPen lpen1(PS_SOLID, 1, RGB(255*i/quantum,255*i/quantum,0));
        dc.SelectObject(&lpen1);

        dc.MoveTo(x0,y0);
        dc.LineTo(x0,y0-ArrowLength);

        dc.LineTo(x0-ArrowCone,y0-ArrowNose);

        dc.MoveTo(x0,y0-ArrowLength);
        dc.LineTo(x0+ArrowCone,y0-ArrowNose);
        //Vy 2ο σημείο
        x0=dify[i].point2.x;
        y0=dify[i].point2.y;
        ArrowLength=-40*dify[i].V2/NormValue;
        ArrowNose=0.7*ArrowLength;
        ArrowCone=0.3*ArrowLength;

        CPen lpen1(PS_SOLID, 1, RGB(255*i/quantum,255*i/quantum,0));
        dc.SelectObject(&lpen1);

        dc.MoveTo(x0,y0);
        dc.LineTo(x0,y0-ArrowLength);

        dc.LineTo(x0-ArrowCone,y0-ArrowNose);

        dc.MoveTo(x0,y0-ArrowLength);
        dc.LineTo(x0+ArrowCone,y0-ArrowNose);

        //Vx 1ο σημείο
        x0=difx[i].point2.x;
        y0=difx[i].point2.y;
        ArrowLength=-40*difx[i].V2/NormValue;
        ArrowNose=0.7*ArrowLength;
        ArrowCone=0.3*ArrowLength;

        CPen lpen2(PS_SOLID, 1, RGB(0,255*i/quantum,255*i/quantum));
        dc.SelectObject(&lpen2);

        dc.MoveTo(x0,y0);
        dc.LineTo(x0-ArrowLength,y0);
    }
}

```

```
        dc.LineTo(x0-ArrowNose,y0-ArrowCone);

        dc.MoveTo(x0-ArrowLength,y0);
        dc.LineTo(x0-ArrowNose,y0+ArrowCone);

        //Vx 2ο σημείο
        x0=difx[i].point1.x;
        y0=difx[i].point1.y;
        ArrowLength=-40*difx[i].V1/NormValue;
        ArrowNose=0.7*ArrowLength;
        ArrowCone=0.3*ArrowLength;

        CPen lpen2(PS_SOLID, 1, RGB(0,255*i/quantum,255*i/quantum));
        dc.SelectObject(&lpen2);

        dc.MoveTo(x0,y0);
        dc.LineTo(x0-ArrowLength,y0);

        dc.LineTo(x0-ArrowNose,y0-ArrowCone);

        dc.MoveTo(x0-ArrowLength,y0);
        dc.LineTo(x0-ArrowNose,y0+ArrowCone);
    }
}
```

ChildView.h

```
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (c) Microsoft Corporation. All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and related
// electronic documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.
// ChildView.h : interface of the CChildView class
//

#pragma once
#include <afxstr.h>
#include <atlimage.h>
#include <comdef.h>

class CChildView : public CWnd
{
// Construction
public:
    CChildView();

// Attributes
public:

// Operations
public:

    double df [MAX_ROISIZE][MAX_ROISIZE];
    double dx [MAX_ROISIZE-1][MAX_ROISIZE];
    double dt [MAX_ROISIZE][MAX_ROISIZE];
    double dy [MAX_ROISIZE][MAX_ROISIZE-1];
    double dtemp [MAX_ROISIZE][MAX_ROISIZE];
    double Vx [MAX_ROISIZE-1][MAX_ROISIZE-1][MAXFRAME+1];
    double Vy [MAX_ROISIZE-1][MAX_ROISIZE-1][MAXFRAME+1];

    struct Vpt
    {
        CPoint point[MAXFRAME];
        double VmaxX;
        double VmaxY;
        int framex;
        int framey;
    } Vpt[LimitPointCounterMAXSIZE];

    CPoint SortedPoints[LimitPointCounterMAXSIZE];
    struct Succes
    {
        double V;
        CPoint point;
        int frame;
    }
    SuccessiveMaxX, SuccessiveMaxY;

    struct MaxMax
    {
        double Vx;
```

```

        double Vy;
        CPoint x,y;
        int framex;
        int framey;
    }
    MaxMax;

struct Vpt VptNorm[LimitPointCounterMAXSIZE];

struct dif
{
    double V;
    CPoint point1;
    CPoint point2;
    double V1;
    double V2;
    int frame;
}
difx[LimitPointCounterMAXSIZE/SEARCHAREA],dify[LimitPointCounterMAXSIZE/SEARCHAR
EA];

CString fname;
int NumFiles,digs;
boolean down,selectROI,OKCompute;
CPoint start,end;
int ROISIZEX,ROISIZEY,XCOORD,YCOORD;
int width,height;
bool MarkLimits;
bool CoordDlg,RoiSet;
CDialog* pDialog;
CDialog* pProg;
CDialog* pCoor;
CDialog* pZoom;
CPoint oldpoint;
int PreWittX[3][3],      PreWittY[3][3],PreWittT[3][3];
int LimitPointCounter;
int MSPrevX,MSPrevY;
bool Computed;
// Overrides
protected:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

// Implementation
public:
    virtual ~CChildView();
    // Generated message map functions

private:
    CImage imgOriginal;
    int      m_nFilterLoad;

protected:
    afx_msg void OnPaint();
    DECLARE_MESSAGE_MAP()
public:
    void OnFileOpenimage(void);
    void OnImageInfo(void);
    void OnUpdateImage(CCmdUI* pCmdUI);
    void OnSelectRoi(void);
    void Compute(void);
    void FirstFrame(CImage &image);

```

```
void MarkROI(void);
void NextFrame(CImage &image,int frame);
afx_msg void OnPlayPlaysequence();
afx_msg void OnMouseMove(UINT nFlags, CPoint point);
afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
afx_msg void OnLButtonUp(UINT nFlags, CPoint point);
afx_msg void OnRunCompute();
int HornSchunk(int frame);
afx_msg void OnRunMarklimits();
int MaxVelocity(void);
afx_msg void OnRunDonemarkinglimits();
void ShowArrows(bool V);
void CompPoints(void);
void OutPutFile(void);
afx_msg void OnShowarrowsVx();
afx_msg void OnShowarrowsVy();
afx_msg void OnShowarrowsBoth();
void MaxMaxVelocity(void);
void SortPoints(void);
void NormVelocities(void);
afx_msg void OnFileExport();
afx_msg void OnMenu();
afx_msg void OnRunMaxdifferences();
void MaxSuccessiveDiff();
void DrawSortPoints(void);
};

int CheckNum(CString inp);
```

Cimage.cpp

```
// CImage.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "CImage.h"
#include "MainFrm.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// CImageApp

BEGIN_MESSAGE_MAP(CImageApp, CWinApp)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
END_MESSAGE_MAP()

// CImageApp construction

CImageApp::CImageApp()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

// The one and only CImageApp object

CImageApp theApp;

// CImageApp initialization

BOOL CImageApp::InitInstance()
{
    CWinApp::InitInstance();

    // Initialize OLE libraries
    if (!AfxOleInit())
    {
        AfxMessageBox(IDP_OLE_INIT_FAILED);
        return FALSE;
    }
    AfxEnableControlContainer();
    SetRegistryKey(_T("Local AppWizard-Generated Applications"));
    // To create the main window, this code creates a new frame window
    // object and then sets it as the application's main window object
    CMainFrame* pFrame = new CMainFrame;
    m_pMainWnd = pFrame;
    // create and load the frame with its resources
    pFrame->LoadFrame(IDR_MAINFRAME,
        WS_OVERLAPPEDWINDOW | FWS_ADDTOTITLE, NULL,
        NULL);
    // The one and only window has been initialized, so show and update it
    pFrame->ShowWindow(SW_SHOW);
    pFrame->UpdateWindow();
    // call DragAcceptFiles only if there's a suffix
```

```
        // In an SDI app, this should occur after ProcessShellCommand
        return TRUE;
    }

    // CImageApp message handlers

    // CAboutDlg dialog used for App About

    class CAboutDlg : public CDialog
    {
    public:
        CAboutDlg();

    // Dialog Data
        enum { IDD = IDD_ABOUTBOX };

    protected:
        virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support

    // Implementation
    protected:
        DECLARE_MESSAGE_MAP()
    };

    CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
    {
    }

    void CAboutDlg::DoDataExchange(CDataExchange* pDX)
    {
        CDialog::DoDataExchange(pDX);
    }

    BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    END_MESSAGE_MAP()

    // App command to run the dialog
    void CImageApp::OnAppAbout()
    {
        CAboutDlg aboutDlg;
        aboutDlg.DoModal();
    }

    // CImageApp message handlers
```

Cimage.h

```
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (c) Microsoft Corporation. All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and related
// electronic documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

// CImage.h : main header file for the CImage application
//
#pragma once

#ifdef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h"    // main symbols

// CImageApp:
// See CImage.cpp for the implementation of this class
//

class CImageApp : public CWinApp
{
public:
    CImageApp();

// Overrides
public:
    virtual BOOL InitInstance();

// Implementation
public:
    afx_msg void OnAppAbout();
    DECLARE_MESSAGE_MAP()
};
extern CImageApp theApp;
```

MainFrame.cpp

```
// MainFrm.cpp : implementation of the CMainFrame class
//

#include "stdafx.h"
#include "CImage.h"
#include "MainFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// CMainFrame

IMPLEMENT_DYNAMIC(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
    ON_WM_CREATE()
    ON_WM_SETFOCUS()
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,      // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};

static UINT BASED_CODE sizes[]= // this is used to create the 'size' toolbar
{
    ID_SIZE_HALF,
    ID_SIZE_ORIGINAL,
    ID_SIZE_DOUBLE,
    ID_SIZE_FILL
};

// CMainFrame construction/destruction
CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here
    m_bIsFullScreen=FALSE;
}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;
    // create a view to occupy the client area of the frame
    if (!m_wndView.Create(NULL, NULL, AFX_WS_DEFAULT_VIEW,
        CRect(0, 0, 0, 0), this, AFX_IDW_PANE_FIRST, NULL))
    {
        TRACE0("Failed to create view window\n");
        return -1;
    }
}
```

```

        return 0;
    }

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    if( !CFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    cs.dwExStyle &= ~WS_EX_CLIENTEDGE;
    cs.lpszClass = AfxRegisterWndClass(0);
    cs.lpszName ="Optical Flow Computation";

    cs.style = WS_OVERLAPPED | WS_SYSMENU | WS_BORDER | WS_SIZEBOX;
    cs.x=0;
    cs.y=0;
    cs.cy = ::GetSystemMetrics(SM_CYSCREEN)*0.85 ;
    cs.cx = ::GetSystemMetrics(SM_CXSCREEN)*0.8;

    return TRUE;
}
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
    CFrameWnd::Dump(dc);
}

#endif // _DEBUG

// CMainFrame message handlers

void CMainFrame::OnSetFocus(CWnd* /*pOldWnd*/)
{
    // forward focus to the view window
    m_wndView.SetFocus();
}

BOOL CMainFrame::OnCmdMsg(UINT nID, int nCode, void* pExtra, AFX_CMDHANDLERINFO*
pHandlerInfo)
{
    // let the view have first crack at the command
    if (m_wndView.OnCmdMsg(nID, nCode, pExtra, pHandlerInfo))
        return TRUE;

    // otherwise, do default handling
    return CFrameWnd::OnCmdMsg(nID, nCode, pExtra, pHandlerInfo);
}

```

MainFrame.h

```
// This is a part of the Microsoft Foundation Classes C++ library.
// Copyright (c) Microsoft Corporation. All rights reserved.
//
// This source code is only intended as a supplement to the
// Microsoft Foundation Classes Reference and related
// electronic documentation provided with the library.
// See these sources for detailed information regarding the
// Microsoft Foundation Classes product.

// MainFrm.h : interface of the CMainFrame class
//

#pragma once

#include "ChildView.h"

class CMainFrame : public CFrameWnd
{
public:
    CMainFrame();
protected:
    DECLARE_DYNAMIC(CMainFrame)

// Attributes
public:
// Operations
public:
// Overrides
public:
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    virtual BOOL OnCmdMsg(UINT nID, int nCode, void* pExtra,
AFX_CMDHANDLERINFO* pHandlerInfo);
// Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected: // control bar embedded members
    CStatusBar          m_wndStatusBar;
    CToolBar            m_wndToolBar;
    CChildView          m_wndView;
    CToolBar            m_wndSizeToolBar;    // this toolbar contains the image 'Size'
buttons
    BOOL m_bIsFullScreen;
    WINDOWPLACEMENT m_wpPrev; // to store the window placement when we switch to
fullscreen

// Generated message map functions
protected:
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
    afx_msg void OnSetFocus(CWnd *pOldWnd);
    DECLARE_MESSAGE_MAP()
};
```

ProgDlg.cpp

```
// ProgDlg.cpp : implementation file
//

#include "stdafx.h"
#include "ChildView.h"
#include "ProgDlg.h"
#include "Resource.h"

// CProgDlg dialog

IMPLEMENT_DYNAMIC(CProgDlg, CDialog)
CProgDlg::CProgDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CProgDlg::IDD, pParent)
{
}

CProgDlg::~CProgDlg()
{
}

void CProgDlg::DoDataExchange(CDataExchange* pDX)
{
    CenterWindow();
    CDialog::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CProgDlg, CDialog)
END_MESSAGE_MAP()

// CProgDlg message handlers
```

ProgDlg.h

```
#include "Resource.h"
#pragma once

// CProgDlg dialog

class CProgDlg : public CDialog
{
    DECLARE_DYNAMIC(CProgDlg)

public:
    CProgDlg(CWnd* pParent = NULL); // standard constructor
    virtual ~CProgDlg();

// Dialog Data
    enum { IDD = IDD_PROG };

protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support

    DECLARE_MESSAGE_MAP()
};
```

Resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by CImage.rc
//
#define IDD_ABOUTBOX 100
#define IDP_OLE_INIT_FAILED 100
#define IDR_HTML_COORD 103
#define IDR_MAINFRAME 128
#define IDR_CImageTYPE 129
#define ID_FILE_OPENIMAGE 130
#define ID_TOOLS_MAKEBW 131
#define ID_FILE_IMAGEINFO 132
#define ID_FILE_SAVEIMAGE 133
#define IDD_DIALOG1 134
#define IDC_PROG 134
#define IDR_TOOLBAR1 142
#define IDD_PROG 145
#define IDD_ROI 146
#define IDD_COOR 147
#define IDC_IMAGESIZE 1002
#define IDC_PITCH 1004
#define IDC_DIBSECTION 1005
#define IDC_INDEXED 1006
#define IDC_BITDEPTH 1007
#define IDC_PROGRESS1 1011
#define IDC_XCOOR 1015
#define IDC_YCOOR 1016
#define IDC_FNAME 1017
#define IDC_TopLeft 1018
#define IDC_BottomLeft 1019
#define IDC_TopRight 1020
#define IDC_BottomRight 1021
#define IDC_VX 1026
#define IDC_VY 1027
#define IDC_XPOS 1029
#define IDC_YPOS 1030
#define IDC_XRANGE 1032
#define IDC_YRANGE 1033
#define IDC_BUTTON1 1035
#define ID_SIZE_HALF 2001
#define ID_SIZE_ORIGINAL 2002
#define ID_SIZE_DOUBLE 2003
#define ID_SIZE_FILL 2004
#define ID_SIZE_FULLSCREEN 2005
#define ID_FILE_SELECTROI 32778
#define ID_SELECTROI_FREE 32780
#define ID_SELECTROI_RECTANGLE 32781
#define ID_RUN_FREE 32783
#define ID_RUN_RECTANGLE 32784
#define ID_PLAY_PLAYSEQUENCE 32785
#define ID_RUN_COMPUTE 32786
#define ID_RUN_GRAPH 32787
#define ID_RUN_MARKLIMITS 32788
#define ID_RUN_DONEMARKINGLIMITS 32789
#define ID_RUN_SHOWARROWS 32790
#define ID_SHOWARROWS_VX 32791
#define ID_SHOWARROWS_VY 32792
#define ID_SHOWARROWS_BOTH 32793
#define ID_FILE_EXPORT 32794
```

```
#define ID_Menu 32795
#define ID_RUN_MAXDIFFERENCES 32796

//Mine
#define MAXFRAME 100
#define MAX_ROISIZE 400
#define LAMDA 1
#define LimitPointCounterMAXSIZE 5000
#define ZoomY 2
#define ZoomX 2
#define NormValue 5
#define ImageDensity 178 //Pixels/cm
#define FrameRate 25//Hz
#define SEARCHAREA 10

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 152
#define _APS_NEXT_COMMAND_VALUE 32797
#define _APS_NEXT_CONTROL_VALUE 1036
#define _APS_NEXT_SYMED_VALUE 104
#endif
#endif
```

stdafx.cpp

```
// This is a part of the Microsoft Foundation Classes C++ library.  
// Copyright (c) Microsoft Corporation. All rights reserved.  
//  
// This source code is only intended as a supplement to the  
// Microsoft Foundation Classes Reference and related  
// electronic documentation provided with the library.  
// See these sources for detailed information regarding the  
// Microsoft Foundation Classes product.  
  
// stdafx.cpp : source file that includes just the standard includes  
// CImage.pch will be the pre-compiled header  
// stdafx.obj will contain the pre-compiled type information  
  
#include "stdafx.h"
```

stdafx.h

```
// This is a part of the Microsoft Foundation Classes C++ library.  
// Copyright (c) Microsoft Corporation. All rights reserved.  
//  
// This source code is only intended as a supplement to the  
// Microsoft Foundation Classes Reference and related  
// electronic documentation provided with the library.  
// See these sources for detailed information regarding the  
// Microsoft Foundation Classes product.  
  
// stdafx.h : include file for standard system include files,  
// or project specific include files that are used frequently,  
// but are changed infrequently  
  
#pragma once  
  
#ifndef VC_EXTRALEAN  
#define VC_EXTRALEAN // Exclude rarely-used stuff from Windows headers  
#endif  
  
// Modify the following defines if you have to target a platform prior to the ones specified below.  
// Refer to MSDN for the latest info on corresponding values for different platforms.  
#ifndef WINVER // Allow use of features specific to Windows 95 and  
Windows NT 4 or later.  
#define WINVER 0x0400 // Change this to 0x0500 to target Windows 98 and Windows 2000.  
#endif  
  
#ifndef _WIN32_WINNT // Allow use of features specific to Windows NT 4 or later.  
#define _WIN32_WINNT 0x0400 // Change this to 0x0500 to target Windows 2000.  
#endif  
  
#ifndef _WIN32_WINDOWS // Allow use of features specific to Windows 98 or later.  
#define _WIN32_WINDOWS 0x0410  
#endif  
  
#ifndef _WIN32_IE // Allow use of features specific to IE 4.0 or later.  
#define _WIN32_IE 0x0400 // Change this to 0x0500 to target IE 5.0.  
#endif
```

```
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS    // some CString constructors will be
explicit

// turns off MFC's hiding of some common and often safely ignored warning messages
#define _AFX_ALL_WARNINGS

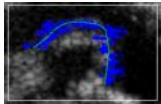
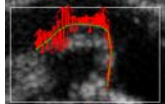
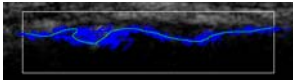
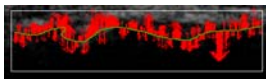
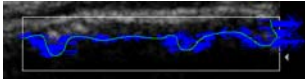
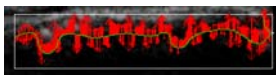
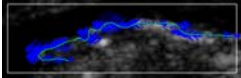
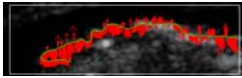
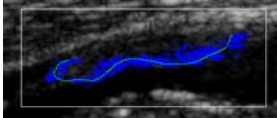
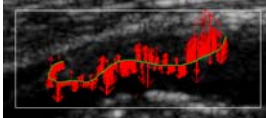
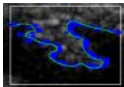
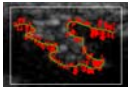
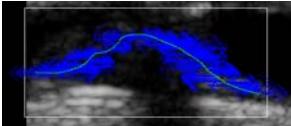
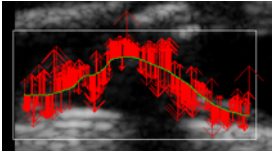
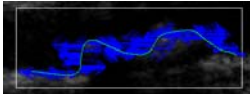
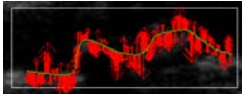
#include <winerror.h>
#include <afxwin.h>        // MFC core and standard components
#include <afxext.h>        // MFC extensions
#include <afxdisp.h>       // MFC Automation classes

#include <afxdtctl.h>       // MFC support for Internet Explorer 4 Common Controls
#ifndef _AFX_NO_AFXCMN_SUPPORT
#include <afxcmn.h>         // MFC support for Windows Common Controls
#endif // _AFX_NO_AFXCMN_SUPPORT
#include <afxhtml.h>
```

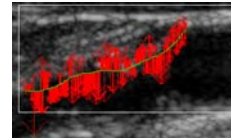
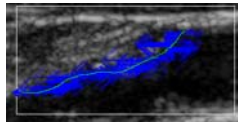

Παράρτημα Β

Παρατίθενται screenshots από την εκτέλεση του προγράμματος για κάθε ακολουθία.

Ασυμπτωματικές πλάκες

Όνομα ακολουθίας	Άξονας x	Άξονας γ
bx		
cbl		
cbr		
ehr		
hh		
ir		
rma		
rtl		

sb



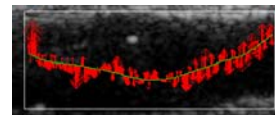
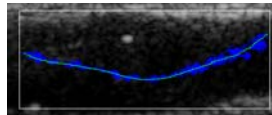
Συμπτωματικές πλάκες

Όνομα
ακολουθίας

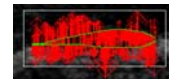
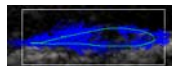
Άξονας x

Άξονας y

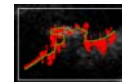
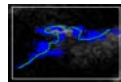
ag



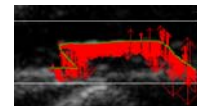
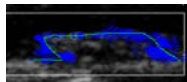
bm



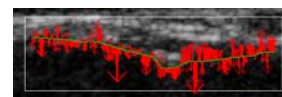
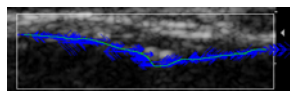
eg



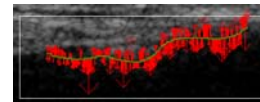
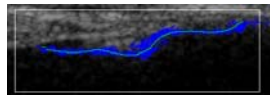
ehl1



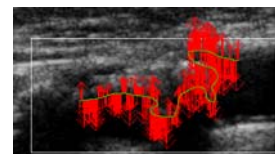
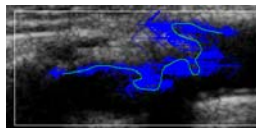
gf



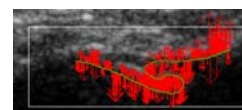
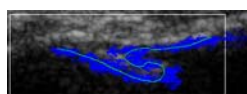
gk



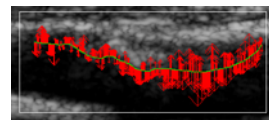
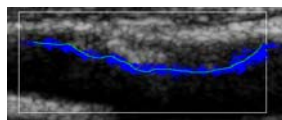
gs



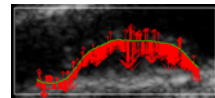
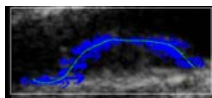
jh



kr



mh



Περίληψη

"Μελέτη κίνησης αθηρωματικών πλακών καρωτίδας από ακολουθίες υπερήχων Β-σάρωσης"

Αλέξανδρος Κ. Δημόπουλος

Διπλωματική εργασία, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο, 2004

Οι καρδιαγγειακές παθήσεις αποτελούν τη σημαντικότερη πηγή νοσηρότητας και θνησιμότητας στο δυτικό κόσμο. Υπάρχουν ενδείξεις ότι η μορφολογία των αθηρωματικών πλακών της καρωτίδας, που λαμβάνεται από υπερηχογράφημα μπορεί να χρησιμοποιηθεί για προγνωστικούς σκοπούς. Έτσι θα αποφεύγεται η χειρουργική επέμβαση σε ένα μεγάλο αριθμό ασθενών για τους οποίους δεν είναι απαραίτητη η επέμβαση. Σκοπός αυτής της μελέτης είναι η ανάπτυξη μιας εφαρμογής σε υπολογιστή, μέσω της οποίας θα συντελείται ο χαρακτηρισμός των αθηρωματικών πλακών με σκοπό των εντοπισμό ασθενών με υψηλή επικινδυνότητα για εγκεφαλικό επεισόδιο. Η κίνηση των αθηρωματικών πλακών μελετήθηκε με τη χρήση της μεθόδου της οπτικής ροής. Η μέθοδος βασίζεται στην υπόθεση ότι κάθε φυσικό σημείο έχει την ίδια φωτεινότητα σε δύο διαδοχικά καρέ της ακολουθίας. Συνεπώς, είναι δυνατόν να προβλεφθεί η κίνηση ενός *pixel* με την επεξεργασία των δύο καρέ. Ένα πυκνό πεδίο διανυσμάτων κίνησης υπολογίζεται για κάθε εικόνα βάσει των χωρικών και χρονικών παραγώγων της έντασης της φωτεινότητας.

Η εφαρμογή έχει γραφεί σε MS Visual Studio .NET C++ και έχει μέγεθος περί τις 3000 γραμμές. Ο χρήστης μπορεί να φορτώσει μια ακολουθία εικόνων μορφής *tiff* - οι οποίες έχουν ληφθεί από υπερηχογράφημα της καρωτίδας - και να επιλέξει μια περιοχή ενδιαφέροντος.

Στη συνέχεια, επιλέγονται τα όρια της πλάκας και η εφαρμογή υπολογίζει την αναπτυσσόμενη ταχύτητα για κάθε σημείο του ορίου. Υπολογίζεται η μέγιστη τιμή της αναπτυσσόμενης ταχύτητας πάνω στα όρια, η μέγιστη διαφορά μεταξύ των αναπτυσσόμενων ταχυτήτων γειτονικών σημείων καθώς και η μέγιστη διαφορά ταχύτητας μεταξύ του ίδιου σημείου σε δύο διαδοχικά καρέ και τα αποτελέσματα αποθηκεύονται σε αρχείο μορφής *text*. Επιπλέον, χαράσσονται πάνω στην εικόνα βέλη που αποδίδουν την φορά και το μέτρο της κίνησης κάθε σημείου. Οι μετατοπίσεις υπολογίζονται ξεχωριστά προς κάθε έναν από τους δύο άξονες *x* και *y*.

Η εφαρμογή δοκιμάστηκε σε 19 ακολουθίες συμπτωματικών και ασυμπτωματικών εικόνων. Στις περισσότερες των περιπτώσεων η εκτίμηση της κίνησης ήταν σωστή. Συμπερασματικά, αυτή η νέα προσέγγιση μπορεί να βοηθήσει στον χαρακτηρισμό των πλακών αν και κάποιες βελτιώσεις πρέπει

να γίνουν προς την κατεύθυνση της στατιστικής επεξεργασίας των αποτελεσμάτων.

Λέξεις κλειδιά:

καρδιαγγειακές παθήσεις, εγκεφαλικό επεισόδιο, υπερηχογράφημα, αθηρωματικές πλάκες, καρωτίδα, οπτική ροή, εκτίμηση κίνησης

Abstract

"Study on the movement of atherosclerotic carotid plaques from a sequence of B-mode ultrasound images"

Alexandros C. Dimopoulos

Diploma Thesis, School of Electrical and Computer Engineering, National Technical University of Athens, Greece, 2004.

Cardiovascular diseases are the greatest cause of morbidity and mortality in the western world. There are indications that the morphology of atherosclerotic carotid plaques, obtained by ultrasound imaging, has prognostic implications. A large number of patients may be operated unnecessarily. The objective of this study was to develop a computer application that will facilitate the characterization of carotid plaques for the identification of individuals with asymptomatic carotid stenosis at risk of stroke. The movement of the carotid plaque was studied by the use of optical flow. This method is based on the assumption that a physical point has the same intensity of brightness in two successive frames of a sequence. Therefore, by processing the two frames, it is possible to predict the movement of a pixel. A dense field of motion vectors is calculated for every image based on local spatial and temporal derivatives of image brightness value.

The application is written in MS Visual Studio .NET C++ and is about 3.000 lines long. The user can open a sequence of tiff images (from ultrasound of carotid) and mark a particular Region Of Interest (ROI). Afterwards, the boundaries of the plaques must be marked. The application will calculate the speed of each point of the boundary. The maximum value of each point's speed, the maximum difference of speed for nearby points and the maximum difference of a point's speed in two successive frames are calculated and stored in a text file. Furthermore, arrows that indicate the direction and size of a pixel's velocity are drawn on the image. The displacement is calculated in axis x and y separately.

The application was tested on 19 sequences of both symptomatic and asymptomatic plaques. In most cases the prediction of displacement was successful. In conclusion, this new approach can aid in the characterization of carotid plaques even though some improvements should be made in the field of statistical process of the results.

Keywords:

Cardiovascular diseases, stroke, ultrasound, atherosclerotic plaques, carotid, optical flow, motion estimation