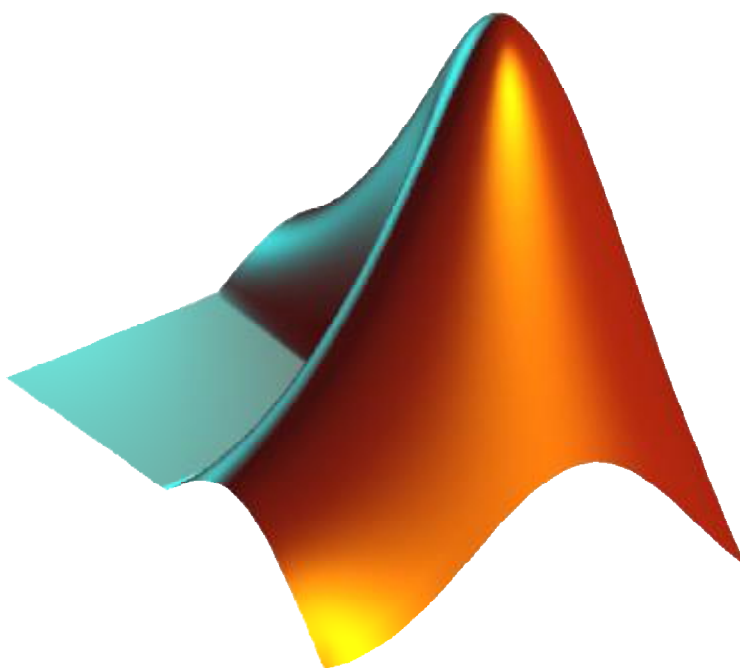


Τ.Ε.Ι. ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΜΜΕ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Δημιουργία Ολοκληρωμένου Γραφικού Περιβάλλοντος
Υποστήριξης Ψηφιακής Επεξεργασίας Εικόνας



ΠΑΡΑΣΚΕΥΑ ΜΑΡΙΑ

ΕΠΙΒΛΕΠΩΝ

Αθανάσιος Κούτρας, Επίκουρος Καθηγητής

ΠΥΡΓΟΣ
ΝΟΕΜΒΡΙΟΣ 2014

Περίληψη

Το Matlab είναι ένα προγραμματιστικό περιβάλλον που χρησιμοποιείται σήμερα σε όλους σχεδόν τους επιστημονικούς κλάδους για την κατασκευή εφαρμογών επίλυσης προβλημάτων, καθώς και για την κατασκευή γραφικών εκπαιδευτικών εφαρμογών.

Σκοπός της παρούσας εργασίας είναι η δημιουργία ενός ολοκληρωμένου εκπαιδευτικού γραφικού περιβάλλοντος με το Matlab για την υποβοήθηση εκμάθησης των βασικών εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας.

Για το σκοπό αυτό, σχεδιάστηκε και αναπτύχθηκε μια εφαρμογή Matlab, που εστιάζει στην τεχνική του χωρικού φιλτραρίσματος ψηφιακών εικόνων αποχρώσεων του γκρι. Η εφαρμογή διαθέτει γραφικές διεπαφές που σχεδιάστηκαν και υλοποιήθηκαν με το περιβάλλον GUIDE του Matlab 2012a και ενσωματώνουν συναρτήσεις της εργαλειοθήκης Image Processing Toolbox του Matlab. Η εφαρμογή έχει ανοικτή αρχιτεκτονική, γεγονός που την καθιστά εύκολα επεκτάσιμη.

Στις καινοτομίες της εφαρμογής συγκαταλέγονται α) η δυνατότητα που δίνει στον χρήστη να επιλέξει ένα *pixel* και μια περιοχή ενδιαφέροντος της εικόνας για να συγκρίνει τις τιμές έντασης φωτεινότητας των *pixel* πριν και μετά το φιλτράρισμα της εικόνας, β) η αναλυτικότερη επεξήγηση σε επίπεδο *pixel* της διαδικασίας του χωρικού φιλτραρίσματος και όλων των εμπλεκόμενων υπολογισμών για τα βασικότερα χωρικά φίλτρα και γ) η δυνατότητα χωρικού φιλτραρίσματος με χωρικές μάσκες που μπορεί να ορίσει ο ίδιος ο χρήστης.

Περιεχόμενα

Περίληψη.....	II
Περιεχόμενα	i
Κατάλογος Εικόνων	iv
Κατάλογος Σχημάτων.....	vi
Κατάλογος Πινάκων.....	vii
Εισαγωγή	viii
1 Θεωρητικό Πλαίσιο	1
1.1 Ψηφιακή Επεξεργασία Εικόνας	1
1.2 Ψηφιακές Εικόνες.....	2
1.3 Είδη Ψηφιακών Εικόνων	3
1.3.1 Δυαδικές Εικόνες (Binary).....	3
1.3.2 Εικόνες Αποχρώσεων του Γκρι (Grayscale).....	3
1.3.3 Έγχρωμες Εικόνες (RGB)	4
1.4 Ιστόγραμμα Εικόνας Αποχρώσεων του Γκρι.....	5
1.5 Τεχνικές Βελτίωσης Ψηφιακής Εικόνας Αποχρώσεων Γκρι.....	6
1.6 Χωρικό Φιλτράρισμα	8
1.6.1 Γραμμικά Χωρικά Φίλτρα	8
1.6.2 Μη Γραμμικά Χωρικά Φίλτρα Στατιστικής Διάταξης.....	11
1.6.3 Βασικά Χωρικά Φίλτρα	12
1.6.3.1 Φίλτρο Αριθμητικού Μέσου	12
1.6.3.2 Φίλτρο Γεωμετρικού Μέσου.....	13
1.6.3.3 Φίλτρο Gauss.....	13
1.6.3.4 Φίλτρο Ενδιάμεσης Τιμής	14
1.6.3.5 Φίλτρο Μεγίστου	15
1.6.3.6 Φίλτρο Ελαχίστου	15

1.6.3.7	Φίλτρο Περικεκομμένου Μέσου (Alpha Trimmed).....	16
1.6.3.8	Φίλτρο Wiener	16
1.6.3.9	Φίλτρα Παραγώγων	17
1.7	Εκπαιδευτικά Λογισμικά Ψηφιακής Επεξεργασίας Εικόνας	22
1.7.1	VcDemo.....	23
1.7.2	SIVA	24
1.7.3	ALMOT 2D DSP και 2D J-DSP	25
1.7.4	IP-Lab.....	27
1.7.5	Εφαρμογές Matlab.....	28
2	MATLAB.....	31
2.1	Βασικά Χαρακτηριστικά.....	31
2.2	Το Περιβάλλον GUIDE.....	31
2.3	Εργαλειοθήκη Επεξεργασίας Εικόνας (IP Toolbox)	41
3	Σχεδίαση και Υλοποίηση Εφαρμογής	44
3.1	Λειτουργικές Απαιτήσεις	45
3.2	Μη Λειτουργικές Απαιτήσεις.....	45
3.3	Σχεδίαση Γραφικών Διεπαφών.....	46
3.3.1	Το Βασικό Παράθυρο της Εφαρμογής.....	47
3.3.2	Τα Παράθυρα Επεξήγησης της Διαδικασίας Φιλτραρίσματος	54
3.4	Προγραμματισμός Συμπεριφοράς και Αλληλεπίδρασης.....	61
3.4.1	Προγραμματισμός του Βασικού Παραθύρου	63
3.4.1.1	Φόρτωση Εικόνας.....	63
3.4.1.2	Αποθήκευση Εικόνας	65
3.4.1.3	Τερματισμός Εφαρμογής	65
3.4.1.4	Επιλογή Περιοχής Ενδιαφέροντος της Αρχικής Εικόνας	66
3.4.1.5	Επιλογή Φίλτρου	70
3.4.1.6	Επιλογή Μεγέθους Μάσκας Φίλτρου	73

3.4.1.7	Επιλογή Λοιπών Παραμέτρων Φίλτρων.....	74
3.4.1.8	Εκτέλεση Φιλτραρίσματος	75
3.4.1.9	Επιλογή Ρίxel Επίδειξης Φιλτραρίσματος	82
3.4.1.10	Εμφάνιση Ιστογραμμάτων	84
3.4.1.11	Επεξήγηση Φιλτραρίσματος.....	85
3.4.2	Προγραμματισμός των Παραθύρων Επεξήγησης.....	86
3.4.2.1	Το Παράθυρο Επεξήγησης Βασικών Γραμμικών Φίλτρων.....	86
3.4.2.2	Το Παράθυρο Επεξήγησης Μη Γραμμικών Φίλτρων Στατιστικής Διάταξης	90
3.4.2.3	Το Παράθυρο Επεξήγησης Φίλτρου Περικεκομένου Μέσου (Alpha Trimmed) .	93
3.4.2.4	Το Παράθυρο Επεξήγησης Φίλτρου Wiener.....	96
3.4.2.5	Το Παράθυρο Επεξήγησης Φίλτρου Sobel (1 ^{ης} Παραγώγου).....	98
3.4.2.6	Το Παράθυρο Επεξήγησης Φίλτρου Laplace (2 ^{ης} Παραγώγου)	102
3.5	Βασικό Σενάριο Χρήσης Εφαρμογής.....	104
4	Συμπεράσματα	120
5	Προοπτικές - Μελλοντικές Βελτιώσεις	121
	Βιβλιογραφία	122

Κατάλογος Εικόνων

Εικόνα 1: Μία δυαδική εικόνα και η αναπαράστασή της ως πίνακας 0 και 1 (ROBOTIX, 2014).....	3
Εικόνα 2: Μία εικόνα αποχρώσεων του γκρι. Οι τιμές φωτεινότητας έχουν κανονικοποιηθεί (ROBOTIX, 2014).	4
Εικόνα 3: Μία έγχρωμη εικόνα. Οι τιμές φωτεινότητας των τριών χρωμάτων (RGB) είναι κανονικοποιημένες (ROBOTIX, 2014).....	5
Εικόνα 4: Μία εικόνα grayscale και το ιστόγραμμα της.....	6
Εικόνα 5: Μάσκες φίλτρου Laplace μεγέθους 3x3.....	18
Εικόνα 6: Μάσκες φίλτρου Laplace 1 (ενός) βήματος μεγέθους 3x3.....	19
Εικόνα 7: Μάσκες μεγέθους 3x3 για ενισχυτικό φιλτράρισμα	20
Εικόνα 8: Μάσκες φίλτρου Sobel 3x3.....	20
Εικόνα 9: Εκτεταμένες μάσκες φίλτρου Sobel μεγέθους 5x5.....	21
Εικόνα 10: Το περιβάλλον του VcDemo (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004) .24	
Εικόνα 11: Το εργαλείο χωρικού φιλτραρίσματος της συλλογής εφαρμογών SIVA (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).....	25
Εικόνα 12: Το περιβάλλον του ALMOT 2D DSP (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004)	26
Εικόνα 13: Το περιβάλλον του 2D J-DSP (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004) 26	
Εικόνα 14: Το περιβάλλον του IP-Lab (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004)	27
Εικόνα 15: Εφαρμογή Matlab για επίδειξη τεχνικών εξομάλυνσης/όξυνσης εικόνας (MathWorks.com, 2012)	28
Εικόνα 16: Εφαρμογή Matlab για επίδειξη φιλτραρίσματος στο πεδίο συχνότητας (MathWorks.com, 2012)	29
Εικόνα 17: Εφαρμογή Matlab για την επεξεργασία εικόνας (Krasula, K., et al, 2011).....	30
Εικόνα 18: Εφαρμογή Matlab για την επεξεργασία εικόνας (Krasula, K., et al, 2011).....	30
Εικόνα 19: Οδηγός δημιουργίας παραθύρου.....	33
Εικόνα 20: Περιβάλλον σχεδίασης νέου παραθύρου του GUIDE.....	33
Εικόνα 21: Αποθήκευση παραθύρου GUIDE.....	34
Εικόνα 22: Γραμμή εργαλείων διαχείρισης του GUIDE.....	36
Εικόνα 23: Παράθυρο Align Objects του GUIDE.....	38
Εικόνα 24: Παράθυρο Menu Editor του GUIDE.....	38
Εικόνα 25: Παράθυρο Tab Order Editor του GUIDE.....	39
Εικόνα 26: Παράθυρο Toolbar Editor του GUIDE.....	39
Εικόνα 27: Παράθυρο Editor του GUIDE.....	40
Εικόνα 28: Παράθυρο Property Inspector του GUIDE.....	40
Εικόνα 29: Παράθυρο Object Browser του GUIDE.....	41

Εικόνα 30: Το κεντρικό παράθυρο εφαρμογής, όπως φαίνεται στο <i>GUIDE</i> του <i>Matlab</i> . Τα τμήματα του παραθύρου εμφανίζονται με κόκκινη σήμανση	48
Εικόνα 31: Το βασικό παράθυρο (<i>BasicFiltering.fig</i>) κατά τον χρόνο εκτέλεσης	53
Εικόνα 32: Το παράθυρο επεξήγησης <i>explain_linear.fig</i> , όπως φαίνεται στο <i>GUIDE</i> του <i>Matlab</i> . Τα τμήματα του παραθύρου φαίνονται με μπλε σήμανση.	55
Εικόνα 33: Το παράθυρο επεξήγησης <i>explain_linear.fig</i> κατά τον χρόνο εκτέλεσης	59
Εικόνα 34: Το παράθυρο επεξήγησης <i>explain_alpha_trimmed.fig</i> κατά τον χρόνο εκτέλεσης	60
Εικόνα 35: Το παράθυρο επεξήγησης <i>explain_wiener.fig</i> κατά τον χρόνο εκτέλεσης	60
Εικόνα 36: Το παράθυρο επεξήγησης <i>explain_sobel.fig</i> κατά τον χρόνο εκτέλεσης	61
Εικόνα 37: Αρχική οθόνη εφαρμογής	105
Εικόνα 38: Πλήκτρο «Φόρτωση εικόνας».....	106
Εικόνα 39: Παράθυρο επιλογής και φόρτωσης εικόνας	106
Εικόνα 40: Ενημέρωση αρχικής οθόνης εφαρμογής μετά την φόρτωση της εικόνας.....	107
Εικόνα 41: Πλήκτρο «Επιλογή περιοχής»	108
Εικόνα 42: Παράθυρο επιλογής της περιοχής ενδιαφέροντος της αρχικής εικόνας.....	109
Εικόνα 43: Ενημέρωση αρχικής οθόνης εφαρμογής με τις τιμές φωτεινότητας των <i>pixels</i> της επιλεγμένης περιοχής ενδιαφέροντος.....	109
Εικόνα 44: Λίστα επιλογής χωρικού φίλτρου	110
Εικόνα 45: Παράμετροι φίλτρου <i>Gauss</i>	111
Εικόνα 46: Πλήκτρο «Εκτέλεση».....	112
Εικόνα 47: Ενημέρωση αρχικής οθόνης εφαρμογής με τα αποτελέσματα του φιλτραρίσματος	112
Εικόνα 48: Επιλογή του <i>pixel</i> (218, 187) της περιοχής ενδιαφέροντος και ενημέρωση της αρχικής οθόνης της εφαρμογής	114
Εικόνα 49: Σύγκριση εικόνας πριν και μετά το φιλτράρισμα	115
Εικόνα 50: Σύγκριση ιστογραμμάτων πριν και μετά το φιλτράρισμα	115
Εικόνα 51: Πλήκτρο «Επεξήγηση».....	116
Εικόνα 52: Παράθυρο επεξήγησης διαδικασίας φιλτραρίσματος με το φίλτρο <i>Gauss</i>	117
Εικόνα 53: Επιλογή «Αποθήκευση εικόνας»	118
Εικόνα 54: Παράθυρο αποθήκευσης εικόνας	118
Εικόνα 55: Επιλογή «Έξοδος»	119
Εικόνα 56: Επιβεβαίωση εξόδου από την εφαρμογή	119

Κατάλογος Σχημάτων

Σχήμα 1: Γραμμικό χωρικό φιλτράρισμα (με συσχέτιση) με χρήση γειτονιάς 3×3	9
Σχήμα 2: Φιλτράρισμα στα άκρα της εικόνας συμπληρώνοντας με 0 τις τιμές των pixels που βρίσκονται έξω από τα όρια της εικόνας κατά την εφαρμογή της μάσκας.	10
Σχήμα 3: Φιλτράρισμα στα άκρα της εικόνας επαναλαμβάνοντας τις τιμές των κοντινότερων pixels, όταν η μάσκα του φίλτρου εφαρμόζεται στα άκρα της εικόνας.	11
Σχήμα 4: Μάσκα φίλτρου αριθμητικού μέσου μεγέθους 3×3	12
Σχήμα 5: Μάσκα φίλτρου Gauss μεγέθους 3×3	14
Σχήμα 6: Επίδειξη φιλτραρίσματος με φίλτρο ενδιάμεσης τιμής μεγέθους 3×3	15

Κατάλογος Πινάκων

Πίνακας 1: Στοιχεία ελέγχου της εργαλειοθήκης του <i>GUIDE</i>	34
Πίνακας 2: Στοιχεία γραμμής εργαλείων διαχείρισης του <i>GUIDE</i>	37
Πίνακας 3: Βασικές συναρτήσεις της εργαλειοθήκης επεξεργασίας εικόνας.....	42
Πίνακας 4: Χειριστήρια του βασικού παραθύρου <i>BasicFiltering.fig</i>	48
Πίνακας 5: Χειριστήρια του παραθύρου επεξήγησης <i>explain_linear.fig</i>	55
Πίνακας 6: Οι καθολικές μεταβλητές της εφαρμογής.....	62

Εισαγωγή

Η ανάπτυξη του επιστημονικού κλάδου της ψηφιακής επεξεργασίας εικόνας οφείλεται κυρίως στη ραγδαία εξέλιξη των υπολογιστών. Η ψηφιακή επεξεργασία εικόνας βρίσκει εφαρμογές σε πολλούς σύγχρονους επιστημονικούς κλάδους, όπως της ιατρικής, της πληροφορικής, της φυσικής, συνεπώς θεωρείται απαραίτητο οι επιστήμονες των κλάδων αυτών να πρέπει να εκπαιδευτούν στο αντικείμενο της ψηφιακής επεξεργασίας εικόνας για να αποκτήσουν καλή θεωρητική, αλλά και πρακτική γνώση των τεχνικών της και των τρόπων αξιοποίησής τους. Για το σκοπό αυτόν, έχει αναπτυχθεί πληθώρα εργαλείων και εκπαιδευτικών λογισμικών - εφαρμογών, που στοχεύουν στην εκπαίδευση μέσω υπολογιστή των βασικών εννοιών, τεχνικών και εφαρμογών της ψηφιακής επεξεργασίας εικόνας.

Το Matlab αποτελεί ένα από τα δημοφιλέστερα προγραμματιστικά περιβάλλοντα που χρησιμοποιούνται για την ανάπτυξη αλληλεπιδραστικών εκπαιδευτικών εφαρμογών στην εκμάθηση εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας. Οι εφαρμογές αυτές είναι εύκολα επεκτάσιμες, λόγω της ανοικτής τους αρχιτεκτονικής, διαθέτουν εύχρηστες και φιλικές προς τον εκπαιδευόμενο γραφικές διεπαφές, παρέχουν άμεση οπτική ανατροφοδότηση και γενικά ενδείκνυνται για μάθηση μέσω προσομοιώσεων, πειραματισμού και παρατήρησης.

Στη συνέχεια ακολουθεί μία σύντομη περιγραφή της δομής της εργασίας.

Στην πρώτη ενότητα παραθέτουμε το θεωρητικό πλαίσιο της εργασίας, όπου γίνεται αναφορά στις βασικές έννοιες της ψηφιακής επεξεργασίας εικόνας, εστιάζοντας στην τεχνική του χωρικού φιλτραρίσματος και στα βασικότερα χωρικά φίλτρα. Επιπλέον, γίνεται αναφορά σε εκπαιδευτικά λογισμικά για την υποβοήθηση εκμάθησης των βασικών εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας.

Στη δεύτερη ενότητα γίνεται αναφορά στο Matlab και παρουσιάζονται συνοπτικά το περιβάλλον δημιουργίας γραφικών διεπαφών GUIDE και η εργαλειοθήκη επεξεργασίας εικόνας (Image Processing Toolbox) του Matlab. Τα εργαλεία αυτά

χρησιμοποιήθηκαν για τη σχεδίαση και την ανάπτυξη της εφαρμογής που πραγματεύεται η παρούσα εργασία, η οποία παρουσιάζεται στην Ενότητα 3.

Στην τρίτη ενότητα παρουσιάζεται η σχεδίαση και η ανάπτυξη της εφαρμογής επεξεργασίας εικόνας με το περιβάλλον GUIDE του Matlab 2012a. Η εφαρμογή ενσωματώνει συναρτήσεις της εργαλειοθήκης επεξεργασίας εικόνας (Image Processing Toolbox) του Matlab και εστιάζει στην αναλυτική επεξήγηση της τεχνικής του χωρικού φιλτραρίσματος εικόνων αποχρώσεων του γκρι, χρησιμοποιώντας τα βασικότερα χωρικά φίλτρα. Το καινοτόμο χαρακτηριστικό της εφαρμογής είναι η ανάλυση της διαδικασίας σε επίπεδο pixel και η επεξήγηση όλων των απαιτούμενων υπολογισμών σε κάθε βήμα της διαδικασίας. Στην ίδια ενότητα παρουσιάζεται ένας αναλυτικός οδηγός, που περιγράφει το βασικό σενάριο χρήσης της εφαρμογής.

Στην τέταρτη ενότητα παρουσιάζονται τα συμπεράσματα και τέλος στην πέμπτη ενότητα παρουσιάζονται οι προοπτικές και οι μελλοντικές βελτιώσεις της εφαρμογής.

1 Θεωρητικό Πλαίσιο

Στην ενότητα αυτή παρουσιάζεται τα βασικό θεωρητικό πλαίσιο της εργασίας, καλύπτοντας βασικές έννοιες και τεχνικές της ψηφιακής επεξεργασίας εικόνας και εστιάζοντας στην τεχνική του χωρικού φιλτραρίσματος. Επιπλέον, γίνεται αναφορά σε εκπαιδευτικά λογισμικά για την υποβοήθηση εκμάθησης των βασικών εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας.

1.1 Ψηφιακή Επεξεργασία Εικόνας

Η ψηφιακή επεξεργασία εικόνας είναι ένας επιστημονικός κλάδος που αναπτύχθηκε λόγω της ραγδαίας εξέλιξης των υπολογιστών. Ο όρος εικόνα περιλαμβάνει την αποτύπωση κάθε είδους πληροφορίας, λχ τα υπερηχογραφήματα, οι τομογραφίες, οι δορυφορικές φωτογραφίες κ.ο.κ, μπορούν να υποστούν επεξεργασία ως ψηφιακές εικόνας.

Η ψηφιακή επεξεργασία εικόνας έχει τους εξής στόχους:

- Την ψηφιοποίηση και κωδικοποίηση των εικόνων με σκοπό την αποθήκευση, μετάδοση και εκτύπωσή τους.
- Τη βελτίωση και την αποκατάσταση των εικόνων με σκοπό την καλύτερη απεικόνισή τους.
- Την ανάλυση και κατανόηση των εικόνων (Στρουθόπουλος, 2009).

Ο κλάδος της ψηφιακής επεξεργασίας εικόνας συνεργάζεται με τους εξής επιστημονικούς κλάδους:

- Ψηφιακή Επεξεργασία Σήματος
- Ρομποτική Όραση
- Τεχνητή Νοημοσύνη
- Αναγνώριση Προτύπων
- Νευρωνικά Δίκτυα
- Ασαφής Λογική
- Κωδικοποίηση
- Γραφικά Υπολογιστή

- Γραφιστική
- Ιατρική Πληροφορική
- Γεωπληροφορική
- Αστρονομία κοκ.

1.2 Ψηφιακές Εικόνες

Η ψηφιακή εικόνα είναι ένα πεπερασμένο σύνολο περιοχών, όπου κάθε περιοχή είναι χρωματισμένη με χρώμα που προέρχεται από ένα πεπερασμένο σύνολο. Στις περισσότερες των περιπτώσεων, μια ψηφιακή εικόνα είναι ένα ορθογώνιο, διαιρεμένο με γραμμές και στήλες σε ορθογώνιες περιοχές, που κάθε μία έχει συγκεκριμένο χρώμα. Μια τέτοια περιοχή ονομάζεται στοιχείο της εικόνας ή εικονοστοιχείο ή *pixel* (Στρουθόπουλος, 2009).

Εναλλακτικά, μία εικόνα μπορεί να οριστεί ως μία συνάρτηση δύο διαστάσεων $f(x,y)$, όπου x και y οι συντεταγμένες, και η τιμή της f για κάθε ζευγάρι συντεταγμένων (x,y) καλείται ως ένταση φωτεινότητας της εικόνας στο σημείο αυτό. Το σημείο αυτό, το οποίο έχει συντεταγμένες (x,y) , ουσιαστικά είναι ταυτόσημο με το *pixel*. Στις μονοχρωματικές εικόνες, με τις οποίες και θα ασχοληθούμε, συνηθίζεται να χρησιμοποιείται ο όρος «επίπεδο γκρι» ή «απόχρωση γκρι» (Gonzalez & Woods, 2007).

Μία αναλογική εικόνα έχει συνεχή πεδία τιμών x , y και συνεχείς τιμές φωτεινότητας. Για την ψηφιοποίηση μίας αναλογικής εικόνας, με άμεσο σκοπό την επεξεργασία της από τον υπολογιστή, απαιτείται η μετατροπή των συντεταγμένων x , y σε διακριτές τιμές (δειγματοληψία - *sampling*) και η μετατροπή των συνεχών τιμών φωτεινότητας σε διακριτές (κβάντιση - *quantization*). Με τον τρόπο αυτό προκύπτει μία ψηφιακή εικόνα.

Για παράδειγμα, μία εικόνα διαστάσεων $N \times M$ *pixels* θα αντιπροσωπεύεται από έναν δισδιάστατο πίνακα $N \times M$ στοιχείων. Λόγω της κβάντισης των επιπέδων του γκρι σε L επίπεδα, κάθε στοιχείο $f(x,y)$ της εικόνας θα περιορίζεται στο διάστημα $[0, L-1]$, με την τιμή 0 να αντιστοιχεί στο μαύρο χρώμα, την τιμή $L-1$ να αντιστοιχεί στο

άσπρο χρώμα, ενώ οι ενδιάμεσες τιμές να αντιστοιχούν σε κάποια ενδιάμεση απόχρωση του γκρι (Gonzalez & Woods, 2007).

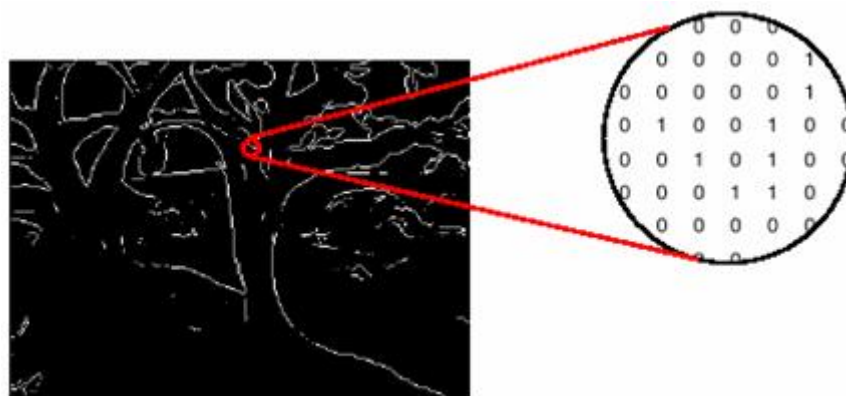
1.3 Είδη Ψηφιακών Εικόνων

Οι ψηφιακές εικόνες κατηγοριοποιούνται σε τρεις (3) βασικές κατηγορίες, ανάλογα με το πλήθος των χρωμάτων τους:

1. Δυαδικές (binary),
2. Ασπρόμαυρες ή αλλιώς αποχρώσεων του γκρι (grayscale) και
3. Έγχρωμες εικόνες (RGB),

1.3.1 Δυαδικές Εικόνες (Binary)

Σε μια δυαδική εικόνα κάθε pixel μπορεί να πάρει μια από τις δύο διακριτές τιμές (0 ή 1), με την τιμή 0 να αντιστοιχεί στο μαύρο και την τιμή 1 στο άσπρο. Στα λογισμικά επεξεργασίας εικόνας οι δυαδικές εικόνες αποθηκεύονται ως δισδιάστατοι πίνακες, που περιέχουν 0 και 1. Στην εικόνα που ακολουθεί (Εικόνα 1) φαίνεται μία δυαδική εικόνα και η αναπαράστασή της ως δυαδικός πίνακας.

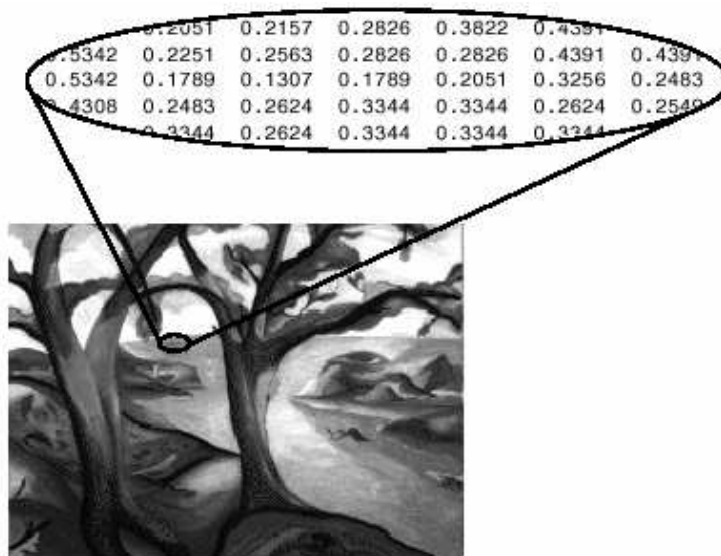


Εικόνα 1: Μία δυαδική εικόνα και η αναπαράστασή της ως πίνακας 0 και 1
(ROBOTIX, 2014)

1.3.2 Εικόνες Αποχρώσεων του Γκρι (Grayscale)

Μια εικόνα αποχρώσεων του γκρι είναι ένας πίνακας δεδομένων, με τις τιμές των στοιχείων να αναπαριστούν την ένταση της φωτεινότητας του κάθε pixel. Τα λογισμικά επεξεργασίας εικόνας (λχ. Matlab) αποθηκεύουν μια εικόνα αποχρώσεων του γκρι σε ένα δισδιάστατο πίνακα, του οποίου κάθε στοιχείο αναφέρεται σε ένα

και μοναδικό pixel. Οι τιμές των pixel αναπαριστούν τις διάφορες εντάσεις φωτεινότητας (επίπεδα του γκρι), οπότε κάθε pixel μπορεί να χρωματιστεί με μία από τις αποχρώσεις του γκρι, που παίρνουν τιμές από το μαύρο (τιμή 0) ως το άσπρο (τιμή 255). Συχνά, το διάστημα τιμών [0, 255] των φωτεινοτήτων κανονικοποιείται στο διάστημα [0, 1], όπως φαίνεται στην Εικόνα 2.



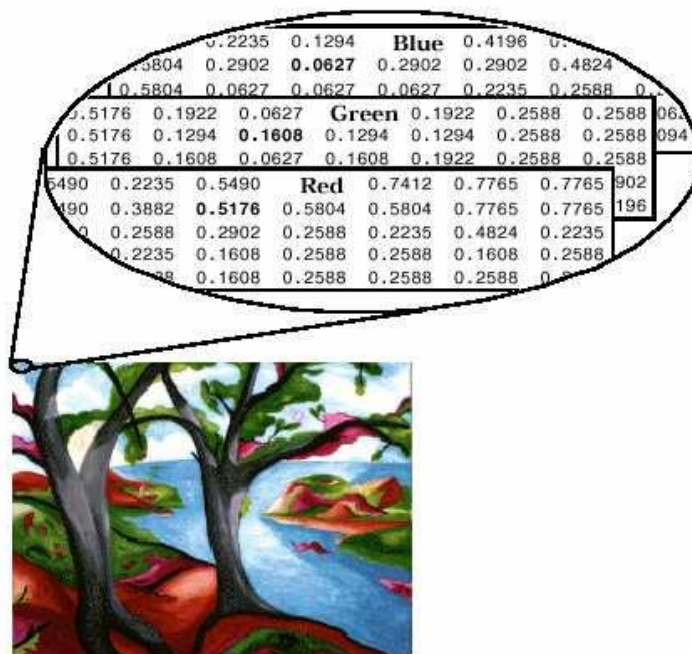
Εικόνα 2: Μία εικόνα αποχρώσεων του γκρι. Οι τιμές φωτεινότητας έχουν κανονικοποιηθεί (ROBOTIX, 2014).

1.3.3 Έγχρωμες Εικόνες (RGB)

Στις έγχρωμες (RGB) εικόνες κάθε pixel χρωματίζεται με χρώματα που προέρχονται από την ανάμειξη των αποχρώσεων του κόκκινου (Red), του πράσινου (Green) και του μπλε (Blue). Κάθε ένα από τα τρία αυτά χρώματα διαθέτει 256 αποχρώσεις, συνεπώς η φωτεινότητα καθενός pixel εκφράζεται από τρεις τιμές, που αντιπροσωπεύουν την ποσότητα του κόκκινου, του πράσινου και του μπλε χρώματος, ο συνδυασμός των οποίων δημιουργεί το τελικό χρώμα του pixel.

Στα λογισμικά επεξεργασίας εικόνας (λχ. Matlab), μια RGB εικόνα αποθηκεύεται ως ένας πίνακας δεδομένων με διαστάσεις $N \times M \times 3$, ο οποίος καθορίζει τα κόκκινα, πράσινα και μπλε χρωματικά στοιχεία κάθε pixel. Κάθε χρώμα λαμβάνει τιμές στο διάστημα [0, 255], ενώ συχνά το διάστημα αυτό κανονικοποιείται στο [0, 1], όπως φαίνεται στην Εικόνα 3. Έτσι, ένα pixel που έχει τις τιμές (0,0,0) απεικονίζει το

μαύρο, ένα pixel το οποίο έχει τιμές (1,1,1) απεικονίζει το λευκό, ενώ το pixel με τιμές (1,0,0) απεικονίζει το κόκκινο χρώμα (Gonzalez & Woods, 2007).



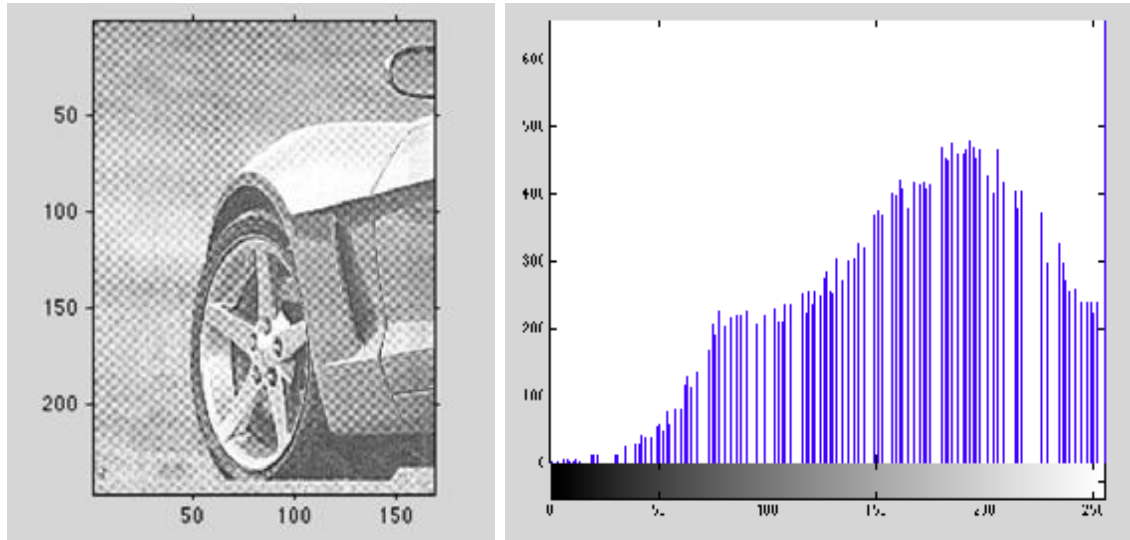
Εικόνα 3: Μία έγχρωμη εικόνα. Οι τιμές φωτεινότητας των τριών χρωμάτων (RGB) είναι κανονικοποιημένες (ROBOTIX, 2014).

1.4 Ιστόγραμμα Εικόνας Αποχρώσεων του Γκρι

Το ιστόγραμμα μιας ψηφιακής εικόνας είναι ένα γράφημα, όπου ο οριζόντιος άξονας αναπαριστά τα επίπεδα φωτεινότητας από 0 ως 255 και ο κατακόρυφος άξονας το πλήθος των pixels που αντιστοιχεί σε κάθε επίπεδο φωτεινότητας. Το ιστόγραμμα, λοιπόν, αποτελεί ένα γραφικό τρόπο αναπαράστασης της κατανομής των διαφορετικών αποχρώσεων του γκρι μιας εικόνας. Το γράφημα αυτό παρέχει γενικές πληροφορίες για τη μορφή και το περιεχόμενο της εικόνας, όπως το δυναμικό εύρος και την αντίθεση της εικόνας (σκοτεινή, φωτεινή, υψηλής ή χαμηλής αντίθεσης) (Στρουθόπουλος, 2009).

Ο κατακόρυφος άξονας μπορεί να κανονικοποιηθεί με βάση τη μέγιστη τιμή του ιστογράμματος, έτσι ώστε η μέγιστη τιμή να αντιστοιχεί στο 1, οπότε το κανονικοποιημένο ιστόγραμμα που προκύπτει εκφράζει την κατανομή της πυκνότητας πιθανότητας των αποχρώσεων του γκρι στην εικόνα.

Παρακάτω φαίνεται μία εικόνα αποχρώσεων του γκρι και το ιστόγραμμα της (Εικόνα 4).



Εικόνα 4: Μία εικόνα grayscale και το ιστόγραμά της

1.5 Τεχνικές Βελτίωσης Ψηφιακής Εικόνας Αποχρώσεων Γκρι

Οι τεχνικές βελτίωσης μίας ψηφιακής εικόνας βελτιώνουν την υποκειμενική ποιότητα της εικόνας τονίζοντας ορισμένα χαρακτηριστικά της και μειώνοντας τον θόρυβο. Οι τεχνικές αυτές κατατάσσονται σε τρεις (3) ομάδες:

1. Μετασχηματισμοί στο πεδίο του χώρου
2. Μετασχηματισμοί στο πεδίο της συχνότητας
3. Τεχνικές που συνδυάζουν στοιχεία των δύο προηγούμενων κατηγοριών

Η υλοποίηση και η εφαρμογή των τεχνικών βελτίωσης ποιότητας της εικόνας συμπεριλαμβάνει την χρήση δισδιάστατων ψηφιακών φίλτρων, που διακρίνονται στα γραμμικά και στα μη γραμμικά. Τα γραμμικά ψηφιακά φίλτρα μπορούν να σχεδιαστούν και να υλοποιηθούν είτε στο πεδίο των χωρικών συντεταγμένων είτε στο πεδίο των συχνοτήτων. Τα περισσότερα φίλτρα αποκατάστασης εικόνας είναι γραμμικά και υλοποιούνται στο πεδίο των συχνοτήτων, ενώ προφανώς υπάρχουν και μη γραμμικές τεχνικές. Ωστόσο, για την βελτίωση της ποιότητας της εικόνας χρησιμοποιούνται κυρίως υλοποιήσεις στο πεδίο του χώρου. Οι υλοποιήσεις αυτές είναι δισδιάστατα ψηφιακά φίλτρα, που έχουν περιοχές υποστήριξης οι οποίες αποτελούνται από ένα pixel ή μία ορθογώνια γειτονιά ενός pixel.

Οι βασικότερες τεχνικές βελτίωσης μίας ψηφιακής εικόνας είναι:

- Μετασχηματισμοί σημείου, που βελτιώνουν την αντίθεση της εικόνας. Οι βασικότεροι είναι:
 - Γραμμικές συναρτήσεις, όπως η κατασκευή αρνητικής εικόνας.
 - Λογαριθμικός, που συμπιέζει τις τιμές φωτεινότητας στο φωτεινό άκρο της κλίμακας, ενώ την επεκτείνει στο σκοτεινό άκρο.
 - Αντίστροφος λογαριθμικός, με το αντίθετο αποτέλεσμα από αυτό του λογαριθμικού μετασχηματισμού.
 - Ύψωσης σε δύναμη (μετασχηματισμοί γ). Όταν εκθέτης έχει τιμές μικρότερες της μονάδας, τότε οι μετασχηματισμοί αυτοί απεικονίζουν μία στενή περιοχή σκοτεινών τιμών σε μία ευρύτερη περιοχή φωτεινότερων τιμών. Το αντίστροφο συμβαίνει όταν ο εκθέτης είναι μεγαλύτερος της μονάδας.
 - Bell curve, που συμπιέζει τις ενδιάμεσες αποχρώσεις του γκρι και διευρύνει την κλίμακα στα δύο άκρα.
- Εξισορρόπηση ιστογράμματος, που βελτιώνει την αντίθεση της εικόνας σε περιοχές χαμηλής αντίθεσης, χρησιμοποιώντας το ιστόγραμμα. Με την τεχνική αυτή διασκορπίζεται το ιστόγραμμα της εικόνας, έτσι ώστε στην τελική εικόνα τα επίπεδα των φωτεινοτήτων να εκτείνονται σε μια ευρύτερη περιοχή της κλίμακας των τιμών φωτεινότητας.
- Φιλτράρισμα στο πεδίο των συχνοτήτων, που επιτρέπει την εξομάλυνση και/ή την όξυνση της εικόνας. Βασίζεται στην τροποποίηση του μετασχηματισμού Fourier της εικόνας και στον υπολογισμό του αντίστροφου μετασχηματισμού, ώστε να λάβουμε την τελική φιλτραρισμένη εικόνα.
- Χωρικό φιλτράρισμα, που επιτρέπει την εξομάλυνση και/ή την όξυνση της εικόνας. Βασίζεται στην χρήση χωρικών φίλτρων, όπως του φίλτρου αριθμητικού μέσου, του φίλτρου ενδιάμεσης τιμής κ.ο.κ.

Το υπόλοιπο της εργασίας επικεντρώνεται στο χωρικό φιλτράρισμα, επειδή σε αυτό εστιάζει η εφαρμογή που αναπτύχθηκε στο πλαίσιο της εργασίας.

1.6 Χωρικό Φιλτράρισμα

Το χωρικό φιλτράρισμα είναι μία βασική τεχνική βελτίωσης της ποιότητας μίας εικόνας, που βασίζεται στην εφαρμογή χωρικών φίλτρων στα pixels της αρχικής εικόνας, προκειμένου να πάρουμε την τελική φιλτραρισμένη εικόνα.

Ένα χωρικό φίλτρο, αποτελείται από μία ορθογώνια γειτονιά και μία προκαθορισμένη λειτουργία, η οποία εφαρμόζεται σε όλα τα pixels της εικόνας που περιλαμβάνονται στη γειτονιά αυτή. Η γειτονιά είναι συνήθως τετραγωνική, μεγέθους $n \times n$, όπου το n είναι συνήθως περιττός αριθμός. Μερικά τυπικά μεγέθη γειτονιών είναι τα 3×3 , 5×5 ή 7×7 κοκ (Gonzalez & Woods, 2007).

Η πράξη του φιλτραρίσματος, δημιουργεί ένα νέο pixel, με συντεταγμένες αυτές του κέντρου της γειτονιάς, του οποίου η τιμή της έντασης είναι το αποτέλεσμα του φιλτραρίσματος. Η διαδικασία αυτή επαναλαμβάνεται με το κέντρο του φίλτρου να σαρώνει όλα τα pixels της αρχικής εικόνας, αρχίζοντας από το πρώτο pixel, που βρίσκεται στην πάνω αριστερά γωνία της εικόνας, μέχρι το τελευταίο pixel, που βρίσκεται στην κάτω δεξιά γωνία της εικόνας. Εάν η λειτουργία του φίλτρου είναι μία γραμμική διαδικασία, τότε τόσο το φίλτρο όσο και η διαδικασία του φιλτραρίσματος χαρακτηρίζονται ως γραμμικά, αλλιώς χαρακτηρίζονται ως μη γραμμικά (Gonzalez & Woods, 2007).

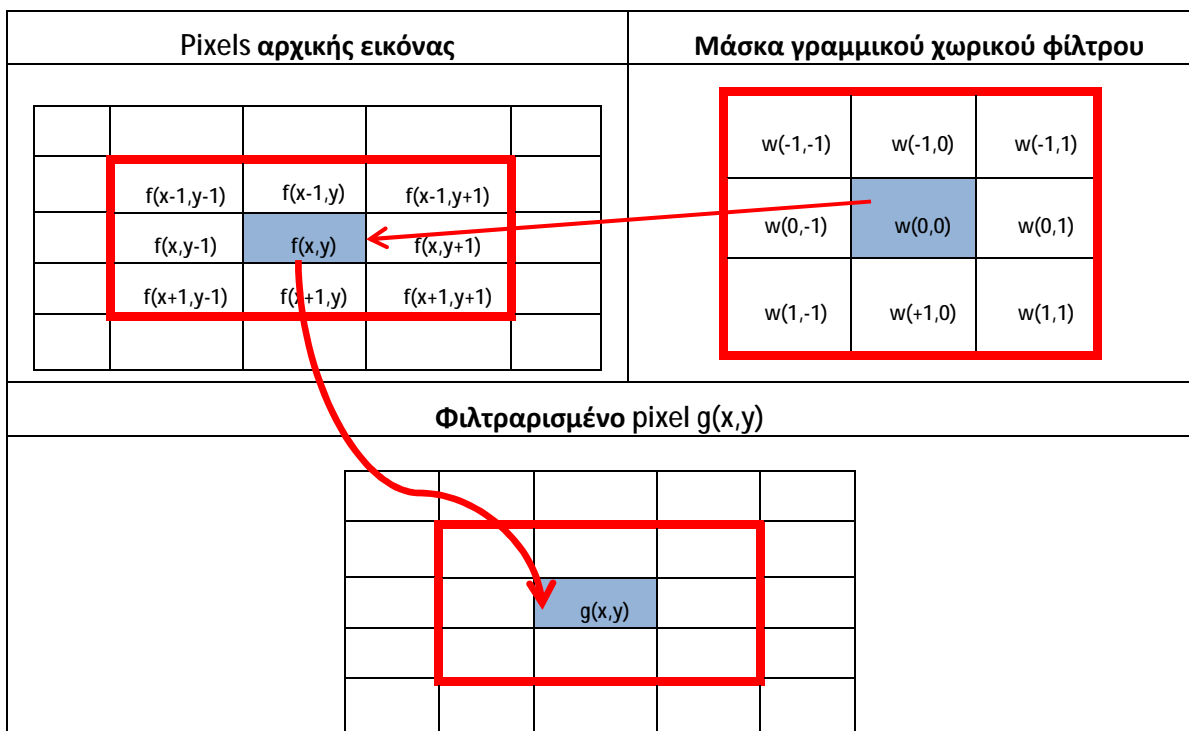
1.6.1 Γραμμικά Χωρικά Φίλτρα

Κατά την επεξεργασία με γραμμικά χωρικά φίλτρα, η τιμή καθενός pixel της φιλτραρισμένης εικόνας αποτελεί γραμμικό συνδυασμό των τιμών των pixels της αρχικής εικόνας, τα οποία βρίσκονται στη γειτονιά του αντίστοιχου pixel της αρχικής εικόνας. Το γραμμικό χωρικό φιλτράρισμα πραγματοποιείται με χρήση της δισδιάστατης συνέλιξης ή της δισδιάστατης συσχέτισης (CS425 Lab: Intensity Transformations and Spatial Filtering, 2011).

Στο παρακάτω παράδειγμα αναλύεται η διαδικασία του γραμμικού χωρικού φιλτραρίσματος με χρήση μίας γειτονιάς (μάσκα) διαστάσεων 3×3 . Η υλοποίηση γίνεται μέσω της συσχέτισης.

Έστω, λοιπόν, ότι πρόκειται να επεξεργαστούμε το pixel της αρχικής εικόνας που έχει τιμή $f(x,y)$, όπως φαίνεται και στο Σχήμα 1. Αρχικά το κεντρικό στοιχείο της μάσκας του φίλτρου, το οποίο έχει τιμή $w(0,0)$ έρχεται και ταυτίζεται με το pixel $f(x,y)$ της αρχικής εικόνας, το οποίο θέλουμε να επεξεργαστούμε. Με τον τρόπο αυτόν οριοθετείται η γειτονιά του pixel $f(x,y)$, της οποίας τα pixels θα συμμετάσχουν στο φιλτράρισμα. Η διαδικασία του γραμμικού χωρικού φιλτραρίσματος (μέσω της συσχέτισης) ορίζει ότι η τιμή έντασης του φιλτραρισμένου pixel υπολογίζεται ως το άθροισμα των γινομένων μεταξύ των συντελεστών του φίλτρου (μάσκας) και των τιμών έντασης φωτεινότητας των pixels της αρχικής εικόνας, τα οποία βρίσκονται στις αντίστοιχες θέσεις, κάτω από το φίλτρο. Πιο συγκεκριμένα, αν θεωρήσουμε ότι $g(x,y)$ θα είναι η φιλτραρισμένη τιμή του $f(x,y)$, τότε η $g(x,y)$ υπολογίζεται ως εξής:

$$g(x,y) = w(-1,-1) * f(x-1,y-1) + w(-1,0) * f(x-1,y) + w(-1,1) * f(x-1,y+1) \\ + w(0,-1) * f(x,y-1) + w(0,0) * f(x,y) + w(0,1) * f(x,y+1) \\ + w(1,-1) * f(x+1,y-1) + w(1,0) * f(x+1,y) + w(1,1) * f(x+1,y+1)$$



Σχήμα 1: Γραμμικό χωρικό φιλτράρισμα (με συσχέτιση) με χρήση γειτονιάς 3x3

Όταν πρόκειται να εφαρμοστεί το φίλτρο στα άκρα της εικόνας, τότε ένα τμήμα της μάσκας του φίλτρου βρίσκεται έξω από τα όρια της εικόνας. Στην περίπτωση αυτή, τα pixels της εικόνας που βρίσκονται έξω από τα όρια της εικόνας μπορούν να θεωρηθούν ότι έχουν την τιμή 0. Το γεγονός αυτό, οδηγεί στη δημιουργία ενός σκοτεινού περιγράμματος στα όρια της εικόνας κατά τη διαδικασία του φιλτραρίσματος (Weisstein, 2014). Στο παρακάτω σχήμα (Σχήμα 2) φαίνεται η συμπλήρωση με 0, όταν το φίλτρο εφαρμόζεται στα pixels που βρίσκονται στα άκρα της εικόνας.

Pixels αρχικής εικόνας, με προσθήκη 0 έξω από τα όρια	Μάσκα φίλτρου μέσου όρου διάστασης 3x3																																												
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td><td></td><td>0</td><td>0</td><td>0</td><td></td><td></td> </tr> <tr> <td>20</td><td>20</td><td>10</td><td style="background-color: #d9e1f2;">40</td><td>50</td><td></td><td></td> </tr> <tr> <td>30</td><td>30</td><td>20</td><td>30</td><td>30</td><td></td><td></td> </tr> <tr> <td>30</td><td>30</td><td>20</td><td>10</td><td>10</td><td></td><td></td> </tr> <tr> <td>40</td><td>40</td><td>30</td><td>40</td><td>50</td><td></td><td></td> </tr> </table>			0	0	0			20	20	10	40	50			30	30	20	30	30			30	30	20	10	10			40	40	30	40	50			<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td>1/9</td><td>1/9</td><td>1/9</td> </tr> <tr> <td>1/9</td><td style="background-color: #d9e1f2;">1/9</td><td>1/9</td> </tr> <tr> <td>1/9</td><td>1/9</td><td>1/9</td> </tr> </table>	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
		0	0	0																																									
20	20	10	40	50																																									
30	30	20	30	30																																									
30	30	20	10	10																																									
40	40	30	40	50																																									
1/9	1/9	1/9																																											
1/9	1/9	1/9																																											
1/9	1/9	1/9																																											
Φιλτραρισμένο pixel																																													
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td><td></td><td style="background-color: #d9e1f2;">20</td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td> </tr> </table>			20																		<p>Υπολογισμός φιλτραρισμένης τιμής:</p> $ \begin{aligned} & 1/9*0 + 1/9*0 + 1/9*0 \\ & + 1/9*10 + 1/9*40 + 1/9*50 \\ & + 1/9*20 + 1/9*30 + 1/9*30 \\ & = 20 \end{aligned} $																								
		20																																											

Σχήμα 2: Φιλτράρισμα στα άκρα της εικόνας συμπληρώνοντας με 0 τις τιμές των pixels που βρίσκονται έξω από τα όρια της εικόνας κατά την εφαρμογή της μάσκας.

Μία λύση στο προηγούμενο φαινόμενο δημιουργίας σκοτεινών περιγραμμάτων στα άκρα της εικόνας, είναι η επανάληψη των τιμών των pixel που βρίσκονται στα άκρα της εικόνας (Weisstein, 2014). Έτσι, οι τιμές των pixels της γειτονιάς τα οποία βρίσκονται έξω από τα όρια της εικόνας, παίρνουν την τιμή του κοντινότερου pixel. Στο παρακάτω σχήμα (Σχήμα 3) φαίνεται η επανάληψη των τιμών των pixels, όταν το φίλτρο εφαρμόζεται στα pixels που βρίσκονται στα άκρα της εικόνας.

Pixels αρχικής εικόνας, με επανάληψη των τιμών των κοντινότερων pixels	Μάσκα φίλτρου αριθμητικού μέσου διάστασης 3x3																																		
<table border="1"> <tr><td></td><td></td><td>10</td><td>40</td><td>50</td></tr> <tr><td>20</td><td>20</td><td>10</td><td>40</td><td>50</td></tr> <tr><td>30</td><td>30</td><td>20</td><td>30</td><td>30</td></tr> <tr><td>30</td><td>30</td><td>20</td><td>10</td><td>10</td></tr> <tr><td>40</td><td>40</td><td>30</td><td>40</td><td>50</td></tr> </table>			10	40	50	20	20	10	40	50	30	30	20	30	30	30	30	20	10	10	40	40	30	40	50	<table border="1"> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> <tr><td>1/9</td><td>1/9</td><td>1/9</td></tr> </table>	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
		10	40	50																															
20	20	10	40	50																															
30	30	20	30	30																															
30	30	20	10	10																															
40	40	30	40	50																															
1/9	1/9	1/9																																	
1/9	1/9	1/9																																	
1/9	1/9	1/9																																	
Φιλτραρισμένο pixel																																			
<table border="1"> <tr><td></td><td></td><td>31</td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>			31														<p>Υπολογισμός φιλτραρισμένης τιμής:</p> $ \begin{aligned} & 1/9 * 10 + 1/9 * 40 + 1/9 * 50 \\ & + 1/9 * 10 + 1/9 * 40 + 1/9 * 50 \\ & + 1/9 * 20 + 1/9 * 30 + 1/9 * 30 \\ & = 31,111 \Rightarrow 31 \end{aligned} $																		
		31																																	

Σχήμα 3: Φιλτράρισμα στα άκρα της εικόνας επαναλαμβάνοντας τις τιμές των κοντινότερων pixels, όταν η μάσκα του φίλτρου εφαρμόζεται στα άκρα της εικόνας.

Στο σημείο αυτό κρίνεται σκόπιμο να αναφέρουμε ότι το αποτέλεσμα εφαρμογής ενός γραμμικού χωρικού φίλτρου ενδέχεται να προκύψει:

- Δεκαδικό, οπότε στρογγυλοποιείται στον πλησιέστερο ακέραιο.
- Αρνητικό, οπότε παίρνει την τιμή 0 (ελάχιστη τιμή κλίμακας γκρι).
- Μεγαλύτερο του 255 (μέγιστη τιμή κλίμακας γκρι), οπότε παίρνει την τιμή 255.

1.6.2 Μη Γραμμικά Χωρικά Φίλτρα Στατιστικής Διάταξης

Τα φίλτρα αυτά βασίζονται στην ταξινόμηση των τιμών φωτεινοτήτων των pixels της χρησιμοποιούμενης γειτονιάς και στην αντικατάσταση της τιμής του κεντρικού pixel με την τιμή που προσδιορίζεται από το αποτέλεσμα της ταξινόμησης. Η διαδικασία της ταξινόμησης είναι μη γραμμική πράξη, συνεπώς τα φίλτρα της κατηγορίας αυτής χαρακτηρίζονται ως μη γραμμικά (Gonzalez & Woods, 2007).

Τα φίλτρα στατιστικής διάταξης έχουν πολύ καλά αποτελέσματα στην μείωση ορισμένων τύπων τυχαίου θορύβου, προκαλώντας πολύ μικρότερη θόλωση σε σχέση με τα γραμμικά φίλτρα παρόμοιου μεγέθους (Gonzalez & Woods, 2007). Τα βασικότερα φίλτρα στατιστικής διάταξης είναι το φίλτρο ενδιάμεσης τιμής, μεγίστου, ελαχίστου, περικεκομμένου μέσου κοκ.

1.6.3 Βασικά Χωρικά Φίλτρα

1.6.3.1 Φίλτρο Αριθμητικού Μέσου

Το φίλτρο του αριθμητικού μέσου (ή μέσου όρου ή μέσης τιμής) υπολογίζει το μέσο όρο των τιμών των pixels μίας γειτονιάς με κέντρο ένα pixel (x,y). Η φιλτραρισμένη τιμή του pixel (x,y) είναι ο μέσος όρος που υπολογίστηκε. Εάν υποθέσουμε ότι η χρησιμοποιούμενη γειτονιά είναι μεγέθους nxm, τότε το φίλτρο αυτό μπορεί να υλοποιηθεί με μία μάσκα μεγέθους nxm, όπου καθένας της συντελεστής ισούται με 1/(n*m). Στην απλούστερη περίπτωση, που η γειτονιά έχει μέγεθος 3x3, η μάσκα του φίλτρου είναι αυτή του παρακάτω σχήματος (Σχήμα 4):

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

ή αλλιώς $1/9 *$

1	1	1
1	1	1
1	1	1

Σχήμα 4: Μάσκα φίλτρου αριθμητικού μέσου μεγέθους 3x3

Το φίλτρο αριθμητικού μέσου χρησιμοποιείται για την μείωση λευκού προσθετικού θορύβου, θορύβου τύπου Gauss και ομοιόμορφου θορύβου, εξομαλύνοντας τις τοπικές μεταβολές της εικόνας, με βασικό μειονέκτημα ότι επιφέρει θόλωση στην εικόνα. Η θόλωση αυτή είναι ανάλογη των διαστάσεων της μάσκας του φίλτρου. Ανάλογα αποτελέσματα μπορούμε να επιτύχουμε στο πεδίο των συχνοτήτων με χρήση χαμηλοπερατών φίλτρων.

Ωστόσο, το φίλτρο αριθμητικού μέσου έχει τα εξής μειονεκτήματα:

- Η θόλωση που επιφέρει παραμορφώνει τυχόν λεπτομέρειες της εικόνας, όπως γωνίες και ακμές.

- Αν στη γειτονιά του pixel που εξετάζεται, υπάρχει έστω και ένα pixel με ακραία τιμή ή γενικότερα με μη αντιπροσωπευτική τιμή, τότε μπορεί να επηρεάσει σημαντικά τον υπολογισμό του αριθμητικού μέσου.
- Όταν το φίλτρο σαρώνει τα άκρα της εικόνας, τότε το μέρος της γειτονιάς που βρίσκεται έξω από τα όρια της εικόνας θα συμπληρωθεί από κάποιες τιμές (συνήθως 0), δημιουργώντας έτσι ένα σκοτεινό και θολό περίγραμμα στην εικόνα (Kim, 2013).

1.6.3.2 Φίλτρο Γεωμετρικού Μέσου

Το φίλτρο αυτό υπολογίζει το γεωμετρικό μέσο των τιμών φωτεινότητας των pixels που βρίσκονται στη γειτονιά, που έχει ως κέντρο της το εξεταζόμενο pixel (x,y) . Ο γεωμετρικός μέσος των φωτεινοτήτων μίας γειτονιάς μεγέθους $n \times m$, υπολογίζεται ως το γινόμενο των φωτεινοτήτων των pixels της γειτονιάς, υψωμένο στη δύναμη $1/(n \cdot m)$. Συνεπώς, το φίλτρο αυτό ανήκει στα μη γραμμικά χωρικά φίλτρα.

Η εξομάλυνση που επιφέρει το φίλτρο είναι ανάλογη των διαστάσεων της γειτονιάς. Αν και η εξομάλυνση αυτή είναι συγκρίσιμη με αυτή του αριθμητικού μέσου, ωστόσο, το συγκεκριμένο φίλτρο τείνει να εξαλείφει λιγότερες λεπτομέρειες. Επιπλέον, το φίλτρο γεωμετρικού μέσου είναι καλύτερο από το φίλτρο αριθμητικού μέσου στην αφαίρεση θορύβου τύπου Gauss. Ωστόσο, έχει την τάση να μεγαλώνει τις μαύρες περιοχές και να δημιουργεί ένα μαύρο περίγραμμα στα όρια της εικόνας, δεδομένου ότι έστω και μία τιμή φωτεινότητας αν είναι 0, τότε ο γεωμετρικός μέσος θα είναι 0 (Gonzalez & Woods, 2007).

1.6.3.3 Φίλτρο Gauss

Τα φίλτρα τύπου Gauss είναι χαμηλοπερατά φίλτρα, τα οποία επιτρέπουν τη μείωση του θορύβου με το τίμημα ότι επιφέρουν θόλωμα στην εικόνα. Για τον υπολογισμό των συντελεστών της μάσκας ενός φίλτρου Gauss χρησιμοποιούμε τη συνάρτηση:
$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$
, όπου σ είναι η τυπική απόκλιση και x,y ακέραιοι αριθμοί, οι οποίοι αντιστοιχούν στις συντεταγμένες θέσης των συντελεστών της μάσκας (Kim, 2013).

Για παράδειγμα, έστω ότι θέλουμε να κατασκευάσουμε μία μάσκα h , διαστάσεων 3×3 με χρήση της παραπάνω συνάρτησης Gauss, όταν $\sigma=0.5$. Έτσι, θα έχουμε:

$$h1 = h(-1, -1) = \frac{1}{2 * 3.14 * 0.5^2} e^{-\frac{(-1)^2 + (-1)^2}{2 * 0.5^2}} = 0.0117$$

$$h2 = h(-1, 0) = \frac{1}{2 * 3.14 * 0.5^2} e^{-\frac{(-1)^2 + (0)^2}{2 * 0.5^2}} = 0.0862$$

$$h3 = h(-1, 1) = \frac{1}{2 * 3.14 * 0.5^2} e^{-\frac{(-1)^2 + (1)^2}{2 * 0.5^2}} = 0.0117$$

ΚΟΚ.

Συνοψίζοντας, τελικά η μάσκα είναι αυτή που φαίνεται παρακάτω, στο Σχήμα 5:

$h =$	h1	h2	h3	$=$	0.0117	0.0862	0.0117
	h4	h5	h6		0.0862	0.6366	0.0862
	h7	h8	h9		0.0117	0.0862	0.0117

Σχήμα 5: Μάσκα φίλτρου Gauss μεγέθους 3×3

1.6.3.4 Φίλτρο Ενδιάμεσης Τιμής

Είναι το σημαντικότερο μη γραμμικό φίλτρο στατιστικής διάταξης. Αρχικά διατάσσει σε αύξουσα σειρά τις τιμές φωτεινότητας των pixels της χρησιμοποιούμενης γειτονιάς και στη συνέχεια αντικαθιστά την τιμή του κεντρικού pixel της γειτονιάς με την ενδιάμεση τιμή των φωτεινοτήτων. Αν η γειτονιά είναι διαστάσεων $n \times m$, τότε η ενδιάμεση τιμή θα βρίσκεται στη θέση $(n * m \div 2) + 1$.

Για παράδειγμα, ας θεωρήσουμε ότι η γειτονιά είναι διαστάσεων 3×3 και έστω ότι οι τιμές φωτεινότητας των pixels της εικόνας στη γειτονιά αυτή είναι αυτές που φαίνονται στο κόκκινο πλαίσιο του σχήματος που ακολουθεί (Σχήμα 6).

Pixels αρχικής εικόνας					Φιλτράρισμα με φίλτρο ενδιάμεσης τιμής				
200	200	200	200	200					
200	200	200	200	200					
200	200	10	200	200			200		
200	200	200	200	200					
200	200	200	200	200					

Σχήμα 6: Επίδειξη φιλτραρίσματος με φίλτρο ενδιάμεσης τιμής μεγέθους 3x3

Αρχικά τα pixels της γειτονιάς ταξινομούνται σε αύξουσα σειρά ως εξής: {10, 200, 200, 200, 200, 200, 200, 200, 200} και στη συνέχεια το κεντρικό pixel της γειτονιάς που έχει τιμή 10, αντικαθίσταται από την τιμή που βρίσκεται στην ενδιάμεση θέση της διάταξης (5^η θέση), δηλαδή από την τιμή 200.

Το φίλτρο ενδιάμεσης τιμής διατηρεί άθικτες τις λεπτομέρειες και τα περιγράμματα της εικόνας, ενώ είναι πολύ αποτελεσματικό για την αφαίρεση του κρουστικού θορύβου, που εμφανίζεται ως μαύρες και άσπρες κουκκίδες στην εικόνα (θόρυβος αλατοπίπερο). Ωστόσο, ένα βασικό μειονέκτημα του συγκεκριμένου φίλτρου είναι ότι μπορεί να εξαφανίσει τις πολύ λεπτές γραμμές και τις γωνίες μίας εικόνας, γεγονός που διορθώνεται με την χρήση γειτονιάς σχήματος σταυρού (Gonzalez & Woods, 2007).

1.6.3.5 Φίλτρο Μεγίστου

Είναι μη γραμμικό φίλτρο στατιστικής διάταξης. Αρχικά διατάσσει σε αύξουσα σειρά τις τιμές φωτεινότητας των pixels της χρησιμοποιούμενης γειτονιάς και στη συνέχεια αντικαθιστά την τιμή του κεντρικού pixel της γειτονιάς με τη μέγιστη τιμή των φωτεινοτήτων.

Χρησιμοποιείται για την αφαίρεση μαύρων κουκκίδων (θόρυβος πιπεριού), αφού έχει την ιδιότητα να επεκτείνει τις φωτεινές περιοχές και να περιορίζει τις σκοτεινές.

1.6.3.6 Φίλτρο Ελαχίστου

Είναι μη γραμμικό φίλτρο στατιστικής διάταξης. Αρχικά διατάσσει σε αύξουσα σειρά τις τιμές φωτεινότητας των pixels της χρησιμοποιούμενης γειτονιάς και στη

συνέχεια αντικαθιστά την τιμή του κεντρικού pixel της γειτονιάς με την ελάχιστη τιμή των φωτεινότητων.

Χρησιμοποιείται για την αφαίρεση λευκών κουκκίδων (θόρυβος αλατιού), αφού έχει την ιδιότητα να επεκτείνει τις σκοτεινές περιοχές και να περιορίζει τις φωτεινές.

1.6.3.7 Φίλτρο Περικεκομμένου Μέσου (Alpha Trimmed)

Ανήκει στην κατηγορία των μη γραμμικών φίλτρων στατιστικής διάταξης, συνδυάζοντας τα οφέλη του φίλτρου ενδιάμεσης τιμής και του φίλτρου αριθμητικού μέσου. Δοθείσας μίας θετικής άρτιας τιμής για το α και μίας γειτονιάς διαστάσεων $n \times m$, το φίλτρο αυτό λειτουργεί ως εξής:

- Αρχικά διατάσσονται σε αύξουσα σειρά οι $n \times m$ τιμές φωτεινότητας των pixels της γειτονιάς.
- Απομακρύνονται οι $\alpha/2$ μικρότερες τιμές και οι $\alpha/2$ μεγαλύτερες τιμές, δηλαδή συνολικά αφαιρούνται α τιμές.
- Υπολογίζεται ο μέσος όρος των τιμών που έχουν απομείνει, δηλαδή των $(n \times m - \alpha)$ τιμών.
- Η τιμή που υπολογίστηκε αντικαθιστά την τιμή του κεντρικού pixel της γειτονιάς.

Η παράμετρος α μπορεί να πάρει άρτιες τιμές στο διάστημα $[0, n \times m - 1]$. Αν $\alpha = 0$, τότε το φίλτρο ταυτίζεται με το φίλτρο αριθμητικού μέσου, ενώ αν είναι $\alpha = n \times m - 1$, τότε το φίλτρο ταυτίζεται με το φίλτρο ενδιάμεσης τιμής. Για τις υπόλοιπες τιμές του α , το φίλτρο είναι πολύ χρήσιμο για την αφαίρεση πολλαπλών τύπων θορύβου, όπως συνδυασμού κρουστικού θορύβου και θορύβου τύπου Gauss (Gonzalez & Woods, 2007).

1.6.3.8 Φίλτρο Wiener

Πρόκειται για μη γραμμικό χαμηλοπερατό φίλτρο που βασίζεται σε στατιστικά μέτρα της εικόνας και της χρησιμοποιούμενης γειτονιάς καθενός pixel της εικόνας. Είναι πολύ αποτελεσματικό στην αφαίρεση προσθετικού θορύβου τύπου Gauss.

Θεωρώντας ότι η γειτονιά μας είναι διαστάσεων $n \times m$, τότε το φίλτρο Wiener χρησιμοποιεί τις εξής παραμέτρους:

- Την τιμή φωτεινότητας του υπό μελέτη pixel $f(x,y)$ της αρχικής αλλοιωμένης εικόνας.
- Τον τοπικό μέσο όρο μ_{local} , καθώς και την τοπική διασπορά σ_{local}^2 των τιμών φωτεινότητας των pixel της γειτονιάς, η οποία έχει ως κέντρο το υπό μελέτη pixel $f(x,y)$.
- Την ολική διασπορά V^2 του θορύβου που αλλοιώνει την εικόνα. Επειδή η παράμετρος αυτή είναι δύσκολο να υπολογιστεί, μπορούμε να την θεωρήσουμε ίση με τον μέσο όρο όλων των τοπικών διασπορών που έχουμε υπολογίσει (Gonzalez & Woods, 2007).

Η τιμή καθενός φιλτραρισμένου pixel $g(x,y)$ με βάση το φίλτρο Wiener δίνεται από την εξής σχέση: $g(x,y) = \mu_{local} + \frac{\sigma_{local}^2 - V^2}{\sigma_{local}^2} * (f(x,y) - \mu_{local})$

Στην προηγούμενη σχέση, η ποσότητα $\sigma_{local}^2 - V^2$ πρέπει να είναι πάντα θετική, συνεπώς απαιτείται να ενσωματώσουμε έναν έλεγχο, έτσι ώστε όταν η εν λόγω ποσότητα προκύπτει αρνητική, τότε ο λόγος $\frac{\sigma_{local}^2 - V^2}{\sigma_{local}^2}$ να τίθεται ίσος με τη μονάδα, προκειμένου να αποφεύγονται οι αρνητικές τιμές (Gonzalez & Woods, 2007).

1.6.3.9 Φίλτρα Παραγώγων

Τα φίλτρα παραγώγων είναι υπερπερατά φίλτρα, που χρησιμοποιούνται για την αύξηση της οξύτητας της εικόνας και για την ανίχνευση ακμών. Έχουν το βασικό χαρακτηριστικό ότι ενισχύουν τις ακμές και τις ασυνέχειες της εικόνας, τονίζοντας τις απότομες μεταβολές φωτεινότητας (που συνήθως εμφανίζονται στα όρια αντικειμένων), ενώ υποβαθμίζουν τις περιοχές στις οποίες οι τιμές φωτεινότητας των pixels μεταβάλλονται με αργό ρυθμό. Τα φίλτρα αυτά χρησιμοποιούν τις παραγώγους 1^{ης} και 2^{ης} τάξης (Gonzalez & Woods, 2007).

Οι χωρικές μάσκες που χρησιμοποιούνται για τα φίλτρα 2^{ης} παραγώγου (φίλτρα Laplace) έχουν τις εξής ιδιότητες:

- Το άθροισμα των συντελεστών της μάσκας είναι ίσο με 0
- Η τιμή του κεντρικού στοιχείου της μάσκας είναι θετική και των γειτόνων του αρνητική ή το αντίστροφο.

Στην παρακάτω εικόνα (Εικόνα 5) φαίνονται οι μάσκες των φίλτρων 2^{ης} παραγώγου (Laplace). Οι μάσκες της πρώτης γραμμής είναι οι βασικές υλοποιήσεις των φίλτρων Laplace, όπου το κεντρικό στοιχείο της μάσκας έχει αρνητικό πρόσημο, και τα γειτονικά είναι θετικά, ενώ στη δεύτερη γραμμή φαίνονται οι εναλλακτικές υλοποιήσεις των φίλτρων Laplace, όπου ακολουθείται η αντίστροφη λογική (Gonzalez & Woods, 2007).

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Εικόνα 5: Μάσκες φίλτρου Laplace μεγέθους 3x3

Η εφαρμογή των φίλτρων Laplace δημιουργεί εικόνες με γκρίζες ακμές-γραμμές και ασυνέχειες, πάνω σε ένα σκοτεινό φόντο. Προκειμένου να επαναφέρουμε το φόντο, διατηρώντας ταυτόχρονα την όξυνση που προκαλείται από την εφαρμογή της μάσκας Laplace, θα πρέπει να προσθέσουμε ή να αφαιρέσουμε την επεξεργασμένη εικόνα κατά Laplace, από την αρχική εικόνα, ανάλογα με το πρόσημο του στοιχείου της μάσκας Laplace που χρησιμοποιήσαμε (Gonzalez & Woods, 2007). Πιο συγκεκριμένα:

- Αν έχουμε χρησιμοποιήσει κάποια από τις μάσκες της πρώτης γραμμής της προηγούμενης εικόνας (Εικόνα 5), όπου το κεντρικό στοιχείο της μάσκας

έχει αρνητικό πρόσημο, τότε πρέπει να αφαιρέσουμε από την αρχική εικόνα μας την επεξεργασμένη εικόνα κατά Laplace.

- Αν έχουμε χρησιμοποιήσει κάποια από τις μάσκες της δεύτερης γραμμής της προηγούμενης εικόνας (Εικόνα 5), όπου το κεντρικό στοιχείο της μάσκας έχει θετικό πρόσημο, τότε πρέπει να προσθέσουμε στην αρχική εικόνα μας την επεξεργασμένη εικόνα κατά Laplace.

Η διαδικασία που περιγράψαμε παραπάνω μπορεί να υλοποιηθεί απευθείας (σε ένα μόνο βήμα) με τη χρήση μίας κατάλληλης χωρικής μάσκας, αποφεύγοντας αρχικά τον υπολογισμό της οξυμένης κατά Laplace εικόνας και μετά την πρόσθεσή της (ή αφαίρεσή της) από την αρχική εικόνα (Gonzalez & Woods, 2007). Παρακάτω (Εικόνα 6) φαίνονται οι μάσκες αυτές, που χαρακτηρίζονται και ως μάσκες Laplace ενός βήματος.

0	-1	0	-1	-1	-1
-1	5	-1	-1	9	-1
0	-1	0	-1	-1	-1

Εικόνα 6: Μάσκες φίλτρου Laplace 1 (ενός) βήματος μεγέθους 3x3

Ένα άλλο βασικό χαρακτηριστικό των φίλτρων παραγώγων είναι ότι επειδή το άθροισμα των συντελεστών της μάσκας τους είναι ίσο με 0, οι ομοιόμορφες περιοχές της εικόνας γίνονται σκοτεινές όπως προαναφέραμε. Προκειμένου να αντιμετωπίσουμε το φαινόμενο αυτό μπορούμε να εκμεταλλευτούμε την τεχνική ενισχυτικού φιλτραρίσματος (**high boost filtering**), που χρησιμοποιεί μία από τις ακόλουθες μάσκες (Εικόνα 7) (Gonzalez & Woods, 2007).

0	-1	0	-1	-1	-1
-1	$A + 4$	-1	-1	$A + 8$	-1
0	-1	0	-1	-1	-1

Εικόνα 7: Μάσκες μεγέθους 3x3 για ενισχυτικό φιλτράρισμα

Με βάση την τεχνική αυτή, το άθροισμα των συντελεστών της μάσκας είναι ίσο με A , όπου το A είναι ένας αριθμός μεγαλύτερος ή ίσος με τη μονάδα. Στην περίπτωση που $A = 1$, τότε οι ομοιόμορφες περιοχές της εικόνας διατηρούν τη φωτεινότητά τους (ίδια περίπτωση με τις μάσκες Laplace ενός βήματος). Αν όμως $A > 1$, τότε η τελική εικόνα εμφανίζεται ταυτόχρονα οξυμένη και φωτεινότερη.

Για τα φίλτρα 1ης παραγώγου, οι πιο συχνά χρησιμοποιούμενες μάσκες έχουν διαστάσεις 3x3 και λέγονται μάσκες Sobel. Οι διαφορές στις φωτεινότητες των pixel μεταξύ της τρίτης και της πρώτης γραμμής της πρώτης μάσκας, προσεγγίζουν την παράγωγο στην κατεύθυνση x , ενώ οι διαφορές στις φωτεινότητες των pixel μεταξύ της τρίτης και της πρώτης στήλης της δεύτερης μάσκας, προσεγγίζουν την παράγωγο στην κατεύθυνση y . Επίσης, ο κεντρικός συντελεστής των μασκών έχει τιμή ίση με 2, προκειμένου να επιτύχουμε κάποιο βαθμό εξομάλυνσης, δίνοντας περισσότερη έμφαση στα κεντρικά pixel (Gonzalez & Woods, 2007). Οι μάσκες Sobel 3x3 φαίνονται παρακάτω (Εικόνα 8).

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Εικόνα 8: Μάσκες φίλτρου Sobel 3x3

Επίσης, έχουν προταθεί και οι εκτεταμένες μάσκες Sobel 5x5 οι οποίες επιτυγχάνουν καλύτερα αποτελέσματα (Kekre & Gharge, 2010). Οι μάσκες αυτές φαίνονται παρακάτω (Εικόνα 9).

+2	+2	+4	+2	+2	+2	+1	0	-1	-2
+1	+1	+2	+1	+1	+2	+1	0	-1	-2
0	0	0	0	0	+4	+2	0	-2	-4
-1	-1	-2	-1	-1	+2	+1	0	-1	-2
-2	-2	-4	-2	-2	+2	+1	0	-1	-2
Gx					Gy				

Εικόνα 9: Εκτεταμένες μάσκες φίλτρου Sobel μεγέθους 5x5

Σε κάθε περίπτωση, οι παράγωγοι 1^{ης} τάξης υλοποιούνται χρησιμοποιώντας το μέτρο της κλίσης σε κάθε pixel της εικόνας. Η φιλτραρισμένη εικόνα καλείται εικόνα κλίση ή απλούστερα κλίση. Έτσι, λοιπόν, σε κάθε pixel της αρχικής εικόνας $f(x,y)$, γίνεται εφαρμογή των 2 масκών Sobel, οι οποίες υπολογίζουν την κλίση στην οριζόντια g_x και στην κατακόρυφη κατεύθυνση g_y αντίστοιχα. Το ζητούμενο μέτρο της κλίσης $M(x,y)$ μπορεί να υπολογιστεί ως: $M(x,y) = \sqrt{g_x^2 + g_y^2} \cong |g_x| + |g_y|$ (Gonzalez & Woods, 2007).

Συνοψίζοντας, θα πρέπει να επισημάνουμε ότι οι μάσκες 1^{ης} παραγώγου δημιουργούν πιο χοντρές ακμές στην εικόνα και έχουν πιο ισχυρή απόκριση στις βηματικές αλλαγές φωτεινότητας, ενώ οι μάσκες 2^{ης} παραγώγου έχουν καλύτερη απόκριση σε λεπτές γραμμές και σε απομονωμένα σημεία, ενώ έχουν διπλή απόκριση στις βηματικές αλλαγές φωτεινότητας, δημιουργώντας διπλές ακμές. Συνεπώς, οι μάσκες 1^{ης} παραγώγου λειτουργούν ικανοποιητικά όταν η μετάβαση τιμών της εικόνας είναι απότομη, ενώ στην αντίθετη περίπτωση προτιμάται η χρήση των масκών 2^{ης} παραγώγου. Ωστόσο, επειδή οι μάσκες 2^{ης} παραγώγου (Laplace) είναι πολύ ευαίσθητες στον θόρυβο και παράγουν διπλές ακμές, στην πράξη υλοποιείται η Λαπλασιανή της εικόνας η οποία έχει συνελιχθεί με μία γκαουσιανή συνάρτηση (Laplacian of gaussian - LoG). Έτσι ανιχνεύονται τα σημεία μηδενισμού της 2^{ης} παραγώγου, τα οποία δίνουν και τις ακριβείς θέσεις των ακμών, χωρίς το μειονέκτημα της ευαισθησίας στο θόρυβο (CS425 Lab: Intensity Transformations and Spatial Filtering, 2011).

1.7 Εκπαιδευτικά Λογισμικά Ψηφιακής Επεξεργασίας Εικόνας

Δεδομένου ότι η ψηφιακή επεξεργασία εικόνας βρίσκει εφαρμογές στα αντικείμενα πολλών σύγχρονων επιστημονικών κλάδων, λχ ιατρική, πληροφορική, αστρονομία κοκ, είναι απαραίτητο οι επιστήμονες αυτών των κλάδων να εκπαιδευτούν στο αντικείμενο της ψηφιακής επεξεργασίας εικόνας και να αποκτήσουν καλή θεωρητική και πρακτική γνώση των τεχνικών της. Έτσι, λοιπόν, υπάρχει πληθώρα εργαλείων και εφαρμογών που έχουν αναπτυχθεί και στοχεύουν στην εκπαίδευση μέσω υπολογιστή της ψηφιακής επεξεργασίας εικόνας και των τεχνικών που την αφορούν, όπως τεχνικές βελτίωσης της ποιότητας της εικόνας, τεχνικές ανάλυσης κοκ. Τα δημοφιλέστερα προγραμματιστικά περιβάλλοντα που έχουν χρησιμοποιηθεί για την ανάπτυξη τέτοιων εφαρμογών είναι το Matlab, το Mathematica, η Visual Basic .Net, η Java και το LabView (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).

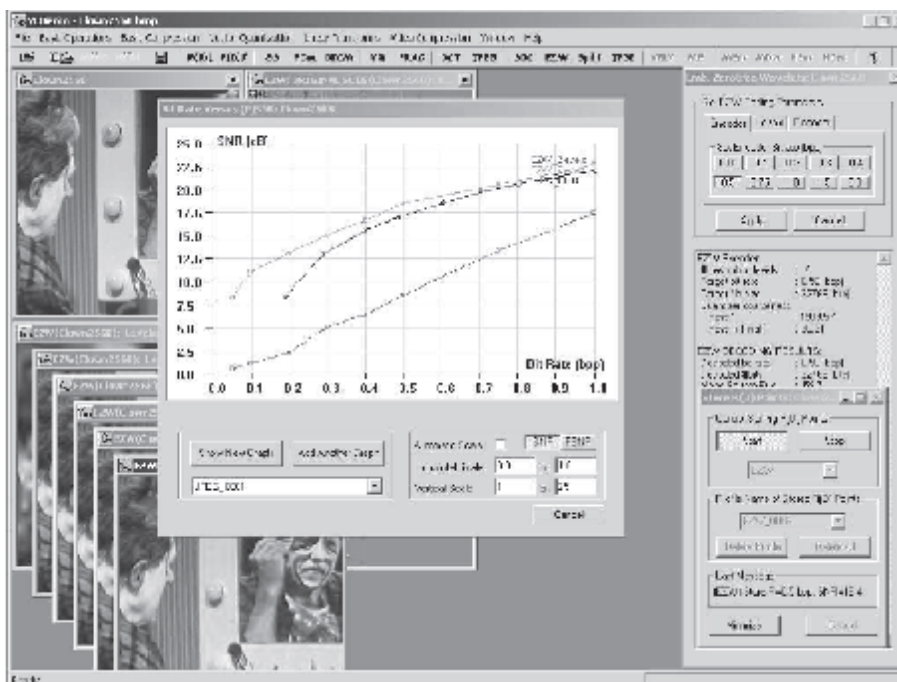
Επισημαίνεται ότι όλες οι εφαρμογές αυτές αν και είναι αλληλεπιδραστικές, ωστόσο, βασίζονται στην οπτική παρατήρηση και σύγκριση μεταξύ της αρχικής και της τελικής επεξεργασμένης εικόνας, που θα πρέπει να κάνει ο εκπαιδευόμενος προκειμένου να εξάγει τα συμπεράσματά του. Επιπλέον, οι περισσότερες από αυτές τις εφαρμογές αποκρύπτουν σχεδόν όλες τις απαιτούμενες υπολογιστικές διαδικασίες (δηλαδή την εκτέλεση των μαθηματικών πράξεων) της ζητούμενης κάθε φορά επεξεργασίας. Αυτό όμως έχει ως αποτέλεσμα συχνά να δημιουργείται γνωστικό κενό στους εκπαιδευόμενους, ειδικά επιστημών όπως της Πληροφορικής, γεγονός που λειτουργεί αρνητικά για την καλύτερη κατανόηση κάποιων εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας.

Παρακάτω παρουσιάζονται συνοπτικά μερικές εκπαιδευτικές εφαρμογές, που πραγματεύονται τεχνικές της ψηφιακής επεξεργασίας εικόνας και προωθούν τη μάθηση κυρίως μέσω παρατήρησης και πειραματισμού.

1.7.1 VcDemo

Το VcDemo είναι ένα εργαλείο που προορίζεται να βοηθήσει τους φοιτητές σε προπτυχιακό και μεταπτυχιακό επίπεδο, καθώς και επαγγελματίες με επαρκή κατάρτιση σε στοχαστικές διεργασίες και στην ψηφιακή επεξεργασία σήματος και εικόνας για τη μελέτη απωλεστικών τεχνικών συμπίεσης. Ενώ στα έντυπα εγχειρίδια δίνεται έμφαση στην θεωρητική εξήγηση των μεθόδων συμπίεσης και της δομής των σχετικών αλγορίθμων, το VcDemo δίνει έμφαση στη μελέτη των τεχνικών συμπίεσης από την πλευρά της παρατηρούμενης απόδοσης (λχ σήμα-προς-θόρυβο κοκ) υπό την επήρεια διαφορετικών ρυθμίσεων στις παραμέτρους των τεχνικών (λχ ο ρυθμός σφαλμάτων του καναλιού μετάδοσης των συμπιεσμένων δεδομένων κοκ) (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).

Το VcDemo περιλαμβάνει σχεδόν είκοσι ενότητες για τη συμπίεση εικόνας και βίντεο. Διαθέτει γραφικό περιβάλλον με μενού και πλήκτρα, ενώ δεν απαιτεί από τον χρήστη να γνωρίζει προγραμματισμό. Παρέχει επεξηγήσεις και ενδείκνυται για αυτοδιδασκαλία, για εξάσκηση των εκπαιδευομένων στο σπίτι και για παρουσιάσεις εντός της τάξης.

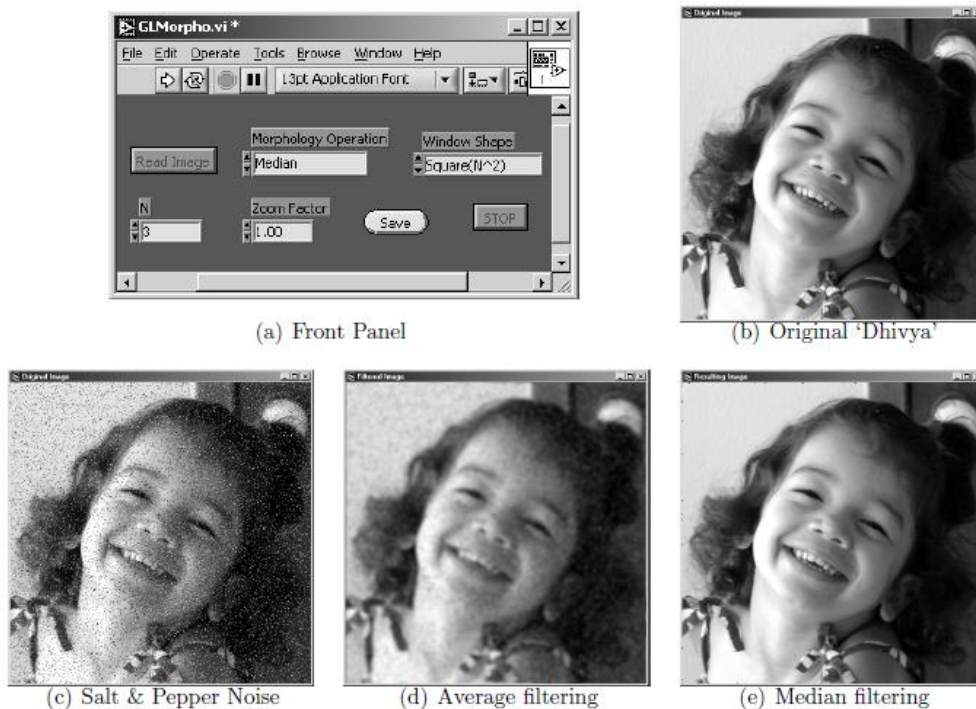


Εικόνα 10: Το περιβάλλον του VcDemo (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004)

1.7.2 SIVA

Πρόκειται για μία συλλογή εργαλείων επεξεργασίας εικόνας, που αναπτύχθηκε από το Πανεπιστήμιο Όστιν του Τέξας. Η συλλογή εργαλείων επεξεργασίας εικόνας SIVA (The Signal, Image and Video-Audio visualization gallery) αποτελείται από μια σουίτα προγραμμάτων ειδικού σκοπού, που βασίζονται στο LabView. Τα προγράμματα αυτά έχουν σχεδιαστεί ώστε να αποτελέσουν ισχυρά εργαλεία απεικόνισης-οπτικοποίησης των βασικών εννοιών και τεχνικών που χρησιμοποιούνται στον χώρο της ψηφιακής επεξεργασίας εικόνας και βίντεο (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).

Η συλλογή SIVA διαθέτει γραφικό περιβάλλον και είναι ιδανική για διδασκαλία εξ' αποστάσεως και δια ζώσης. Η συλλογή εφαρμογών SIVA είναι χρήσιμη για τη διδασκαλία της ψηφιακής επεξεργασίας εικόνας σε προπτυχιακό επίπεδο, ως εργαλείο επίδειξης στην τάξη για να παρέχει παραδείγματα, τα οποία απεικονίζουν τον αλγόριθμο ή την τεχνική επεξεργασίας εικόνας και βίντεο που διδάσκεται κάθε φορά.

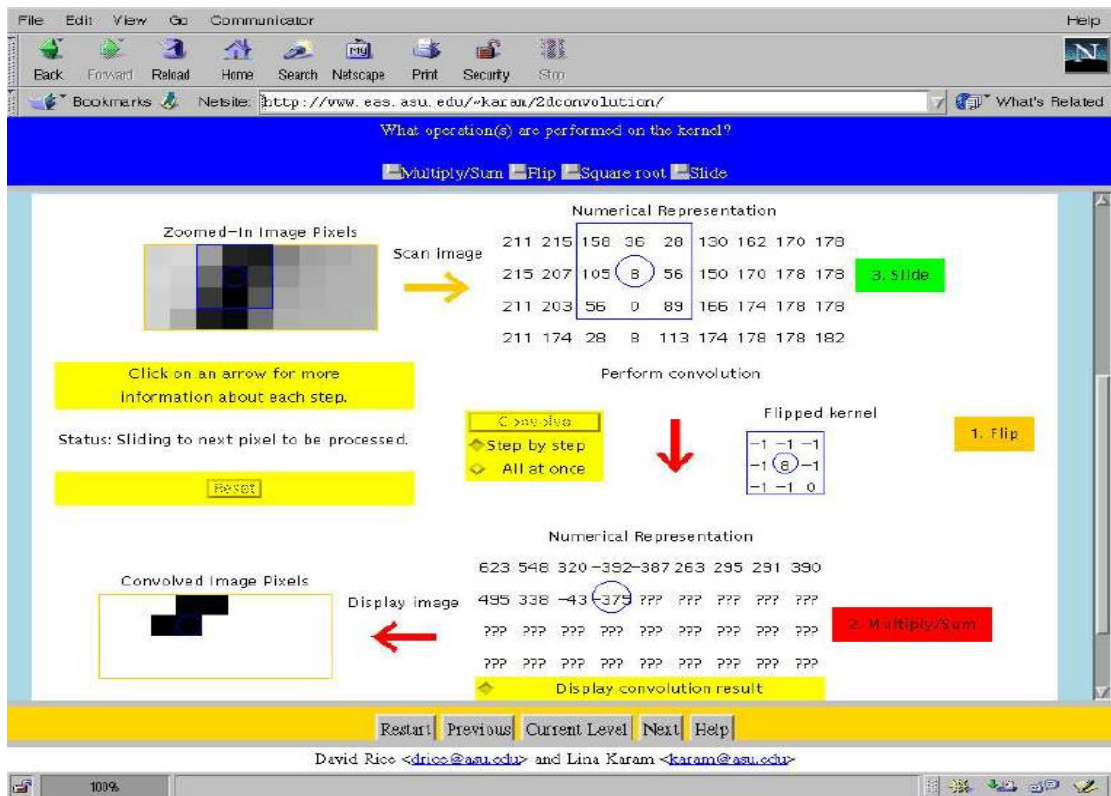


Εικόνα 11: Το εργαλείο χωρικού φιλτραρίσματος της συλλογής εφαρμογών SIVA (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004)

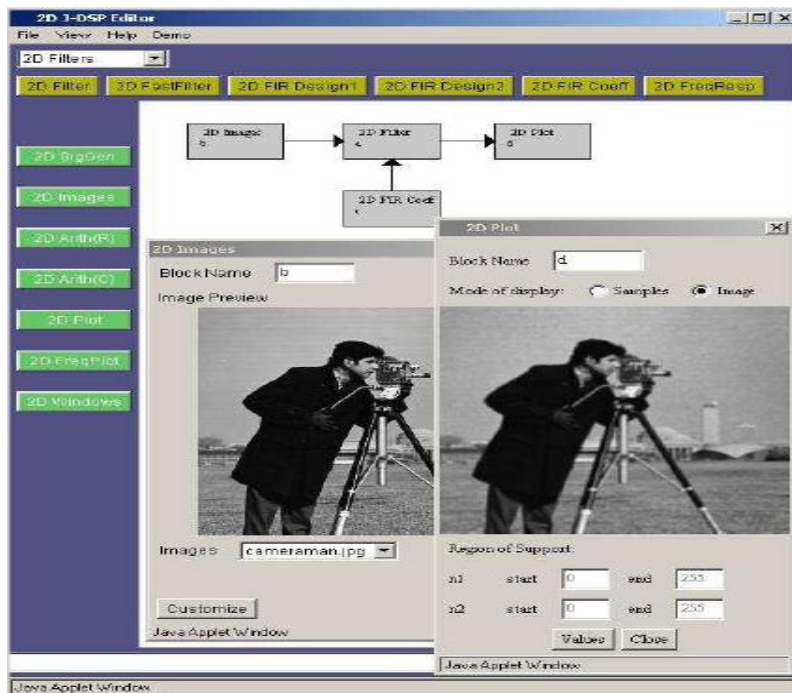
1.7.3 ALMOT 2D DSP και 2D J-DSP

Τα εργαλεία ALMOT 2D DSP και 2D J-DSP αναπτύχθηκαν και χρησιμοποιούνται από το πανεπιστήμιο Αριζόνα των ΗΠΑ. Πρόκειται για διαδραστικά, ανεξάρτητα της πλατφόρμας, online εκπαιδευτικά λογισμικά, που αφορούν τη διδασκαλία της θεωρίας για τα δυσδιάστατα σήματα και συστήματα, καθώς και τις εφαρμογές τους στην επεξεργασία ψηφιακών εικόνων. Και τα δύο εργαλεία βασίζονται στην τεχνολογία Java.

Το ALMOT 2D DSP εστιάζει στην ενεργητική μάθηση μέσω τμηματικών δοκιμών (Active Learning through MODular Testing) και απευθύνεται σε χρήστες χωρίς καμία γνώση της θεματικής των σημάτων και συστημάτων. Το 2D J-DSP απευθύνεται σε προπτυχιακούς και μεταπτυχιακούς φοιτητές, που θέλουν να αποκτήσουν βαθύτερη κατανόηση της θεωρίας της δυσδιάστατης επεξεργασίας σήματος, παρέχοντας online εργαστήρια και πειράματα (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).



Εικόνα 12: Το περιβάλλον του ALMOT 2D DSP (Rajashekar, Bovik, Sage, Unser, Karam, & Legendijk, 2004)

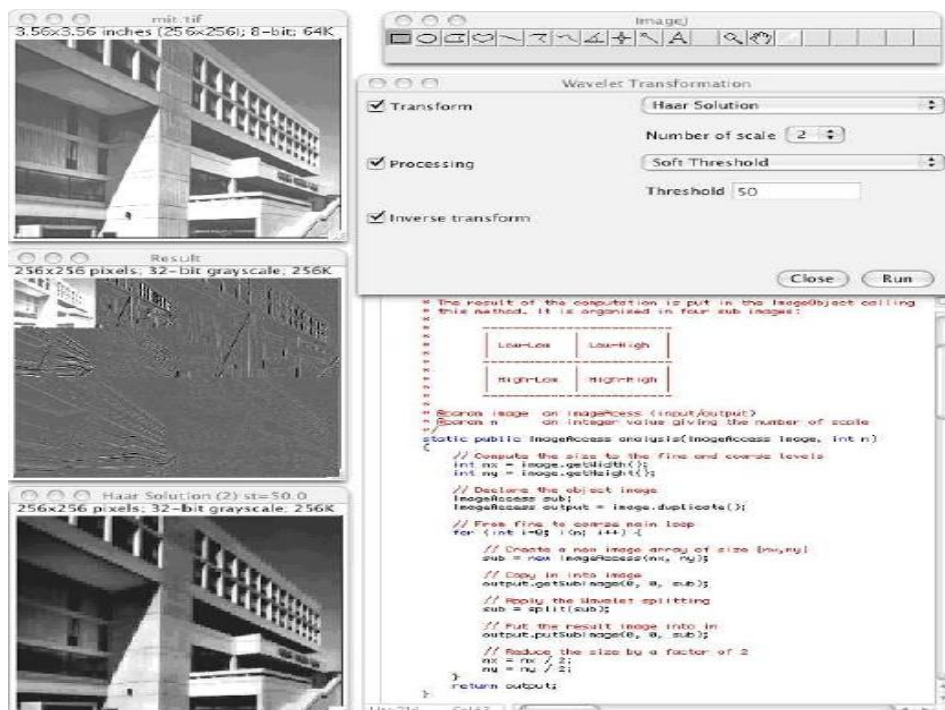


Εικόνα 13: Το περιβάλλον του 2D J-DSP (Rajashekar, Bovik, Sage, Unser, Karam, & Legendijk, 2004)

1.7.4 IP-Lab

Το IP-Lab, είναι ένα εκπαιδευτικό εργαλείο για τη διδασκαλία των πρακτικών και των υπολογιστικών πτυχών της ψηφιακής επεξεργασίας εικόνας. Πρόκειται για μια συλλογή εργαστηριακών μαθημάτων με μορφή ασκήσεων για την υποστήριξη των βασικών θεωρητικών διαλέξεων της ψηφιακής επεξεργασίας εικόνας.

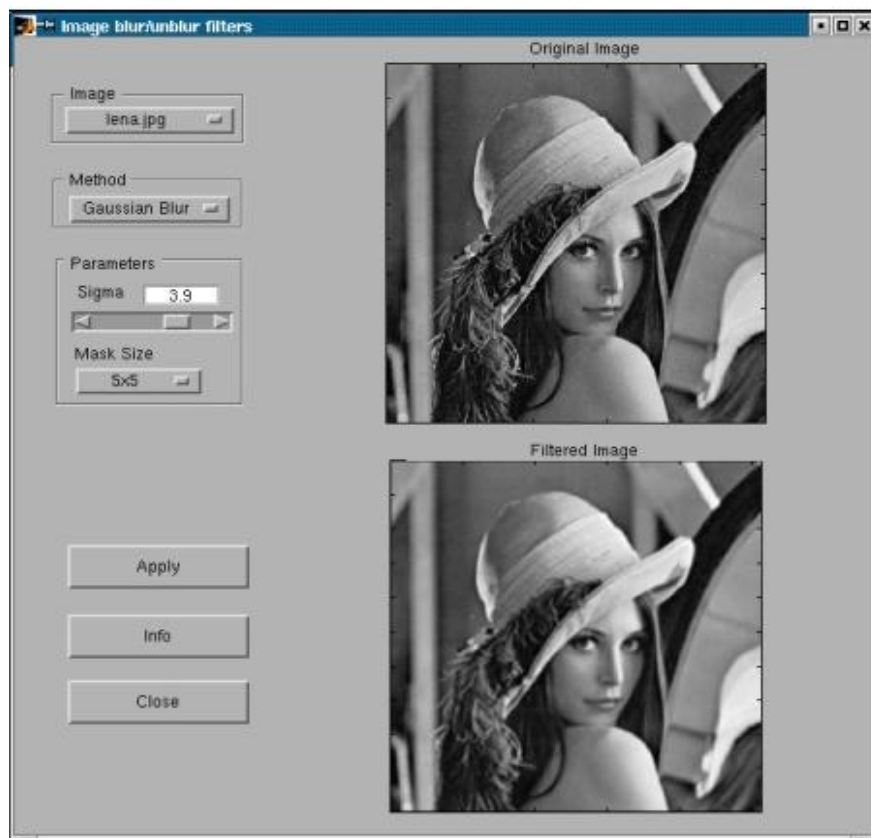
Το IP-LAB παρέχει ένα φιλικό στον χρήστη γραφικό περιβάλλον, στο οποίο οι εκπαιδευόμενοι μπορούν να πειραματιστούν με αλγορίθμους επεξεργασίας εικόνας, να αλλάξουν τις παραμέτρους και να λάβουν άμεση οπτική ανατροφοδότηση και να γράψουν το δικό τους κώδικα σε επίπεδο pixel. Το IP-LAB βασίζεται στο λογισμικό επεξεργασίας εικόνας γενικής χρήσης ImageJ, που είναι γραμμένο σε Java και διατίθεται ελεύθερα. Βασίζεται στη στρατηγική μάθησης μέσω παραδειγμάτων και απαιτεί στοιχειώδεις γνώσεις προγραμματισμού σε Java για την εφαρμογή των τυποποιημένων αλγορίθμων επεξεργασίας εικόνας (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004).



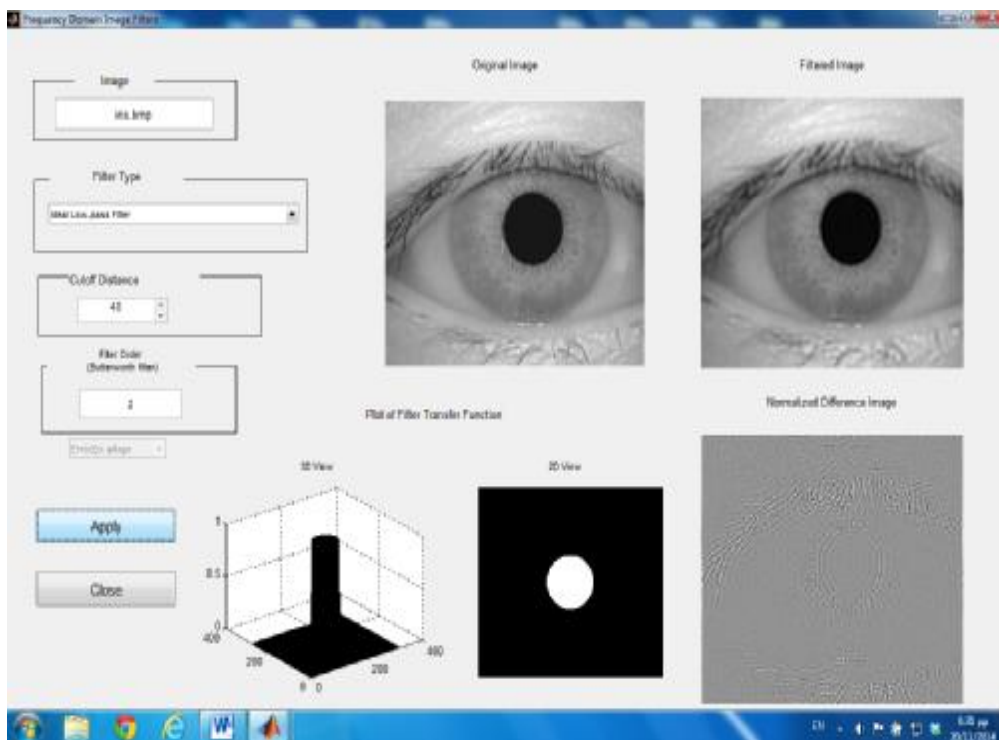
Εικόνα 14: Το περιβάλλον του IP-Lab (Rajashekar, Bovik, Sage, Unser, Karam, & Lagendijk, 2004)

1.7.5 Εφαρμογές Matlab

Το Matlab είναι το δημοφιλέστερο περιβάλλον δημιουργίας εκπαιδευτικών εφαρμογών στον χώρο της ψηφιακής επεξεργασίας εικόνας. Το ίδιο το Matlab συνοδεύεται από πολλά παραδείγματα (demos) και εφαρμογές τεχνικών και αλγορίθμων για όλα σχεδόν τα είδη ψηφιακής επεξεργασίας και ανάλυσης μίας εικόνας. Όλες οι εφαρμογές διαθέτουν απλή και εύχρηστη γραφική διεπαφή, δεν απαιτούν γνώσεις προγραμματισμού από τους εκπαιδευόμενους, ενώ διαθέτουν χειριστήρια μέσω των οποίων οι χρήστες μπορούν να τροποποιήσουν διάφορες παραμέτρους και να έχουν άμεση οπτική ανατροφοδότηση. Παρακάτω φαίνονται παραδείγματα τέτοιων εφαρμογών σε Matlab.



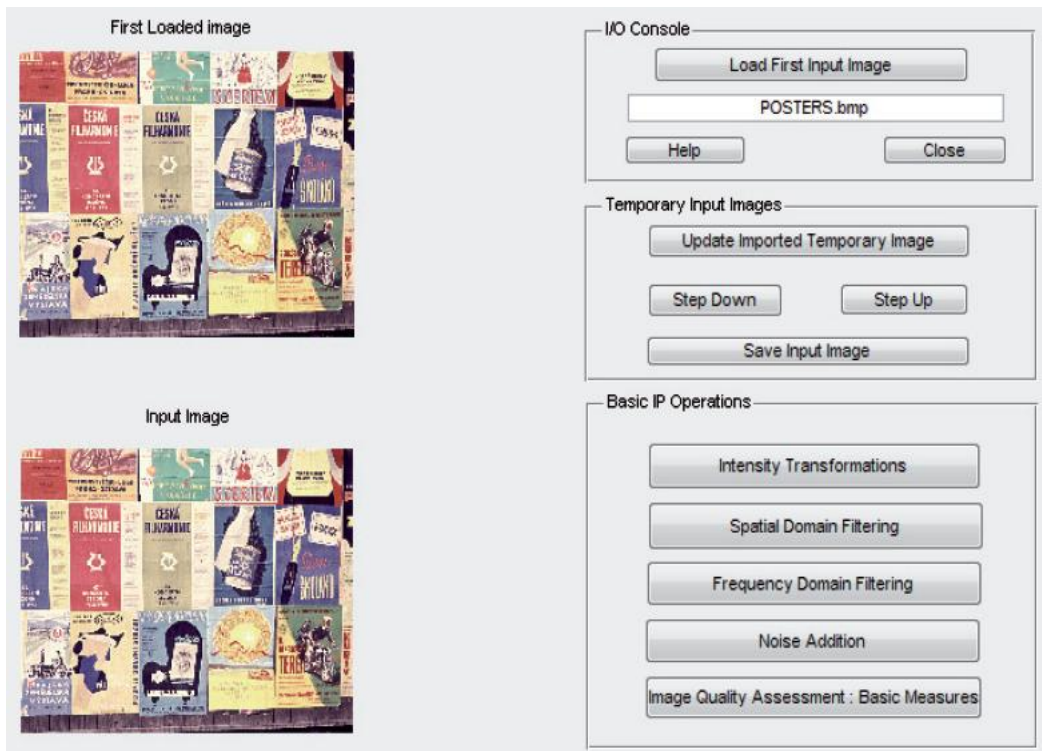
Εικόνα 15: Εφαρμογή Matlab για επίδειξη τεχνικών εξομάλυνσης/όξυνσης εικόνας (MathWorks.com, 2012)



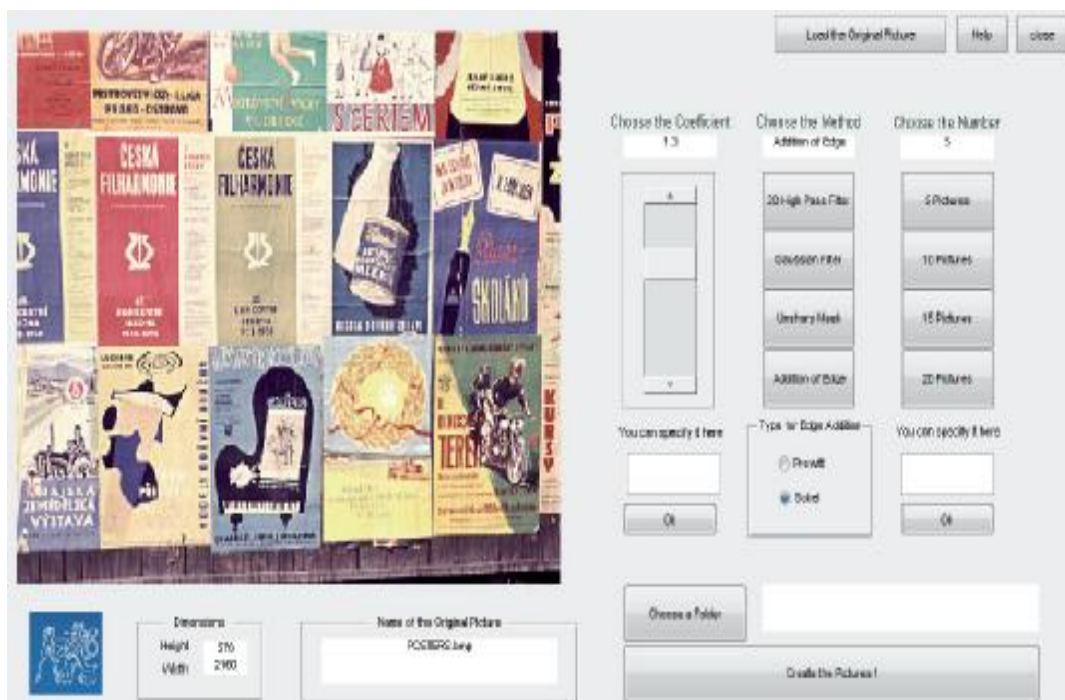
Εικόνα 16: Εφαρμογή Matlab για επίδειξη φιλτραρίσματος στο πεδίο συχνότητας (MathWorks.com, 2012)

Επίσης, αξίζει να αναφερθεί και η αξιολογή συλλογή εφαρμογών σε Matlab των Krasula K., και συνεργατών (2011) και (2012). Οι εφαρμογές αυτές πραγματεύονται πληθώρα εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνων, ενώ επιπλέον πραγματεύονται και την αξιολόγηση της ποιότητας των ψηφιακών εικόνων. Στα πλεονεκτήματά τους συγκαταλέγεται το πλούσιο και εύχρηστο γραφικό περιβάλλον που διαθέτουν, το οποίο επιτρέπει στον εκπαιδευόμενο να τροποποιήσει μία πληθώρα παραμέτρων κάθε τεχνικής της ψηφιακής επεξεργασίας εικόνας.

Στις παρακάτω εικόνες (Εικόνα 17 και Εικόνα 18) φαίνονται ενδεικτικά δύο (2) από τις εφαρμογές αυτές.



Εικόνα 17: Εφαρμογή Matlab για την επεξεργασία εικόνας (Krasula, K., et al, 2011)



Εικόνα 18: Εφαρμογή Matlab για την επεξεργασία εικόνας (Krasula, K., et al, 2011)

2 MATLAB

Στην ενότητα αυτή γίνεται αναφορά στο εργαλείο Matlab και παρουσιάζονται συνοπτικά το περιβάλλον δημιουργίας γραφικών διεπαφών GUIDE και η εργαλειοθήκη επεξεργασίας εικόνας (Image Processing Toolbox) του Matlab. Τα εργαλεία αυτά χρησιμοποιήθηκαν για τη σχεδίαση και την ανάπτυξη της εφαρμογής, η οποία παρουσιάζεται στην ενότητα 3.

2.1 Βασικά Χαρακτηριστικά

Το MATLAB είναι μία γλώσσα υψηλού επιπέδου, που χρησιμοποιείται για τεχνικούς υπολογισμούς. Το όνομα Matlab προέρχεται από τα αρχικά των λέξεων Matrix Laboratory και η λειτουργία του βασίζεται κυρίως στην χρήση πινάκων, τα στοιχεία των οποίων μπορούν να είναι πραγματικοί ή μιγαδικοί αριθμοί. Σχεδιάστηκε με αρχικό σκοπό την υπολογιστική επίλυση απλών και σύνθετων μαθηματικών προβλημάτων, ωστόσο σήμερα χρησιμοποιείται σχεδόν σε όλους τους επιστημονικούς κλάδους. Τα προγράμματα που είναι γραμμένα σε Matlab είναι γενικά σαφή, ευανάγνωστα και εύκολα στην κατανόηση. Η γλώσσα είναι αυστηρά δομημένη στη σύνταξη των εντολών και διαθέτει όλα τα χαρακτηριστικά μίας σύγχρονης γλώσσας προγραμματισμού, η οποία υποστηρίζεται από ένα εξελιγμένο προγραμματιστικό περιβάλλον με πληθώρα βοηθητικών εργαλείων. Τα προγράμματα σε Matlab είναι ανεξάρτητα της πλατφόρμας στην οποία εκτελούνται, ενώ ένα από τα βασικά χαρακτηριστικά τους είναι το γεγονός ότι ο προγραμματιστής δεν απαιτείται να ορίζει τον τύπο των μεταβλητών. Επιπλέον, η δυνατότητα του Matlab να κάνει συμβολικές πράξεις και ο μεγάλος αριθμός έτοιμων βιβλιοθηκών που διαθέτει, το κατατάσσει στα καλύτερα λογισμικά της κατηγορίας του (Marchand & Holland, 2003).

2.2 Το Περιβάλλον GUIDE

Το MATLAB διαθέτει ένα ενσωματωμένο περιβάλλον – εργαλειοθήκη για τη σχεδίαση φιλικών στον χρήστη γραφικών διεπαφών. Το περιβάλλον αυτό, που καλείται και ως GUIDE (Graphical User Interface Design Environment), διαθέτει μία

πληθώρα έτοιμων στοιχείων ελέγχου, όπως πλήκτρα, πλαίσια, μενού, άξονες, πίνακες κοκ.

Μία γραφική διεπαφή αποτελείται από διάφορα παράθυρα, που μπορούν να διαθέτουν ποικίλα στοιχεία ελέγχου, όπως πεδία κειμένου, πλήκτρα, μπάρες κύλισης κοκ. Τα παράθυρα μπορούν να καλούν το ένα το άλλο, να δέχονται είσοδο από τον χρήστη, να επικοινωνούν μεταβιβάζοντας δεδομένα μεταξύ τους και γενικότερα να επιτελούν διάφορες λειτουργίες (Lent, 2013). Προκειμένου το GUIDE να το πετύχει αυτό, για κάθε ένα παράθυρο δημιουργεί δύο αρχεία, ένα Fig - file και ένα M - file. Πιο συγκεκριμένα:

- Το Fig – file είναι το παράθυρο (figure), όπου το MATLAB αποθηκεύει τα στοιχεία ελέγχου και την ακριβή τους θέση. Ουσιαστικά, στο αρχείο αυτό ο προγραμματιστής σχεδιάζει την εμφάνιση του παραθύρου.
- Το M – file είναι το αρχείο στο οποίο ο προγραμματιστής συγγράφει τον κώδικα που αφορά τη συμπεριφορά των στοιχείων ελέγχου του παραθύρου, προκειμένου το παράθυρο να επιτελεί τις επιθυμητές λειτουργίες.

Επισημαίνεται ότι κάθε fig - file πρέπει να συνοδεύεται από ένα ομώνυμο αρχείο m – file. Αν χαθεί ή καταστραφεί το M – file, τότε το παράθυρο δε θα είναι λειτουργικό. Όποτε ο χρήστης δημιουργεί ένα νέο παράθυρο, το GUIDE δημιουργεί αυτομάτως και τους δύο προαναφερθέντες τύπους αρχείων (Βαρσάμης, 2006).

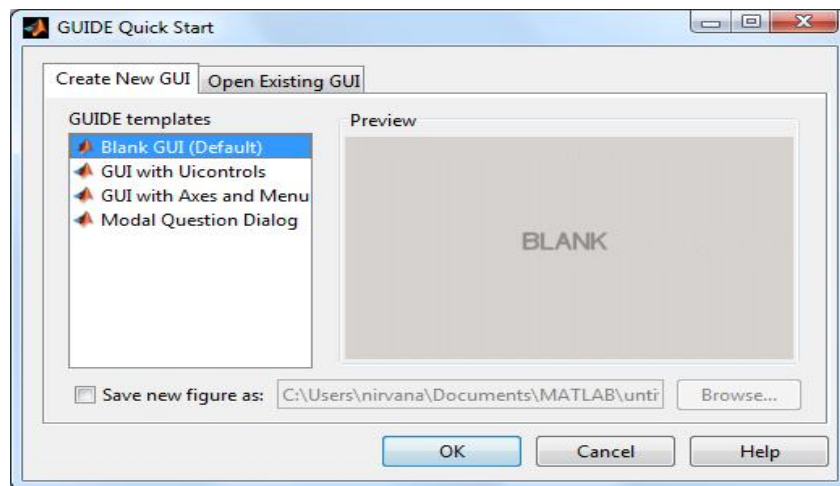
Η εκκίνηση του GUIDE γίνεται με δύο τρόπους:

1. Γράφοντας την εντολή «guide» στη γραμμή εντολών του Matlab
2. Από το μενού του Matlab «File» à «New» à «GUI»

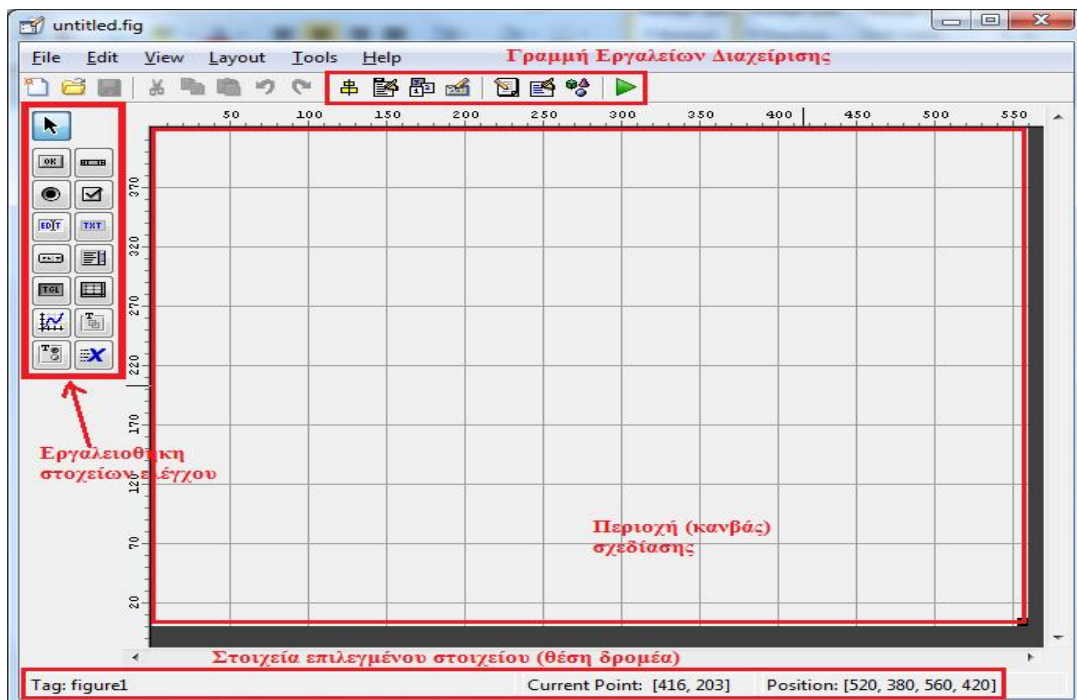
Αμέσως θα ξεκινήσει ένας οδηγός που θα καθοδηγήσει τον χρήστη στη δημιουργία ενός παραθύρου. Αρχικά ο χρήστης θα πρέπει να επιλέξει αν θέλει να δημιουργήσει ένα κενό παράθυρο (Blank GUI) ή ένα παράθυρο βασισμένο σε κάποια πρότυπα (πχ με άξονες και μενού) ή αν θέλει να ανοίξει ένα έτοιμο

παράθυρο. Μόλις ο χρήστης επιλέξει το είδος του παραθύρου που επιθυμεί, τότε θα εμφανιστεί το νέο παράθυρο στο περιβάλλον σχεδίασης του GUIDE.

Στις ακόλουθες εικόνες (Εικόνα 19 και Εικόνα 20) φαίνονται ο οδηγός για τη δημιουργία ενός παραθύρου του GUIDE και το περιβάλλον σχεδίασης του νέου παραθύρου στο GUIDE. Τα βασικά τμήματα του περιβάλλοντος σχεδίασης του GUIDE εμφανίζονται με κόκκινη σήμανση.

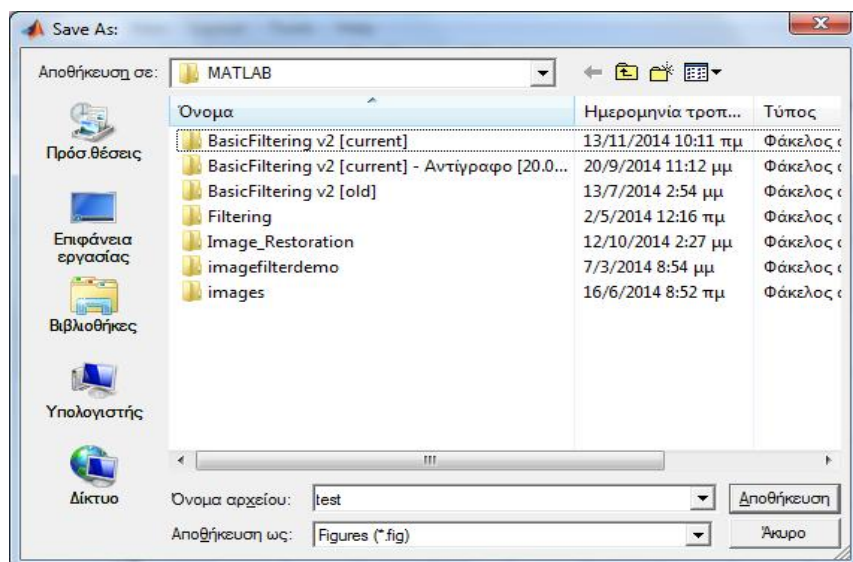


Εικόνα 19: Οδηγός δημιουργίας παραθύρου



Εικόνα 20: Περιβάλλον σχεδίασης νέου παραθύρου του GUIDE

Για την αποθήκευση του παραθύρου, ο χρήστης πρέπει να επιλέξει «File» à «Save as...», να επιλέξει τη θέση και το όνομα που θέλει να δώσει στο παράθυρο και να πατήσει το πλήκτρο «Αποθήκευση». Το Matlab θα δημιουργήσει το fig – file και το M – file και θα εμφανίσει σε ένα νέο παράθυρο τον κώδικα του M-file που αφορά το παράθυρο. Στην Εικόνα 21 φαίνεται η αποθήκευση του παραθύρου GUIDE.










Εικόνα 21: Αποθήκευση παραθύρου GUIDE









Στο παράθυρο σχεδίασης του GUIDE ο προγραμματιστής μπορεί να σχεδιάσει τη γραφική διεπαφή που θα εξυπηρετεί την εφαρμογή του. Για το σκοπό αυτόν, το GUIDE διαθέτει μία εργαλειοθήκη στοιχείων ελέγχου, όπου ο προγραμματιστής μπορεί να επιλέξει και να σύρει πάνω στον κανβά σχεδίασης το επιθυμητό στοιχείο.

Στον πίνακα που ακολουθεί (Πίνακας 1) περιγράφεται η λειτουργία καθενός στοιχείου ελέγχου της εργαλειοθήκης του GUIDE. Επισημαίνεται ότι ο προγραμματιστής μπορεί να γράψει κώδικα για οποιοδήποτε στοιχείο ελέγχου, προκειμένου να καθορίσει τη συμπεριφορά του κατά την εκτέλεση της εφαρμογής.

Πίνακας 1: Στοιχεία ελέγχου της εργαλειοθήκης του GUIDE

Εικονίδιο	Λεκτικό	Περιγραφή
	Select	Εργαλείο επιλογής και μετακίνησης αντικειμένων.

	Push Button	<p>Δημιουργεί ένα πλήκτρο, το οποίο εκτελεί μια διαδικασία (Callback) όταν πατηθεί από τον χρήστη κατά την εκτέλεση της εφαρμογής.</p>
	Slider	<p>Δημιουργεί μία μπάρα κύλισης, η οποία δέχεται μία αριθμητική είσοδο εντός ενός πεδίου τιμών που ορίζει ο προγραμματιστής.</p>
	Radio Button	<p>Δημιουργεί ένα πλήκτρο τύπου radio. Ο προγραμματιστής πρέπει να φροντίζει ώστε όταν κατά την εκτέλεση της εφαρμογής ο χρήστης επιλέξει κάποιο radio button, τότε να απενεργοποιούνται τα υπόλοιπα radio buttons. Χρησιμοποιείται όταν θέλουμε να δώσουμε τη δυνατότητα στον χρήστη να επιλέξει μόνο μία επιλογή από ένα πλήθος διαθέσιμων επιλογών.</p>
	Check Box	<p>Δημιουργεί ένα πλήκτρο τύπου check. Μοιάζει με ένα radio button, όμως όταν επιλέγεται (τσεκάρεται) από τον χρήστη κατά την εκτέλεση της εφαρμογής, δεν χρειάζεται να απενεργοποιούνται τα υπόλοιπα.</p>
	Edit Text	<p>Δημιουργεί ένα πλαίσιο μέσα στο οποίο ο χρήστης μπορεί να εισάγει δεδομένα μέσω του πληκτρολογίου κατά την εκτέλεση της εφαρμογής. Στη συνέχεια, τα δεδομένα αυτά μπορεί να τα αξιοποιήσει η εφαρμογή.</p>
	Static Text	<p>Επιτρέπει τη δημιουργία στατικών κειμένων, για την παροχή πληροφοριών προς τον χρήστη κατά την εκτέλεση της εφαρμογής.</p>

	Pop-Up Menu	Δημιουργεί μία αναπτυσσόμενη λίστα επιλογών.
	List Box	Δημιουργεί μία απλή λίστα επιλογών.
	Toggle Button	Δημιουργεί ένα πλήκτρο τύπου «toggle». Μοιάζει με ένα απλό πλήκτρο (push button) με τη διαφορά ότι μπορεί να έχει δύο τιμές, τις «On» και «Off».
	Table	Χρησιμοποιείται για την οργάνωση και την παρουσίαση δεδομένων σε μορφή πίνακα.
	Axes	Επιτρέπει την τοποθέτηση γραφικών στοιχείων, όπως εικόνων και γραφημάτων.
	Panel	Χρησιμοποιείται για την ομαδοποίηση πολλών εργαλείων.
	Button Group	Χρησιμοποιείται για την ομαδοποίηση πολλών radio buttons ή toggle buttons.
	ActiveX Control	Επιτρέπει την προσθήκη και εμφάνιση ActiveX χειριστηρίων. Χρησιμοποιείται μόνο στα Windows.








Επιπλέον, το περιβάλλον του GUIDE διαθέτει μία γραμμή εργαλείων διαχείρισης, η οποία παρέχει άμεση πρόσβαση σε χρήσιμα εργαλεία. Η γραμμή των εργαλείων διαχείρισης φαίνεται στην Εικόνα 22.




Εικόνα 22: Γραμμή εργαλείων διαχείρισης του GUIDE

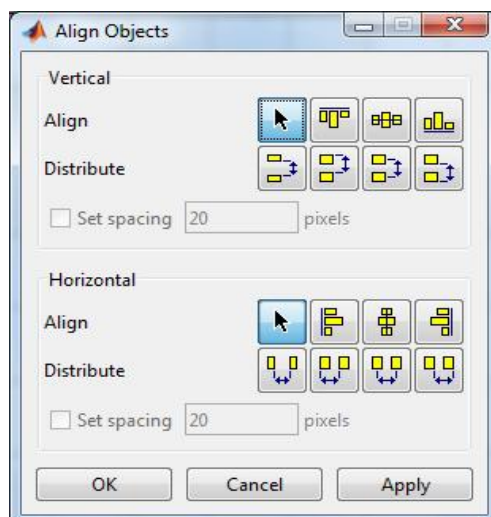
Στον πίνακα που ακολουθεί (Πίνακας 2) περιγράφεται η λειτουργία καθενός στοιχείου της γραμμής εργαλείων διαχείρισης του GUIDE.

Πίνακας 2: Στοιχεία γραμμής εργαλείων διαχείρισης του GUIDE

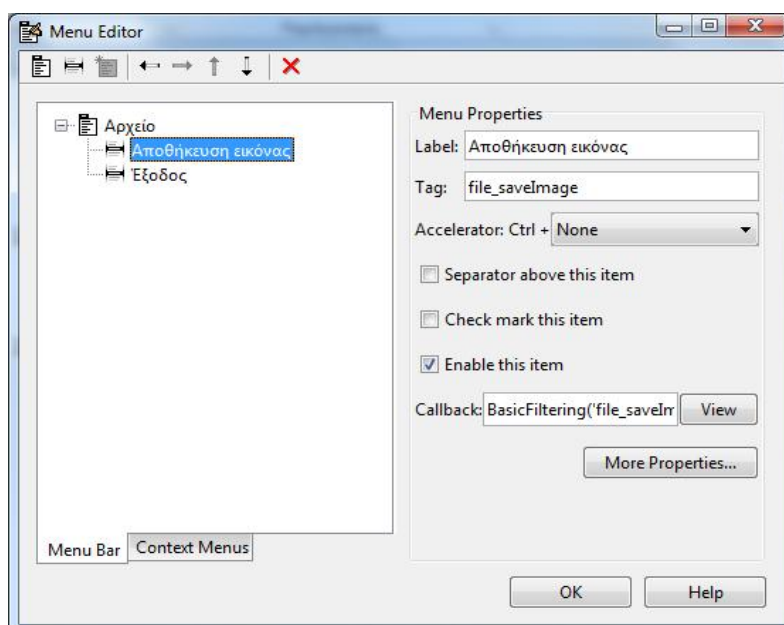
Εικονίδιο	Λεκτικό	Περιγραφή
	Align Objects	Εργαλείο για τη στοίχιση των αντικειμένων – στοιχείων ελέγχου στο παράθυρο της εφαρμογής.
	Menu Editor	Επιτρέπει τη δημιουργία και διαχείριση ενός βασικού μενού για το παράθυρο της εφαρμογής. Επιπλέον, επιτρέπει και τη δημιουργία αναπτυσσόμενων μενού, που ενεργοποιούνται όταν ο χρήστης κάνει δεξί κλικ κατά την εκτέλεση της εφαρμογής.
	Tab Order Editor	Χρησιμοποιείται για τον ορισμό της σειράς με την οποία τα στοιχεία ελέγχου του παραθύρου αποκτούν την εστίαση, όταν ο χρήστης πατάει το πλήκτρο «Tab» κατά την εκτέλεση της εφαρμογής.
	Toolbar Editor	Επιτρέπει τη δημιουργία γραμμής εργαλείων στο παράθυρο της εφαρμογής.
	Editor	Ανοίγει το παράθυρο επεξεργασίας του κώδικα του παραθύρου, ο οποίος βρίσκεται στο ομώνυμο M – file.
	Property Inspector	Πολύ χρήσιμο εργαλείο για τον προγραμματιστή, αφού παρέχει άμεση πρόσβαση στις ιδιότητες του επιλεγμένου στοιχείου ελέγχου. Οι ιδιότητες αυτές εμφανίζονται σε ένα παράθυρο.
	Object Browser	Επίσης πολύ χρήσιμο εργαλείο για τον προγραμματιστή, αφού εμφανίζει την ιεραρχία των στοιχείων ελέγχου που έχει τοποθετήσει στο παράθυρο της εφαρμογής.

	Run Figure	Εκτελεί το παράθυρο της εφαρμογής.
---	------------	------------------------------------

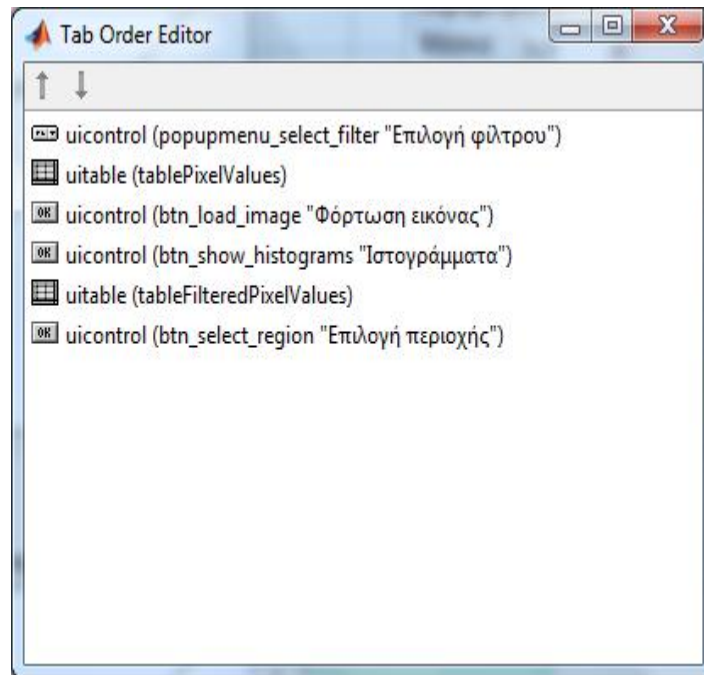
Επίσης, στις παρακάτω εικόνες (Εικόνα 23 ως Εικόνα 29) παρουσιάζονται τα παράθυρα που αντιστοιχούν στα στοιχεία της γραμμής εργαλείων διαχείρισης του GUIDE.



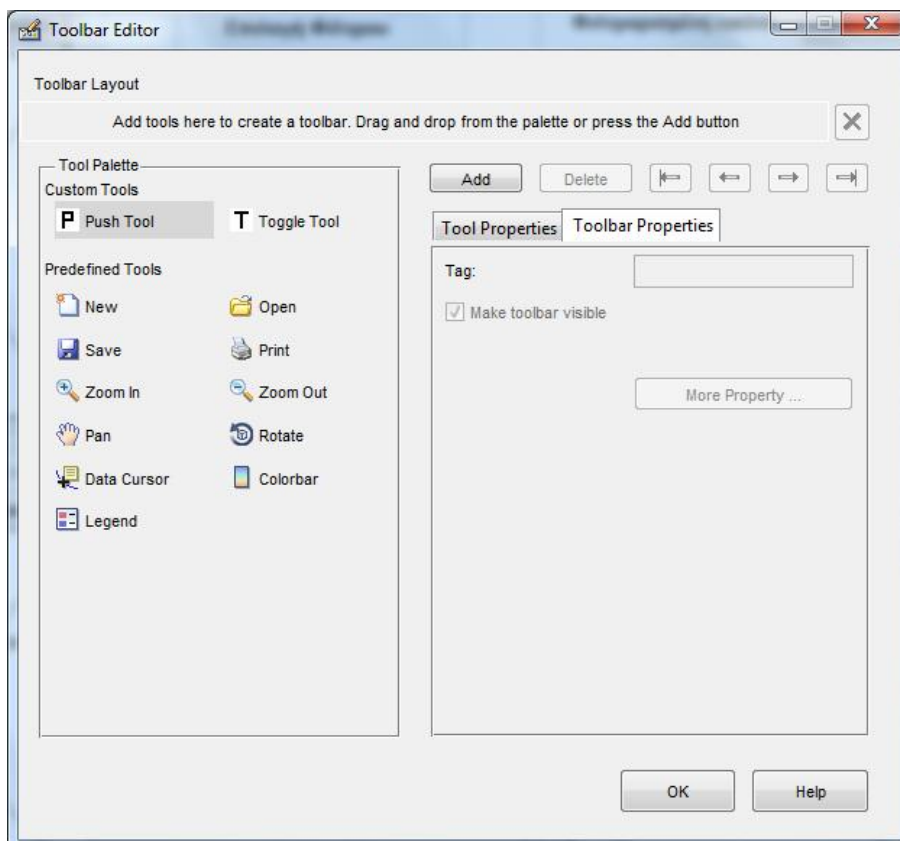
Εικόνα 23: Παράθυρο Align Objects του GUIDE



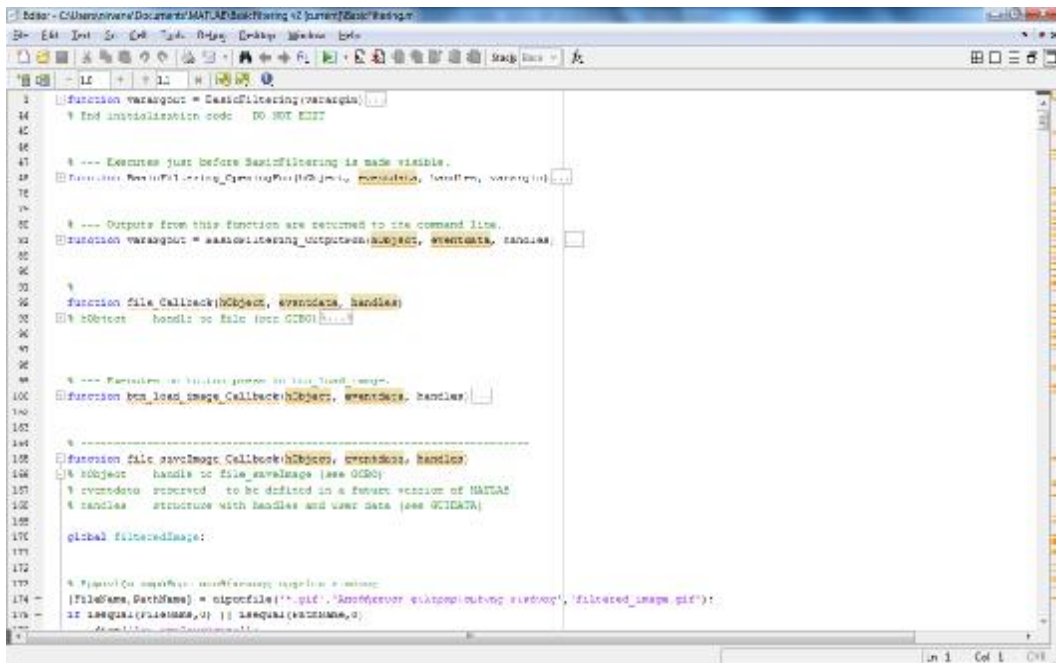
Εικόνα 24: Παράθυρο Menu Editor του GUIDE



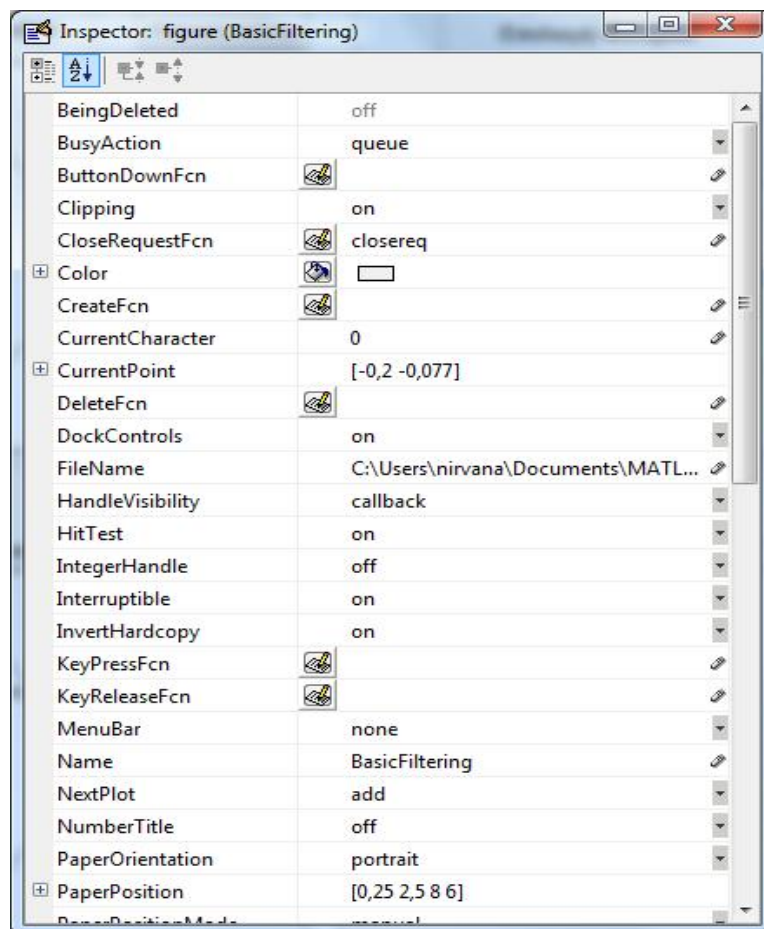
Εικόνα 25: Παράθυρο Tab Order Editor του GUIDE



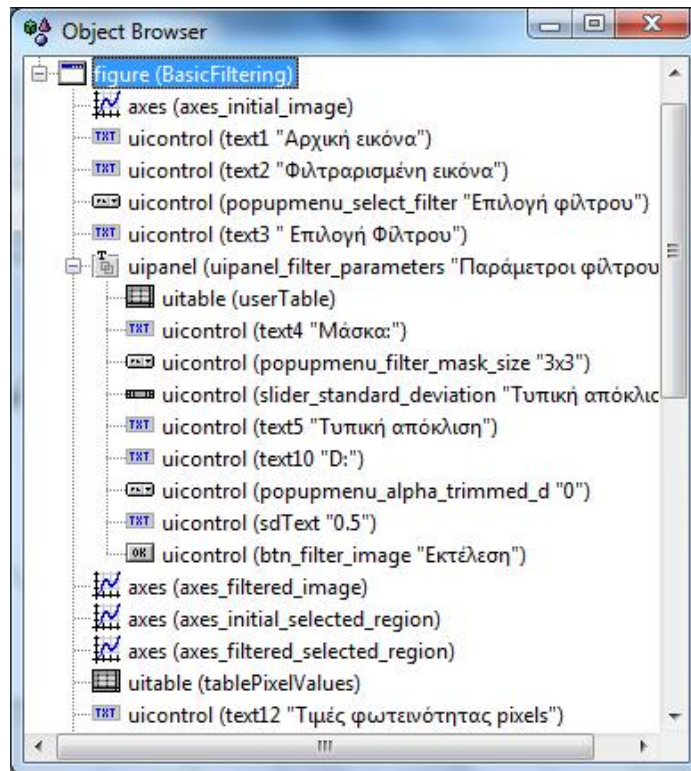
Εικόνα 26: Παράθυρο Toolbar Editor του GUIDE



Εικόνα 27: Παράθυρο Editor του GUIDE



Εικόνα 28: Παράθυρο Property Inspector του GUIDE



Εικόνα 29: Παράθυρο Object Browser του GUIDE

2.3 Εργαλειοθήκη Επεξεργασίας Εικόνας (IP Toolbox)

Η Εργαλειοθήκη Επεξεργασίας Εικόνας (Image Processing Toolbox) του Matlab παρέχει μια ολοκληρωμένη σειρά τυπικών αλγορίθμων, λειτουργιών - συναρτήσεων και εφαρμογών για την επεξεργασία, ανάλυση και απεικόνιση εικόνας, καθώς και για την ανάπτυξη σχετικών αλγορίθμων. Παρέχει τη δυνατότητα στον προγραμματιστή να εκτελέσει ανάλυση εικόνας, κατάτμηση εικόνας, βελτίωση ποιότητας εικόνας, μείωση θορύβου, γεωμετρικούς μετασχηματισμούς και καταγραφή εικόνας. Πολλές από τις λειτουργίες της εργαλειοθήκης υποστηρίζουν πολυπύρηνους επεξεργαστές, κάρτες γραφικών και την παραγωγή κώδικα σε γλώσσα C (MathWorks.com, 2012).

Επίσης, η εργαλειοθήκη επεξεργασίας εικόνας υποστηρίζει ένα ευρύ σύνολο τύπων εικόνας, όπως αυτών με υψηλό δυναμικό εύρος, με ανάλυση $gigapixel$, εικόνες τομογράφων κ.ο.κ. Οι λειτουργίες απεικόνισης και οι εφαρμογές της εργαλειοθήκης, επιτρέπουν στον προγραμματιστή να εξερευνήσει εικόνες και βίντεο, να εξετάσει μια περιοχή σε επίπεδο $pixel$, να ρυθμίσει το χρώμα και την

αντίθεση, να δημιουργήσει γραφικές παραστάσεις και ιστογράμματα και να επιλέξει και να διαχειριστεί περιοχές ενδιαφέροντος (MathWorks.com, 2012).

Στον παρακάτω πίνακα (Πίνακας 3) φαίνονται οι βασικότερες συναρτήσεις της εργαλειοθήκης επεξεργασίας εικόνας και μία σύντομη περιγραφή τους.

Πίνακας 3: Βασικές συναρτήσεις της εργαλειοθήκης επεξεργασίας εικόνας

Συνάρτηση	Περιγραφή
imread	Διαβάζει και αποθηκεύει σε μεταβλητή ένα αρχείο εικόνας
imwrite	Αποθηκεύει σε αρχείο μία εικόνα
imfinfo	Εμφανίζει πληροφορίες για ένα αρχείο εικόνας
rgb2gray	Μετατρέπει μία έγχρωμη εικόνα σε εικόνα αποχρώσεων του γκρι
im2double	Μετατρέπει τις τιμές μίας εικόνας σε πραγματικές διπλής ακριβείας
imshow	Εμφανίζει μία εικόνα
imhist	Εμφανίζει το ιστόγραμμα μίας εικόνας
imadjust	Τροποποιεί τις τιμές έντασης φωτεινότητας μίας εικόνας
histeq	Εκτελεί εξισορρόπηση ιστογράμματος εικόνας
imnoise	Προσθέτει θόρυβο σε μία εικόνα
imfilter	Εκτελεί φιλτράρισμα μία εικόνας
fspecial	Δημιουργεί τις μάσκες των βασικότερων χωρικών φίλτρων

wiener2	Εκτελεί φιλτράρισμα με το προσαρμοζόμενο φίλτρο wiener
medfilt2	Εκτελεί φιλτράρισμα με το φίλτρο ενδιάμεσης τιμής
ordfilt2	Εκτελεί φιλτράρισμα με φίλτρα στατιστικής διάταξης
impoly	Επιτρέπει την επιλογή των pixels μίας ορθογώνιας περιοχής ενδιαφέροντος της εικόνας

3 Σχεδίαση και Υλοποίηση Εφαρμογής

Στην ενότητα αυτή παρουσιάζεται η σχεδίαση και η υλοποίηση της εφαρμογής επεξεργασίας εικόνας της παρούσας εργασίας με το περιβάλλον GUIDE και την εργαλειοθήκη επεξεργασίας εικόνας (Image Processing Toolbox) του Matlab έκδοσης 2012a.

Σκοπός της εφαρμογής είναι η παροχή ενός γραφικού εκπαιδευτικού περιβάλλοντος σε Matlab για την υποβοήθηση εκμάθησης βασικών και προχωρημένων εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας. Η εφαρμογή θα μπορεί να χρησιμοποιηθεί ως βοήθημα από τους σπουδαστές για την αναλυτικότερη παρουσίαση και επεξήγηση βασικών και προχωρημένων τεχνικών της ψηφιακής επεξεργασίας εικόνας, όπως η εφαρμογή φίλτρων σε εικόνες, η ανάλυση εικόνας στο πεδίο της συχνότητας, κ.ο.κ.

Επισημαίνεται ότι στην τρέχουσα έκδοσή της η εφαρμογή υποστηρίζει το χωρικό φιλτράρισμα εικόνων αποχρώσεων του γκρι. Ωστόσο, δεδομένου της ανοικτής αρχιτεκτονικής της, η εφαρμογή είναι εύκολα επεκτάσιμη και μελλοντικά μπορεί να καλύψει όλες τις τεχνικές επεξεργασίας, που εφαρμόζονται στην ψηφιακή επεξεργασία εικόνας.

Τέλος, αξίζει να αναφερθεί ότι η καινοτομία της συγκεκριμένης εφαρμογής είναι η αναλυτικότητα, βήμα προς βήμα επεξήγηση της τεχνικής του χωρικού φιλτραρίσματος σε επίπεδο pixel, παρουσιάζοντας όλους τους ενδιάμεσους υπολογισμούς και ενσωματώνοντας τα βασικότερα χωρικά φίλτρα. Επιπλέον, παρέχει τη δυνατότητα στον χρήστη να δημιουργήσει και να πειραματιστεί με τις δικές του χωρικές μάσκες.

Τα βήματα που ακολουθήθηκαν για την κατασκευή της εφαρμογής συνοψίζονται στα εξής:

- Καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων της εφαρμογής.

- Σχεδίαση των γραφικών διεπαφών και των χειριστηρίων.
- Προγραμματισμός της συμπεριφοράς των χειριστηρίων και της αλληλεπίδρασης με τον χρήστη.

3.1 Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις σχετίζονται με τις λειτουργίες που πρέπει να υποστηρίζει η εφαρμογή. Η καταγραφή τους είναι σημαντική προκειμένου η σχεδίαση της εφαρμογής να εστιάζει σε αυτές. Πιο συγκεκριμένα, η εφαρμογή υποστηρίζει τις εξής λειτουργίες:

- Φόρτωση της αρχικής εικόνας
- Αποθήκευση της φιλτραρισμένης εικόνας
- Επιλογή μίας περιοχής ενδιαφέροντος, η οποία θα χρησιμοποιηθεί προκειμένου να γίνει αναπαράσταση της διαδικασίας του χωρικού φιλτραρίσματος
- Επιλογή ενός pixel της περιοχής ενδιαφέροντος, το οποίο θα χρησιμοποιηθεί ως κεντρικό για την επίδειξη του φιλτραρίσματος
- Επιλογή φίλτρου
- Επιλογή παραμέτρων του επιλεγμένου φίλτρου
- Εκτέλεση φιλτραρίσματος
- Σύγκριση ιστογραμμάτων πριν και μετά το φιλτράρισμα
- Αναλυτική επεξήγηση της διαδικασίας φιλτραρίσματος μέσω επίδειξης

3.2 Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις σχετίζονται με τους περιορισμούς που θα πρέπει να ικανοποιεί η εφαρμογή προκειμένου να λειτουργεί ικανοποιητικά. Πιο συγκεκριμένα:

- Υποστηρίζονται εικόνες με 256 επίπεδα γκρι, λόγω της απλότητας που παρουσιάζουν στη διαχείρισή τους. Συνεπώς, δεν υποστηρίζονται οι έγχρωμες (RGB) εικόνες.

- Υποστηρίζονται αναλύσεις οθόνης τουλάχιστον 1024x768, προκειμένου να φαίνονται σωστά όλα τα χειριστήρια της εφαρμογής.
- Υποστηρίζονται τα βασικά χωρικά φίλτρα.
- Υποστηρίζεται η δυνατότητα να δημιουργήσει ο χρήστης τη δική του μάσκα γραμμικού χωρικού φίλτρου.
- Υποστηρίζονται οι μάσκες μεγέθους 3x3 και 5x5.
- Ο χρόνος που απαιτείται για την επεξεργασία – φιλτράρισμα μίας εικόνας δεν πρέπει να ξεπερνάει τα 2 ή 3 δευτερόλεπτα, προκειμένου να μην δημιουργούνται ενοχλητικές καθυστερήσεις για τον χρήστη.
- Οι γραφικές διεπαφές θα πρέπει να είναι λιτές και φιλικές προς τον χρήστη, ώστε να μην προκαλείται σύγχυση ή γνωστική υπερφόρτωση στον χρήστη.
- Η εφαρμογή θα τρέχει μέσα από το περιβάλλον GUIDE του Matlab 2012a, συνεπώς απαιτείται ο χρήστης να διαθέτει το Matlab έκδοσης τουλάχιστον 2012a.

3.3 Σχεδίαση Γραφικών Διεπαφών

Η εφαρμογή αποτελείται από 2 ειδών παράθυρα:

1. Το βασικό παράθυρο (αρχική οθόνη) της εφαρμογής και
2. Τα παράθυρα επεξήγησης της διαδικασίας φιλτραρίσματος, ανάλογα με το επιλεγμένο φίλτρο.

Τα παράθυρα επεξήγησης της διαδικασίας φιλτραρίσματος διακρίνονται στα εξής:

1. Φιλτράρισμα με τα βασικά γραμμικά φίλτρα. Αφορά τα φίλτρα μέσης τιμής (average), gauss, το χειροκίνητο που μπορεί να ορίσει ο ίδιος ο χρήστης τα βάρη της μάσκας του φίλτρου, καθώς και του γεωμετρικού μέσου.
2. Φιλτράρισμα με τα μη γραμμικά φίλτρα στατιστικής διάταξης. Αφορά τα φίλτρα ελάχιστης τιμής (min), μέγιστης τιμής (max) και ενδιάμεσης τιμής (median).
3. Φιλτράρισμα με το φίλτρο περικεκομμένου μέσου (alpha trimmed).

4. Φιλτράρισμα με το προσαρμοζόμενο φίλτρο wiener.
5. Φιλτράρισμα με το φίλτρο Sobel (1^{ης} παραγώγου).
6. Φιλτράρισμα με το Laplacian φίλτρο (2^{ης} παραγώγου).

Ωστόσο, προκειμένου να υπάρχει συνέπεια στην απεικόνιση της διαδικασίας φιλτραρίσματος, τα έξι (6) προηγούμενα παράθυρα έχουν την ίδια δομή – διάταξη και παρόμοια χειριστήρια.

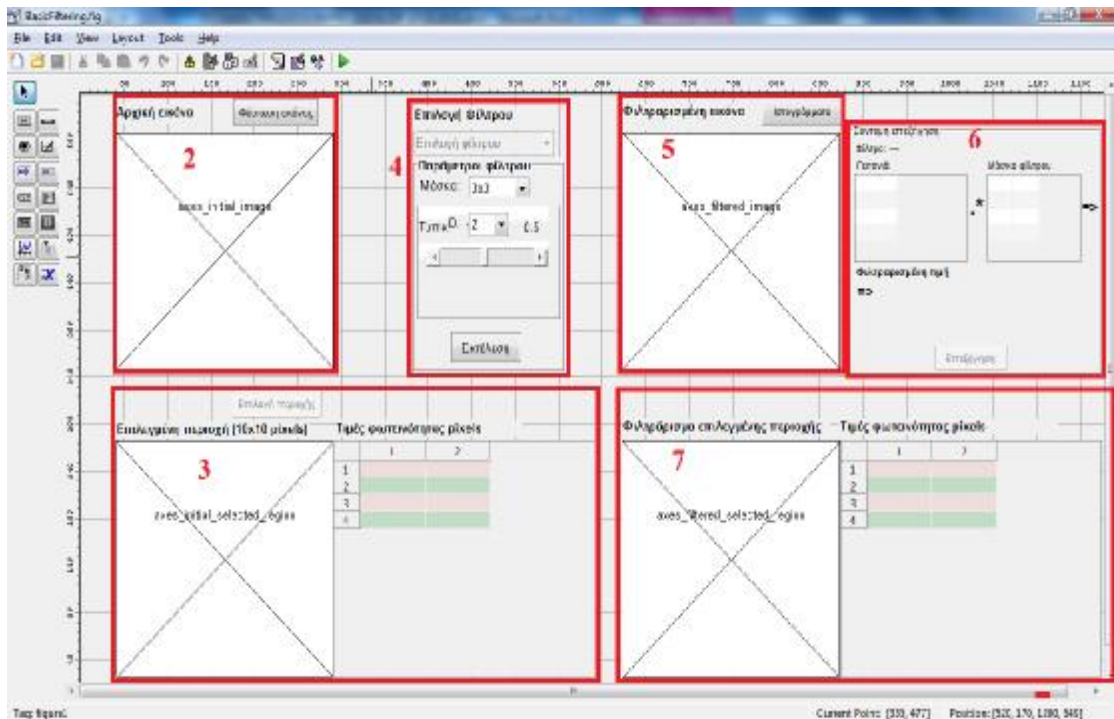
Επισημαίνεται, ότι αν και το φίλτρο γεωμετρικού μέσου δεν ανήκει στην κατηγορία των γραμμικών φίλτρων, ωστόσο η επεξήγησή του ήταν εύκολο να γίνει χρησιμοποιώντας το παράθυρο επεξήγησης των βασικών γραμμικών φίλτρων.

3.3.1 Το Βασικό Παράθυρο της Εφαρμογής

Το βασικό παράθυρο της εφαρμογής ονομάζεται BasicFiltering.fig και αποτελείται από τα εξής επτά (7) τμήματα:

1. κεντρικό μενού,
2. αρχική εικόνα (πριν το φιλτράρισμα),
3. περιοχή ενδιαφέροντος και πίνακας τιμών φωτεινοτήτων των pixels της αρχικής εικόνας (πριν το φιλτράρισμα)
4. παράμετροι φίλτρων,
5. τελική εικόνα (μετά το φιλτράρισμα),
6. σύντομη επεξήγηση φιλτραρίσματος,
7. περιοχή ενδιαφέροντος και πίνακας τιμών φωτεινοτήτων των pixels της τελικής εικόνας (μετά το φιλτράρισμα)

Στην εικόνα που ακολουθεί (Εικόνα 30) παρουσιάζεται το βασικό παράθυρο (κεντρική ή αρχική οθόνη) της εφαρμογής και τα χειριστήριά του, όπως φαίνονται μέσα από το περιβάλλον GUIDE του Matlab. Οι διακριτές κόκκινες περιοχές αντιπροσωπεύουν τα τμήματα του παραθύρου. Σημειώνεται ότι το κεντρικό μενού της εφαρμογής δεν είναι άμεσα ορατό στο περιβάλλον GUIDE, αλλά ενεργοποιείται μέσω της επιλογής «Tools» → «Menu Editor...».



Εικόνα 30: Το κεντρικό παράθυρο εφαρμογής, όπως φαίνεται στο GUIDE του Matlab. Τα τμήματα του παραθύρου εμφανίζονται με κόκκινη σήμανση

Στον πίνακα που ακολουθεί (Πίνακας 4) παρουσιάζονται όλα τα χειριστήρια του βασικού παραθύρου BasicFiltering.fig της εφαρμογής.

Πίνακας 4: Χειριστήρια του βασικού παραθύρου BasicFiltering.fig

A/A	Τύπος	Όνομα	Λεκτικό	Περιγραφή λειτουργίας
1	menu	file	Αρχείο	Μενού
2	menu	file_saveImage	Αποθήκευση εικόνας	Αποθήκευση εικόνας
3	menu	file_exit	Έξοδος	Τερματισμός εφαρμογής
4	Push button	btn_load_image	Φόρτωση εικόνας	Πλήκτρο για τη φόρτωση εικόνας
5	Static text	text1	Αρχική εικόνα	---

6	Axes	axes_initial_image	---	Περιοχή εμφάνισης της αρχικής φορτωμένης εικόνας
7	Push button	btn_select_region	Επιλογή περιοχής	Πλήκτρο επιλογής μίας περιοχής ενδιαφέροντος της αρχικής εικόνας
8	Static text	text13	Επιλεγμένη περιοχή (10x10 pixels)	---
9	axes	axes_initial_ _selected_region	---	Περιοχή εμφάνισης της περιοχής ενδιαφέροντος της αρχικής εικόνας
10	Static text	text12	Τιμές φωτεινότητας pixels	---
11	table	tablePixelValues	---	Πίνακας εμφάνισης των τιμών φωτεινότητας των pixels της επιλεγμένης περιοχής ενδιαφέροντος
12	Static text	text3	Επιλογή Φίλτρου	---
13	panel	uipanel_filter_ _parameters	Παράμετροι φίλτρου	Ομαδοποιεί τις παραμέτρους επιλογής φίλτρου
14	Push button (popupmenu)	popupmenu_select_ _filter	Επιλογή Φίλτρου	Λίστα επιλογής του φίλτρου

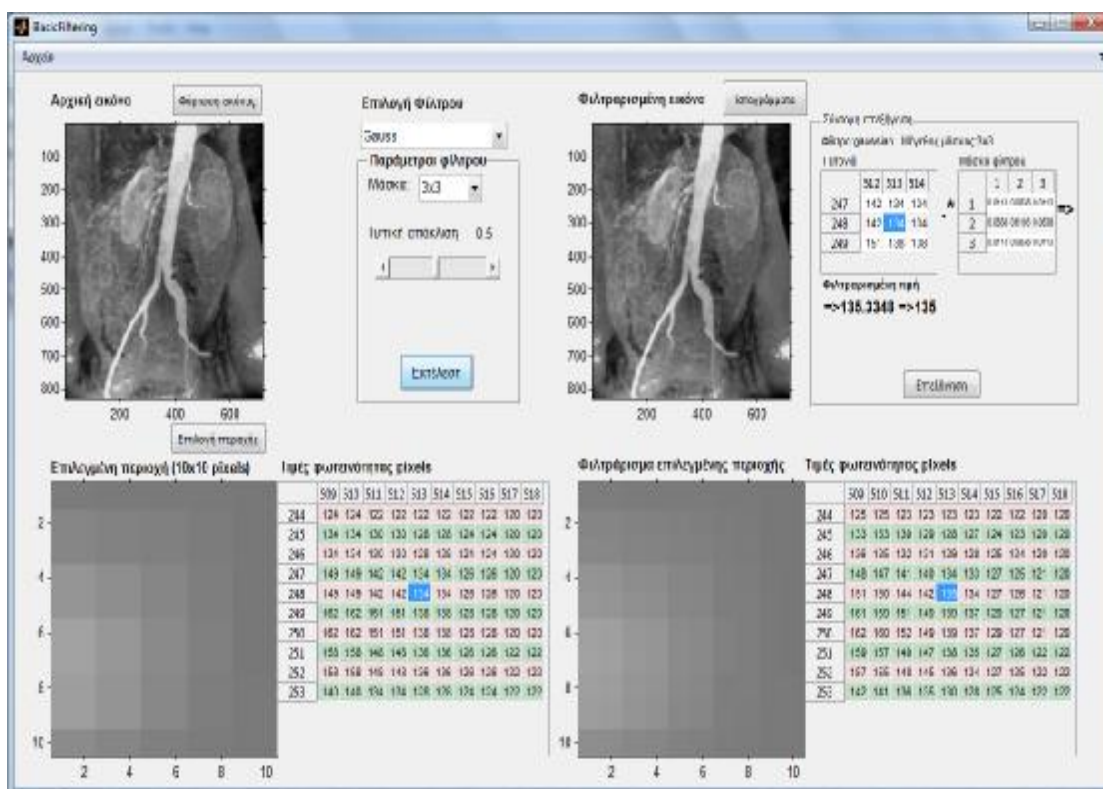
15	Static text	text4	Μάσκα:	---
16	Push button (popupmenu)	popupmenu_filter_ _mask_size	3x3	Λίστα επιλογής του μεγέθους της μάσκας του φίλτρου
17	table	userTable	---	---
18	Static text	text5	Τυπική απόκλιση	---
19	slider	slider_standard_ _deviation	0	Μπάρα κύλισης για την επιλογή της τιμής της τυπικής απόκλισης, όταν είναι επιλεγμένο το φίλτρο Gauss
20	Static text	text10	D:	---
21	Push button (popupmenu)	popupmenu_alpha_ _trimmed_d	2	Λίστα επιλογής της τιμής περικοπής, όταν είναι επιλεγμένο το φίλτρο περικεκομμένου μέσου
22	Static text	sdText	0.5	Εμφανίζει την επιλεγμένη τιμή της τυπικής απόκλισης, όταν είναι επιλεγμένο το φίλτρο Gauss
23	Push button	btn_filter_image	Εκτέλεση	Πλήκτρο για την εκτέλεση του φιλτραρίσματος

24	Static text	text2	Φιλτραρισμένη εικόνα	---
25	axes	axes_filtered_image	---	Περιοχή εμφάνισης της φιλτραρισμένης εικόνας
26	Push button	btn_show_histograms	Ιστογράμματα	Πλήκτρο για την εμφάνιση και σύγκριση των ιστογραμμάτων της εικόνας και της επιλεγμένης περιοχής ενδιαφέροντος πριν και μετά το φιλτράρισμα
27	panel	uipanel_reg_test	Σύντομη επεξήγηση	Ομαδοποιεί τα χειριστήρια σύντομης επεξήγησης της διαδικασίας φιλτραρίσματος
28	Static Text	selectedFilterNameText	Φίλτρο: ---	Εμφανίζει τα στοιχεία του επιλεγμένου φίλτρου, όπως το όνομά του και το μέγεθος της επιλεγμένης μάσκας
29	Static Text	neighborhoodText	Γειτονιά	---
30	table	neighborhoodPixelsTable	---	Εμφανίζει πίνακα με τις τιμές φωτεινότητας των pixel της γειτονιάς που έχει ως κέντρο το επιλεγμένο pixel της περιοχής ενδιαφέροντος της αρχικής εικόνας

31	Static Text	text20	.*	---
32	Static Text	filterMaskText	Μάσκα φίλτρου	---
33	table	maskTable	---	Εμφανίζει πίνακα με τις τιμές των βαρών της μάσκας του επιλεγμένου φίλτρου
34	Static Text	text21	=>	---
35	Static Text	text26	Φιλτραρισμένη τιμή	---
36	Static Text	maskingText	=>	Εμφανίζει μία σύντομη επεξήγηση της φιλτραρισμένης τιμής του επιλεγμένου pixel της περιοχής ενδιαφέροντος
37	Push button	btn_explain_filtering	Επεξήγηση	Πλήκτρο που ανοίγει το κατάλληλο παράθυρο αναλυτικής επεξήγησης της διαδικασίας φιλτραρίσματος, ανάλογα με το επιλεγμένο φίλτρο
38	Static Text	text15	Φιλτράρισμα επιλεγμένης περιοχής	---
39	axes	axes_filtered_ _selected_region	---	Περιοχή εμφάνισης της φιλτραρισμένης περιοχής ενδιαφέροντος της αρχικής

				εικόνας
40	Static Text	text14	Τιμές φωτεινότητας pixels	---
41	table	tableFilteredPixelValues	---	Πίνακας εμφάνισης των φιλτραρισμένων τιμών φωτεινότητας των pixels της επιλεγμένης περιοχής ενδιαφέροντος

Στην εικόνα που ακολουθεί (Εικόνα 31) παρουσιάζεται το βασικό παράθυρο και τα χειριστήρια της εφαρμογής κατά τον χρόνο εκτέλεσης.



Εικόνα 31: Το βασικό παράθυρο (BasicFiltering.fig) κατά τον χρόνο εκτέλεσης

3.3.2 Τα Παράθυρα Επεξήγησης της Διαδικασίας Φιλτραρίσματος

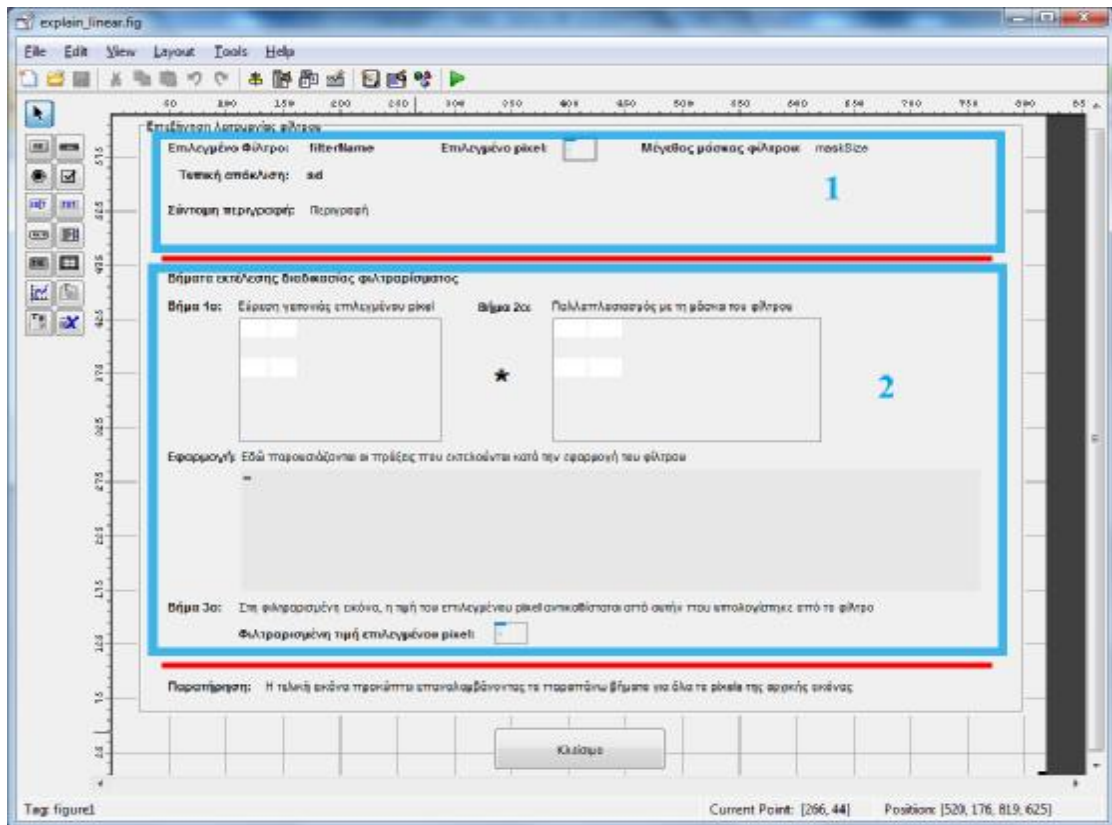
Για την αναλυτική επεξήγηση (επίδειξη) της διαδικασίας φιλτραρίσματος χρησιμοποιούνται τα εξής έξι (6) παράθυρα:

1. Explain_linear.fig, για φιλτράρισμα με τα φίλτρα:
 - μέσης τιμής (average),
 - gauss,
 - γεωμετρικού μέσου και
 - χειροκίνητο, που ορίζει ο χρήστης τα βάρη της μάσκας του φίλτρου
2. Explain_order.fig, για φιλτράρισμα με τα φίλτρα:
 - min,
 - max και
 - median
3. Explain_alpha_trimmed.fig, για φιλτράρισμα με το φίλτρο περικεκομμένου μέσου (alpha trimmed)
4. Explain_wiener.fig, για φιλτράρισμα με το προσαρμοζόμενο φίλτρο wiener
5. Explain_Sobel.fig, για φιλτράρισμα με το φίλτρο Sobel (1^{ης} παραγώγου)
6. Explain_laplacian.fig, για φιλτράρισμα με το Laplacian φίλτρο (2^{ης} παραγώγου)

Όλα τα παράθυρα επεξήγησης ακολουθούν την ίδια δομή και έχουν παρόμοια χειριστήρια, προκειμένου να υπάρχει συνέπεια στην απεικόνιση της διαδικασίας φιλτραρίσματος. Πιο συγκεκριμένα διακρίνονται σε δύο τμήματα:

1. Στοιχεία επιλεγμένου φίλτρου, όπως όνομα, μέγεθος μάσκας, σύντομη περιγραφή κοκ.
2. Αριθμημένα βήματα επεξήγησης της διαδικασίας φιλτραρίσματος και επίδειξης των μαθηματικών υπολογισμών (πράξεων).

Στην παρακάτω εικόνα (Εικόνα 32) παρουσιάζεται το παράθυρο επεξήγησης explain_linear.fig και τα χειριστήριά του, όπως φαίνεται μέσα από το περιβάλλον GUIDE του Matlab. Οι διακριτές μπλε περιοχές αντιπροσωπεύουν τα δύο τμήματα του παραθύρου.



Εικόνα 32: Το παράθυρο επεξήγησης explain_linear.fig, όπως φαίνεται στο GUIDE του Matlab. Τα τμήματα του παραθύρου φαίνονται με μπλε σήμανση.

Στον πίνακα που ακολουθεί (Πίνακας 5) παρουσιάζονται ενδεικτικά τα χειριστήρια του παραθύρου επεξήγησης explain_linear.fig.

Πίνακας 5: Χειριστήρια του παραθύρου επεξήγησης explain_linear.fig

A/A	Τύπος	Όνομα	Λεκτικό	Περιγραφή λειτουργίας
1	panel	explainPanel	Επεξήγηση λειτουργίας φίλτρου	Ομαδοποιεί όλα τα χειριστήρια που σχετίζονται με τη λειτουργία του φιλτραρίσματος
2	Static Text	text3	Επιλεγμένο Φίλτρο:	---

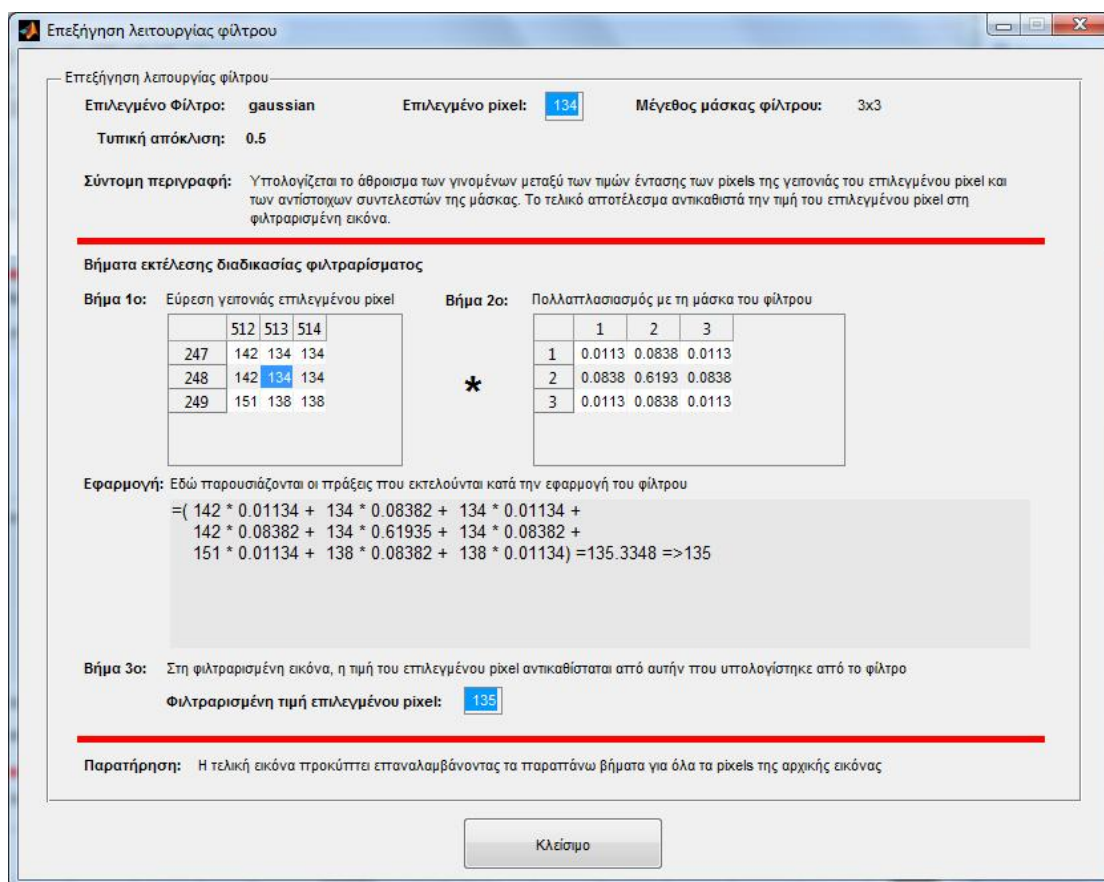
3	Static Text	filterNameText	filterName	Εμφανίζει το όνομα του επιλεγμένου φίλτρου
4	Static Text	text10	Επιλεγμένο pixel:	---
5	table	selectedPixelTable	---	Πίνακας ενός στοιχείου, που εμφανίζει την τιμή φωτεινότητας του επιλεγμένου pixel, με βάση το οποίο γίνεται η επίδειξη του φιλτραρίσματος
6	Static Text	text13	Μέγεθος μάσκας φίλτρου:	---
7	Static Text	maskSizeText	maskSize	Εμφανίζει το μέγεθος της μάσκας του φίλτρου
8	Static Text	sdText	Τυπική απόκλιση:	Είναι ορατό μόνο όταν έχει επιλεγεί το φίλτρο Gauss
9	Static Text	sdValueText	sd	Εμφανίζει την τιμή της τυπικής απόκλισης, όταν έχει επιλεγεί το φίλτρο Gauss
10	Static Text	text5	Σύντομη περιγραφή:	---
11	Static Text	shortDescriptionText	Περιγραφή	Εμφανίζει μία σύντομη λεκτική περιγραφή της λειτουργίας του επιλεγμένου φίλτρου

12	Static Text	text1	Βήματα εκτέλεσης διαδικασίας φιλτραρίσματος	---
13	Static Text	text2	Βήμα 1ο:	---
14	Static Text	text9	Εύρεση γειτονιάς επιλεγμένου pixel	---
15	table	neighborhoodTable	--	Εμφανίζει σε πίνακα τις τιμές φωτεινότητας των pixels, που ανήκουν στη γειτονιά του επιλεγμένου pixel
16	Static Text	text45	*	---
17	Static Text	text44	Βήμα 2ο:	---
18	Static Text	text43	Πολλαπλασιασμός με τη μάσκα του φίλτρου	---
19	table	maskTable	---	Εμφανίζει σε πίνακα τα βάρη (συντελεστές) της μάσκας του επιλεγμένου φίλτρου
20	Static Text	text7	Εφαρμογή:	---
21	Static Text	text11	Εδώ παρουσιάζονται οι πράξεις που εκτελούνται κατά	---

			την εφαρμογή του φίλτρου	
22	Static Text	mathsText	=	Εμφανίζει αναλυτικά τις πράξεις για τον υπολογισμό της φιλτραρισμένης τιμής για το επιλεγμένο pixel
23	Static Text	text8	Βήμα 3ο:	---
24	Static Text	text17	Στη φιλτραρισμένη εικόνα, η τιμή του επιλεγμένου pixel αντικαθίσταται από αυτήν που υπολογίστηκε από το φίλτρο	---
25	Static Text	text20	Φιλτραρισμένη τιμή επιλεγμένου pixel:	---
26	table	filteredValueTable	---	Πίνακας ενός στοιχείου, που εμφανίζει την φιλτραρισμένη τιμή του επιλεγμένου pixel
27	Static Text	text19	Παρατήρηση:	---
28	Static Text	text18	Η τελική εικόνα προκύπτει επαναλαμβάνοντας τα παραπάνω βήματα για όλα τα	---

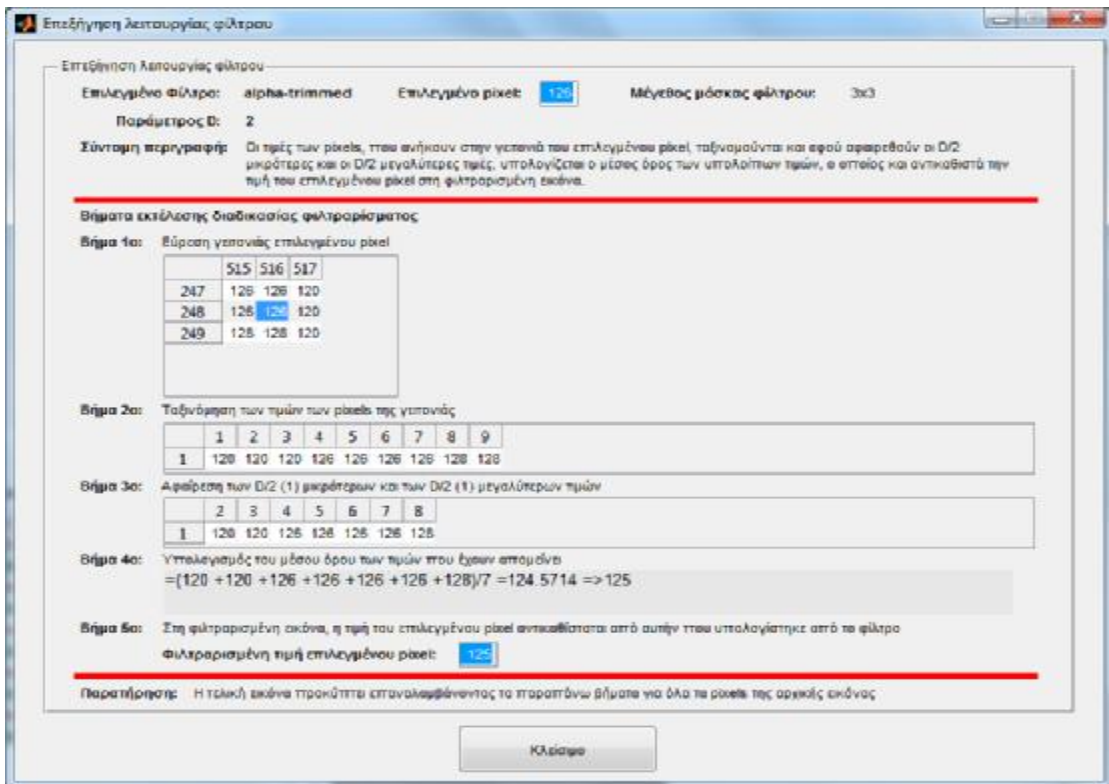
			pixels της αρχικής εικόνας	
29	Push Button	closeButton	Κλείσιμο	Κλείνει το παράθυρο επεξήγησης

Στην εικόνα που ακολουθεί (Εικόνα 33) παρουσιάζεται το παράθυρο επεξήγησης explain_linear.fig και τα χειριστήριά του κατά τον χρόνο εκτέλεσης.

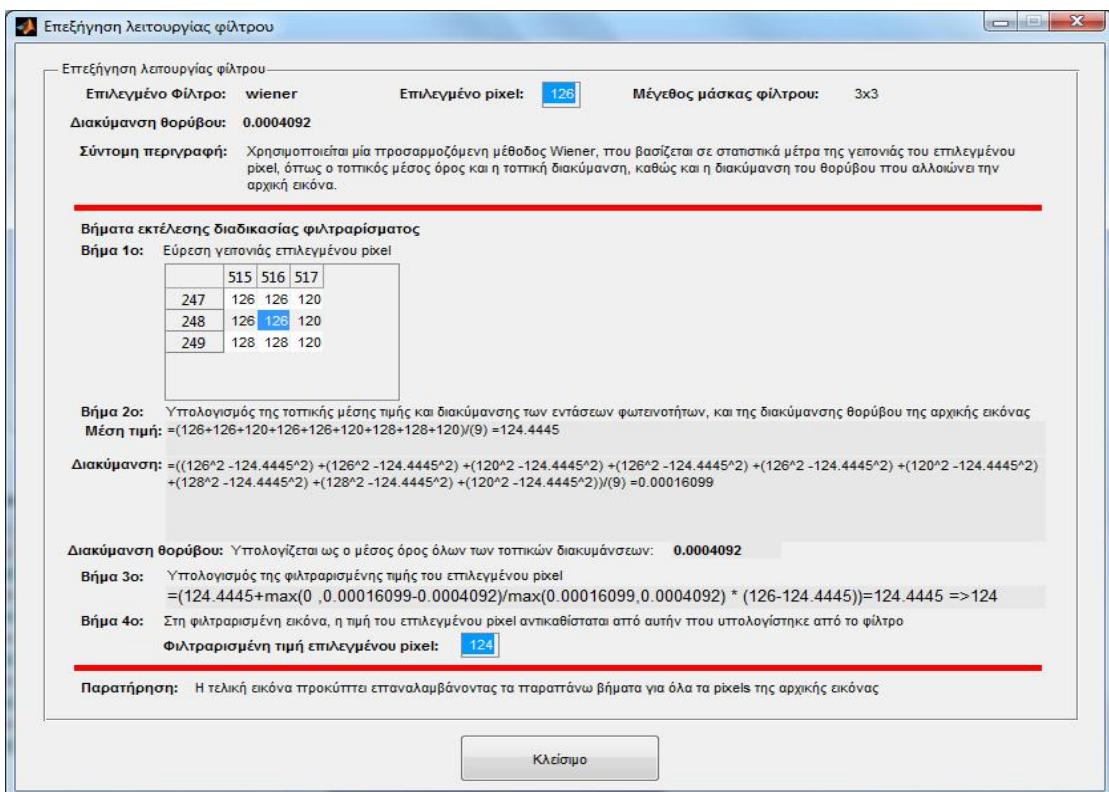


Εικόνα 33: Το παράθυρο επεξήγησης explain_linear.fig κατά τον χρόνο εκτέλεσης

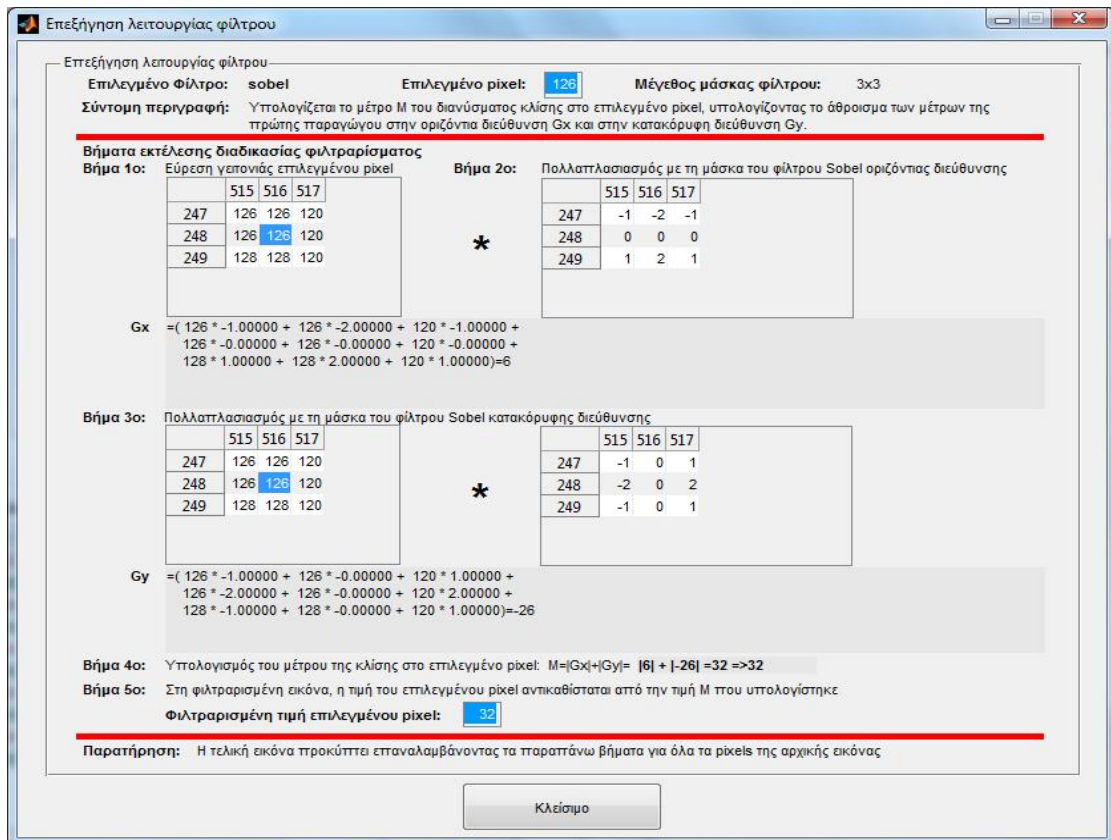
Επίσης, στις επόμενες εικόνες (Εικόνα 34 ως Εικόνα 36) παρουσιάζονται με τη σειρά τα παράθυρα explain_alpha_trimmed.fig, explain_wiener.fig και explain_sobel.fig, κατά τον χρόνο εκτέλεσης της εφαρμογής.



Εικόνα 34: Το παράθυρο επεξήγησης explain_alpha_trimmed.fig κατά τον χρόνο εκτέλεσης



Εικόνα 35: Το παράθυρο επεξήγησης explain_wiener.fig κατά τον χρόνο εκτέλεσης



Εικόνα 36: Το παράθυρο επεξήγησης explain_sobel.fig κατά τον χρόνο εκτέλεσης

3.4 Προγραμματισμός Συμπεριφοράς και Αλληλεπίδρασης

Στην υποενότητα αυτή καταγράφεται ο προγραμματισμός (κώδικας Matlab) της συμπεριφοράς των χειριστηρίων της εφαρμογής και της συνολικής αλληλεπίδρασης με τον χρήστη. Υπενθυμίζεται ότι κάθε παράθυρο της εφαρμογής αποτελείται από:

- ένα αρχείο με επέκταση «.fig», το οποίο σχετίζεται με τη γραφική διεπαφή του παραθύρου και
- ένα αρχείο με επέκταση «.m», το οποίο περιέχει τον κώδικα Matlab που ρυθμίζει τη συμπεριφορά των χειριστηρίων.

Ο κύριος κορμός του κώδικα βρίσκεται στο αρχείο BasicFiltering.m, το οποίο είναι το M-file του βασικού παραθύρου BasicFiltering.fig. Τα υπόλοιπα παράθυρα, δηλαδή τα παράθυρα επεξήγησης της διαδικασίας φιλτραρίσματος, προφανώς θα πρέπει να καλούνται από το βασικό παράθυρο. Επιπλέον, τα παράθυρα αυτά θα πρέπει να έχουν πρόσβαση σε κάποιες μεταβλητές, προκειμένου να λειτουργήσουν

σωστά. Για αυτό, είναι σκόπιμο να οριστούν αυτές οι μεταβλητές ως καθολικές, ώστε να είναι ορατές από τα δευτερεύοντα παράθυρα της εφαρμογής.

Οι καθολικές μεταβλητές της εφαρμογής, καθώς και η περιγραφή τους φαίνονται στον ακόλουθο πίνακα (Πίνακας 6).

Πίνακας 6: Οι καθολικές μεταβλητές της εφαρμογής

Όνομα μεταβλητής	Περιγραφή μεταβλητής
InitialImage	Πίνακας με τις τιμές φωτεινότητας της αρχικής εικόνας
filteredImage	Πίνακας με τις τιμές φωτεινότητας της φιλτραρισμένης εικόνας
selected_region_from_image	Πίνακας με τις τιμές φωτεινότητας της επιλεγμένης περιοχής (διαστάσεις 10x10)
selected_filtered_region_from_image	Πίνακας με τις τιμές φωτεινότητας της φιλτραρισμένης επιλεγμένης περιοχής (διαστάσεις 10x10)
filterName	Όνομα του επιλεγμένου φίλτρου
maskSize	Μέγεθος του επιλεγμένου φίλτρου
hFilter	Πίνακας με τα βάρη του επιλεγμένου φίλτρου
selectedPixelValue	Τιμή φωτεινότητας του επιλεγμένου pixel από την περιοχή ενδιαφέροντος
dataArea	Πίνακας με τις τιμές φωτεινότητας της γειτονιάς του επιλεγμένου pixel

roi_fixed	Πίνακας με τις διορθωμένες συντεταγμένες της επιλεγμένης περιοχής ενδιαφέροντος
Sigma	Τυπική απόκλιση φίλτρου Gauss
noiseVarianceWiener	Διακύμανση θορύβου φίλτρου Wiener
alpha_trimmed_d	Συντελεστής περικοπής φίλτρου alpha-trimmed
sx	Μάσκα Sobel οριζόντιας κατεύθυνσης

3.4.1 Προγραμματισμός του Βασικού Παραθύρου

Ο προγραμματισμός του M-file του βασικού παραθύρου της εφαρμογής ουσιαστικά καλύπτει όλες τις λειτουργικές απαιτήσεις, δηλαδή τις απαιτούμενες λειτουργίες της εφαρμογής. Αυτές αναλύονται παρακάτω, παραθέτοντας και το σχετικό κώδικα Matlab.

3.4.1.1 Φόρτωση Εικόνας

Ο χρήστης πατώντας το πλήκτρο «Φόρτωση εικόνας» (btn_load_image) μπορεί να φορτώσει μία εικόνα αποχρώσεων του γκρι. Η επιλογή της εικόνας γίνεται μέσω κατάλληλου παραθύρου, ενώ η αποθήκευση της φορτωμένης εικόνας γίνεται στην καθολική μεταβλητή InitialImage. Ταυτόχρονα, ενημερώνονται δυναμικά τα διάφορα χειριστήρια της εφαρμογής, όπως η περιοχή εμφάνισης της εικόνας κοκ.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο btn_load_image:

```
% --- Executes on button press in btn_load_image.
function btn_load_image_Callback(hObject, eventdata, handles)
% hObject    handle to btn_load_image (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global InitialImage;
global filteredImage;
global selected_region_from_image;
```

```

global selected_filtered_region_from_image;
global maskSize;

% Ο χρήστης επιλέγει την εικόνα που θα εισάγει
[filename, pathname] = uigetfile('*.');
if isequal(filename,0)
    disp('Δεν επιλέξατε κάποια εικόνα');
else
    % Φορτώνω την εικόνα
    InitialImage = imread(fullfile(pathname, filename));

    % Καθαρίζω τις μεταβλητές
    filteredImage=[];
    selected_region_from_image=[];
    selected_filtered_region_from_image=[];

    % Εμφανίζω την εικόνα
    iptsetpref('imshowAxesVisible','on');
    axes(handles.axes_initial_image);
    imshow(InitialImage);

    % Ενεργοποιώ το πλήκτρο επιλογής περιοχής
    set(handles.btn_select_region,'Enable','on');

    % Απενεργοποιώ τις επιλογές φίλτρου
    set(handles.popupmenu_select_filter,'Enable','off');
    set(handles.uipanel_filter_parameters,'Visible','off');
    % Απενεργοποιώ το πλήκτρο επεξήγησης του φιλτραρίσματος
    set(handles.btn_explain_filtering,'Enable','off');

    % Καθαρίζω το παράθυρο με την προηγούμενη φιλτραρισμένη εικόνα
    cla(handles.axes_filtered_image,'reset');

    % Καθαρίζω τους πίνακες των pixels (2 Πίνακες)
    set(handles.tablePixelValues,
'Data',selected_region_from_image,'ColumnWidth',{25},
'ColumnName',1:10, 'RowName',1:10);
    set(handles.tableFilteredPixelValues,
'Data',selected_filtered_region_from_image, 'ColumnWidth',{25},
'ColumnName',1:10, 'RowName',1:10);

    % Απενεργοποιώ τη δυνατότητα επιλογής κάποιου pixel
    set(handles.tablePixelValues,'Enable','inactive');

    % Καθαρίζω τις εικόνες με τα επιλεγμένα pixel (2 εικόνες)
    cla(handles.axes_initial_selected_region,'reset');
    cla(handles.axes_filtered_selected_region,'reset');

    % Καθαρίζω τους πίνακες της γειτονιάς και της μάσκας, και τα
λεκτικά
    set(handles.neighborhoodPixelsTable,
'Data',[],'ColumnWidth',{25}, 'ColumnName',[1:maskSize],
'RowName',[1:maskSize]);
    set(handles.maskTable, 'Data',[],'ColumnWidth',{25},
'ColumnName',[1:maskSize], 'RowName',[1:maskSize]);
    set(handles.selectedFilterNameText, 'String','Φίλτρο: ---');
    set(handles.maskingText, 'String','=>');

```

```

    % Απενεργοποιώ το πλήκτρο επεξήγησης
    set(handles.btn_explain_filtering, 'Enable', 'off');
end

```

3.4.1.2 Αποθήκευση Εικόνας

Μέσω του μενού «Αρχείο» \rightarrow «Αποθήκευση εικόνας» (`file_saveImage`), ο χρήστης μπορεί να αποθηκεύσει μία φιλτραρισμένη εικόνα. Η αποθήκευση γίνεται μέσω κατάλληλου παραθύρου, όπου ο χρήστης μπορεί να ορίσει τη διαδρομή και το όνομα του αρχείου. Επισημαίνεται ότι η τιμές φωτεινότητας των pixels της φιλτραρισμένης εικόνας βρίσκονται στην καθολική μεταβλητή `filteredImage`.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί η επιλογή μενού `file_saveImage`:

```

function file_saveImage_Callback(hObject, eventdata, handles)
% hObject    handle to file_saveImage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global filteredImage;

% Εμφανίζω παράθυρο αποθήκευσης αρχείου εικόνας
[FileName,PathName] = uinputfile('*.gif','Αποθήκευση φιλτραρισμένης
εικόνας','filtered_image.gif');
if isequal(FileName,0) || isequal(PathName,0)
    disp('Δεν αποθηκεύσατε');
else
    % Αποθηκεύω την εικόνα
    imwrite(filteredImage,fullfile(PathName,FileName));
    disp(['Αποθήκευση στο αρχείο: ' fullfile(PathName,FileName)]);
end

```

3.4.1.3 Τερματισμός Εφαρμογής

Μέσω του μενού «Αρχείο» \rightarrow «Έξοδος» (`file_exit`), ο χρήστης μπορεί να τερματίσει την εφαρμογή. Ο τερματισμός της εφαρμογής συνοδεύεται από παράθυρο επιβεβαίωσης.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί η επιλογή μενού `file_exit`:

```

function file_exit_Callback(hObject, eventdata, handles)
% hObject    handle to file_exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

h = questdlg('Κλείσιμο εφαρμογής;');
if (isequal(h,'Yes'))
    closereq;
    close all;
end

```

3.4.1.4 Επιλογή Περιοχής Ενδιαφέροντος της Αρχικής Εικόνας

Ο χρήστης πατώντας το πλήκτρο «Επιλογή περιοχής» (btn_select_region) μπορεί επιλέξει κάποια συγκεκριμένη περιοχή της φορτωμένης εικόνας, προκειμένου να γίνει επίδειξη της διαδικασίας φιλτραρίσματος με βάση την περιοχή αυτή. Βασική προϋπόθεση είναι να έχει φορτωθεί προηγουμένως κάποια εικόνα στην εφαρμογή.

Μόλις ο χρήστης πατήσει το πλήκτρο «Επιλογή περιοχής», αρχικά εμφανίζεται η φορτωμένη εικόνα (InitialImage) σε νέο παράθυρο, όπου ο χρήστης χρησιμοποιώντας το ποντίκι μαρκάρει την περιοχή ενδιαφέροντος και κάνοντας διπλό κλικ πάνω της, οι τιμές φωτεινότητας των pixels της επιλεγμένης περιοχής αποθηκεύονται στον πίνακα selected_region_from_image. Εξαιτίας του περιορισμένου χώρου του βασικού παραθύρου της εφαρμογής, επιλέχτηκε το μέγεθος της περιοχής ενδιαφέροντος να περιορίζεται αυτόματα σε μέγεθος 10x10 pixels, ανεξαρτήτως του μεγέθους της περιοχής που επιλέγει ο χρήστης. Αυτό γίνεται μέσω της συνάρτησης selectedRoilIndices().

Μόλις ο χρήστης επιλέξει την περιοχή ενδιαφέροντος από την αρχική εικόνα, τότε ενημερώνεται ο πίνακας tablePixelValues με τις τιμές φωτεινότητας της επιλεγμένης περιοχής, ενώ στην επάνω και στην αριστερή πλευρά του πίνακα εμφανίζεται και η αρίθμηση των συντεταγμένων των pixels της επιλεγμένης περιοχής. Ταυτόχρονα, επιλέγεται αυτόματα το κεντρικό pixel της περιοχής ενδιαφέροντος.

Επίσης, στο πλαίσιο axes_initial_selected_region εμφανίζεται η επιλεγμένη περιοχή, ενώ ενεργοποιείται το χειριστήριο επιλογής φίλτρου και καθαρίζονται όλα τα χειριστήρια του βασικού παραθύρου που σχετίζονται με τα αποτελέσματα του προηγούμενου φιλτραρίσματος.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο btn_select_region:

```

% --- Executes on button press in btn_select_region.
function btn_select_region_Callback(hObject, eventdata, handles)
% hObject    handle to btn_select_region (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global InitialImage;
global position;
global maskSize;

hf=figure;
imshow(InitialImage,'InitialMagnification', 200);
h = imrect();
position = wait(h);
position(3) = 10;
position(4) = 10;
if (ishandle(hf))
    close(hf);
end

global selected_region_from_image;
global selected_filtered_region_from_image;
global roi_fixed;

% Εμφάνιση επιλεγμένης περιοχής στο axes axes_initial_selected_region
[selected_region_from_image roi_fixed] = selectedRoiIndices(position,
InitialImage);
axes(handles.axes_initial_selected_region);
imshow(selected_region_from_image);

% Καθαρίζω τη μεταβλητή με τα φιλτραρισμένα επιλεγμένα pixels
selected_filtered_region_from_image=[];

% Εμφάνιση τιμών pixel στον πίνακα tablePixelValues
cnames = [];
for i = 0:roi_fixed(3)-1
    cnames = [cnames roi_fixed(1)+i];
end

rnames = [];
for i = 0:roi_fixed(4)-1
    rnames = [rnames roi_fixed(2)+i];
end

set(handles.tablePixelValues,'Data',selected_region_from_image,
'ColumnWidth',{25}, 'ColumnName',cnames, 'RowName',rnames);
jscroll=findobj(handles.tablePixelValues);
h=jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
jtable.setRowSelectionAllowed(true);
jtable.changeSelection(5-1,5-1, false, false); % Επιλέγω το κεντρικό
pixel (5,5)

```



```

% Απενεργοποιώ τη δυνατότητα επιλογής κάποιου pixel
set(handles.tablePixelValues, 'Enable', 'inactive');

% Ενεργοποιώ την επιλογή φίλτρου και εκτελώ το callback του
set(handles.popupmenu_select_filter, 'Enable', 'on');
popupmenu_select_filter_Callback(handles.popupmenu_select_filter,
 eventdata, handles);

% Καθαρίζω την εικόνα με τα φιλτραρισμένα επιλεγμένα pixel (άξονας:
axes_filtered_selected_region)
cla(handles.axes_filtered_selected_region, 'reset');

% Καθαρίζω τον πίνακα των φιλτραρισμένων pixels (Πίνακας:
tableFilteredPixelValues)
set(handles.tableFilteredPixelValues,
 'Data', selected_filtered_region_from_image, 'ColumnWidth', {25},
 'ColumnName', 1:10, 'RowName', 1:10);

% Καθαρίζω τους πίνακες της γειτονιάς και της μάσκας, και τα λεκτικά
set(handles.neighborhoodPixelsTable, 'Data', [], 'ColumnWidth', {25},
 'ColumnName', [1:maskSize], 'RowName', [1:maskSize]);
set(handles.maskTable, 'Data', [], 'ColumnWidth', {25},
 'ColumnName', [1:maskSize], 'RowName', [1:maskSize]);
set(handles.selectedFilterNameText, 'String', 'Φίλτρο: ---');
set(handles.maskingText, 'String', '=>');

% Απενεργοποιώ το πλήκτρο επεξήγησης
set(handles.btn_explain_filtering, 'Enable', 'off');

```

Στο σημείο αυτό είναι χρήσιμο να παρουσιάσουμε τον κώδικα της συνάρτησης `selectedRoiIndices()`, που βρίσκεται στο αρχείο `selectedRoiIndices.m` και υπολογίζει τις φωτεινότητες και τις συντεταγμένες της περιοχής ενδιαφέροντος της αρχικής εικόνας που έχει επιλέξει ο χρήστης.

Ο κώδικας του αρχείου `selectedRoiIndices.m` παρατίθεται παρακάτω:

```

function [ newI roi_fixed ] = selectedRoiIndices( roi, I )
%SELECTEDROIINDICES Summary of this function goes here
% Detailed explanation goes here

% Pixel Region Selected [xmin ymin width height]
% Πχ [106.104718056421 73.9827147620672 12.1930530779483
9.5249261691267]

%1ο πεδίο - xmin: στήλη στον πίνακα της εικόνας
%2ο πεδίο - ymin: γραμμή στον πίνακα της εικόνας
%3ο πεδίο - width: πλήθος διαδοχικών στηλών στον πίνακα της εικόνας,
%δηλαδή, ξεκινώντας από τη στήλη xmin μετρώ συνολικά width στήλες
%4ο πεδίο - height: πλήθος διαδοχικών γραμμών στον πίνακα της
εικόνας,
%δηλαδή, ξεκινώντας από τη γραμμή ymin μετρώ συνολικά height γραμμές

```

```

% Παρατήρηση: σε εικόνα μεγέθους 50x50
% Παρατήρηση: Στην εικόνα το πάνω αριστερά pixel έχει τιμή (1, 1)
% Παρατήρηση: Στην εικόνα το πάνω δεξιά pixel έχει τιμή (50, 1)
% Παρατήρηση: Στην εικόνα το κάτω αριστερά pixel έχει τιμή (1, 50)
% Παρατήρηση: Στην εικόνα το κάτω δεξιά pixel έχει τιμή (50, 50)
[rows cols]=size(I);

% Παράδειγμα ROI:  roi = [103.11    178.23    11.01    7.13]

% Αρχικά ρυθμίζω τις xmin, ymin της roi προσθέτοντας 1
roi(1) = roi(1) + 1;
roi(2) = roi(2) + 1;

% Μετατρέπω σε ακέραιες τις τιμές της ROI
roi = round(roi);
% Τώρα θα έχει γίνει έτσι:  roi = [104    179    11    7]

% Εδώ διορθώνω την επιλεγμένη περιοχή στην περίπτωση που είναι εκτός ορίων
if (roi(1)<=0)
    roi(1) = 1;
end
if (roi(2)<=0)
    roi(2) = 1;
end
if (roi(1) + roi(3)> cols)
    roi(1) = cols - roi(3) + 1;
end
if (roi(2) + roi(4)> rows)
    roi(2) = rows - roi(4) + 1;
end
% Στη συνέχεια δημιουργώ μία νέα εικόνα μεγέθους όσο είναι και η
επιλεγμένη
% περιοχή ROI, χρησιμοποιώντας τα width, height της roi
xmin = roi(1);
ymin = roi(2);
width = roi(3);
height = roi(4);

newI = zeros(height, width);

for i=1:height
    for j=1:width
        newI(i,j) = I(ymin+i-1, xmin+j-1);
    end
end

% Μετατρέπω σε uint8 για να μπορεί να εμφανιστεί η εικόνα με την
imshow
newI = uint8(newI);
roi_fixed = roi;
end

```

3.4.1.5 Επιλογή Φίλτρου

Ο χρήστης πατώντας στο πλήκτρο της λίστας επιλογών «Επιλογή φίλτρου» (popupmenu_select_filter) μπορεί να επιλέξει το επιθυμητό φίλτρο. Η λίστα αποτελείται από τις εξής επιλογές:

- Επιλογή φίλτρου
- Μέσης τιμής (average)
- Gauss
- Μεσαίου (Median)
- Min
- Max
- Wiener
- Alpha-trimmed
- Γεωμετρικού μέσου
- Sobel
- Laplacian
- Χειροκίνητο

Ανάλογα με την επιλογή του χρήστη ενεργοποιείται η δυνατότητα επιλογής μεγέθους της μάσκας του φίλτρου, ενώ ενημερώνονται δυναμικά τα χειριστήρια που σχετίζονται τις παραμέτρους του επιλεγμένου φίλτρου.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο της λίστας επιλογών popupmenu_select_filter:

```
% --- Executes on selection change in popupmenu_select_filter.
function popupmenu_select_filter_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu_select_filter (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu_select_filter contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu_select_filter

val = get(hObject, 'Value');
```

```

switch val
case 1
    % Ο χρήστης επέλεξε την 1η τιμή ('Επιλέξτε φίλτρο')
    set(handles.uipanel_filter_parameters, 'Visible', 'off');
    filterName = '';
case 2
    % Ο χρήστης επέλεξε την 2η τιμή ('Μέσης τιμής (average)')
    set(handles.uipanel_filter_parameters, 'Visible', 'on');
    set(handles.text5, 'Visible', 'off');
    set(handles.sdText, 'Visible', 'off');
    set(handles.slider_standard_deviation, 'Visible', 'off');
    set(handles.text10, 'Visible', 'off');
    set(handles.popupmenu_alpha_trimmed_d, 'Visible', 'off');
    set(handles.userTable, 'Visible', 'off');
    filterName = 'average';
case 3
    % Ο χρήστης επέλεξε την 3η τιμή ('Gauss')
    % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Gauss
    set(handles.uipanel_filter_parameters, 'Visible', 'on');
    set(handles.text5, 'Visible', 'on');
    set(handles.sdText, 'Visible', 'on');
    set(handles.slider_standard_deviation, 'Visible', 'on');
    set(handles.text10, 'Visible', 'off');
    set(handles.popupmenu_alpha_trimmed_d, 'Visible', 'off');
    set(handles.userTable, 'Visible', 'off');
    filterName = 'gaussian';
case 4
    % Ο χρήστης επέλεξε την 4η τιμή ('Μεσαίου (Median)')
    % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Median
    set(handles.uipanel_filter_parameters, 'Visible', 'on');
    set(handles.text5, 'Visible', 'off');
    set(handles.sdText, 'Visible', 'off');
    set(handles.slider_standard_deviation, 'Visible', 'off');
    set(handles.text10, 'Visible', 'off');
    set(handles.popupmenu_alpha_trimmed_d, 'Visible', 'off');
    set(handles.userTable, 'Visible', 'off');
    filterName = 'median';
case 5
    % Ο χρήστης επέλεξε την 5η τιμή ('Ελαχίστου (Min)')
    % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Ελαχίστου
    set(handles.uipanel_filter_parameters, 'Visible', 'on');
    set(handles.text5, 'Visible', 'off');
    set(handles.sdText, 'Visible', 'off');
    set(handles.slider_standard_deviation, 'Visible', 'off');
    set(handles.text10, 'Visible', 'off');
    set(handles.popupmenu_alpha_trimmed_d, 'Visible', 'off');
    set(handles.userTable, 'Visible', 'off');
    filterName = 'min';
case 6
    % Ο χρήστης επέλεξε την 6η τιμή ('Μεγίστου (Max)')
    % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Μεγίστου
    set(handles.uipanel_filter_parameters, 'Visible', 'on');
    set(handles.text5, 'Visible', 'off');
    set(handles.sdText, 'Visible', 'off');
    set(handles.slider_standard_deviation, 'Visible', 'off');

```

```

        set(handles.text10,'Visible','off');
        set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
        set(handles.userTable,'Visible','off');
        filterName = 'max';
    case 7
        % Ο χρήστης επέλεξε την 7η τιμή ('Wiener')
        % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Wiener
        set(handles.uipanel_filter_parameters,'Visible','on');
        set(handles.text5,'Visible','off');
        set(handles.sdText,'Visible','off');
        set(handles.slider_standard_deviation,'Visible','off');
        set(handles.text10,'Visible','off');
        set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
        set(handles.userTable,'Visible','off');
        filterName = 'wiener';
    case 8
        % Ο χρήστης επέλεξε την 8η τιμή ('alpha-trimmed')
        % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
alpha-trimmed
        set(handles.uipanel_filter_parameters,'Visible','on');
        set(handles.text5,'Visible','off');
        set(handles.sdText,'Visible','off');
        set(handles.slider_standard_deviation,'Visible','off');
        set(handles.text10,'Visible','on');
        set(handles.popupmenu_alpha_trimmed_d,'Visible','on');
        set(handles.userTable,'Visible','off');
        filterName = 'alpha-trimmed';
    case 9
        % Ο χρήστης επέλεξε την 9η τιμή ('Γεωμετρικού μέσου')
        % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Γεωμετρικού μέσου
        set(handles.uipanel_filter_parameters,'Visible','on');
        set(handles.text5,'Visible','off');
        set(handles.sdText,'Visible','off');
        set(handles.slider_standard_deviation,'Visible','off');
        set(handles.text10,'Visible','off');
        set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
        set(handles.userTable,'Visible','off');
        filterName = 'geometrical-mean';
    case 10
        % Ο χρήστης επέλεξε την 10η τιμή ('Sobel')
        % Ενεργοποίηση του πεδίου με τις παραμέτρους του φίλτρου
Sobel (1ης παραγώγου)
        set(handles.uipanel_filter_parameters,'Visible','on');
        set(handles.text5,'Visible','off');
        set(handles.sdText,'Visible','off');
        set(handles.slider_standard_deviation,'Visible','off');
        set(handles.text10,'Visible','off');
        set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
        set(handles.userTable,'Visible','off');
        filterName = 'sobel';
    case 11
        % Ο χρήστης επέλεξε την 11η τιμή ('laplacian')
        % Ενεργοποίηση του πεδίου με τις παραμέτρους του Laplacian
φίλτρου
        % (2ης παραγώγου)
        set(handles.uipanel_filter_parameters,'Visible','on');
        set(handles.text5,'Visible','off');

```

```

set(handles.sdText,'Visible','off');
set(handles.slider_standard_deviation,'Visible','off');
set(handles.text10,'Visible','off');
set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
set(handles.userTable,'Visible','off');
filterName = 'laplacian';

case 12
% Ο χρήστης επέλεξε την 12η τιμή ('Χειροκίνητο')
% Ενεργοποίηση του πεδίου με τις παραμέτρους του χειροκίνητου
φίλτρου
set(handles.uipanel_filter_parameters,'Visible','on');
set(handles.text5,'Visible','off');
set(handles.sdText,'Visible','off');
set(handles.slider_standard_deviation,'Visible','off');
set(handles.text10,'Visible','off');
set(handles.popupmenu_alpha_trimmed_d,'Visible','off');
set(handles.userTable,'Visible','on');
filterName = 'Χειροκίνητο';

End

```

3.4.1.6 Επιλογή Μεγέθους Μάσκας Φίλτρου

Ο χρήστης πατώντας στο πλήκτρο της λίστας επιλογών που σχετίζεται με το μέγεθος της μάσκας του φίλτρου (popupmenu_filter_mask_size), μπορεί επιλέξει το επιθυμητό μέγεθος μάσκας. Υποστηρίζονται τα μεγέθη 3x3 και 5x5, με το μέγεθος 3x3 να είναι προεπιλεγμένο. Το επιλεγμένο μέγεθος αποθηκεύεται στη μεταβλητή maskSize. Ανάλογα με το επιλεγμένο μέγεθος ενημερώνονται δυναμικά οι διαθέσιμες τιμές του φίλτρου περικεκομμένου μέσου (δεδομένου ότι πρέπει να είναι άρτιες και στο διάστημα από 0 ως (μέγεθος μάσκας - 1)).

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο της λίστας επιλογών popupmenu_filter_mask_size:

```

% --- Executes on selection change in popupmenu_filter_mask_size.
function popupmenu_filter_mask_size_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu_filter_mask_size (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu_filter_mask_size contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu_filter_mask_size

% Όταν αλλάζει το μέγεθος της μάσκας, τότε πρέπει να αλλάζουν
% και οι τιμές της παραμέτρου D του φίλτρου alpha-trimmed
% Λαμβάνω την τιμή του popup-menu με το μέγεθος N της μάσκας του
φίλτρου

```

```

global alpha_trimmed_d;

items = get(hObject, 'String');
index_selected = get(hObject, 'Value');
N = items{index_selected};
switch (N)
    case '3x3'
        maskSize=3;
    case '5x5'
        maskSize=5;
end

i=1;
for k=0:2:(maskSize*maskSize-1)
    D{i} = k;
    i = i + 1;
end
set(handles.popupmenu_alpha_trimmed_d, 'String', D);
set(handles.popupmenu_alpha_trimmed_d, 'Value', 2);
alpha_trimmed_d = 2;

set(handles.userTable, 'Data', zeros(maskSize), 'ColumnWidth', {25},
'ColumnName', 1:maskSize, 'RowName', 1:maskSize);

```

3.4.1.7 Επιλογή Λοιπών Παραμέτρων Φίλτρων

Εκτός από το μέγεθος της μάσκας, ο χρήστης μπορεί να επιλέξει τιμή και για κάποια άλλη παράμετρο του επιλεγμένου φίλτρου (εφόσον το επιλεγμένο φίλτρο διαθέτει επιπλέον παραμέτρους). Πιο συγκεκριμένα:

- Το φίλτρο Gauss έχει ως επιπλέον παράμετρο την τυπική απόκλιση (sd).
- Το φίλτρο περικεκομμένου μέσου (alpha trimmed) έχει ως επιπλέον παράμετρο την τιμή περικοπής (alpha_trimmed_d).

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν έχει επιλεγεί το φίλτρο Gauss και αλλάξει τιμή η μπάρα κύλισης slider_standard_deviation, που αντιστοιχεί στην παράμετρο της τυπικής απόκλισης του φίλτρου:

```

% --- Executes on slider movement.
function slider_standard_deviation_Callback(hObject, eventdata,
handles)
% hObject    handle to slider_standard_deviation (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
of slider

```

```

global sigma;
sigma = get(hObject, 'Value');
set(handles.sdText, 'String', sigma);

```

Επίσης, ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν επιλεγεί κάποια τιμή περικοπής για το φίλτρο alpha trimmed, από τη λίστα επιλογών popupmenu_alpha_trimmed_d:

```

% --- Executes on selection change in popupmenu_alpha_trimmed_d.
function popupmenu_alpha_trimmed_d_Callback(hObject, eventdata,
handles)
% hObject    handle to popupmenu_alpha_trimmed_d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject, 'String')) returns
popupmenu_alpha_trimmed_d contents as cell array
%         contents{get(hObject, 'Value')} returns selected item from
popupmenu_alpha_trimmed_d
contents = cellstr(get(hObject, 'String'));
item = contents{get(hObject, 'Value')};
global alpha_trimmed_d;
alpha_trimmed_d = str2num(item);

```

3.4.1.8 Εκτέλεση Φιλτραρίσματος

Μόλις ο χρήστης πατήσει το πλήκτρο «Εκτέλεση» (btn_filter_image), τότε εκτελείται το φιλτράρισμα της εικόνας και της επιλεγμένης περιοχής. Οι τιμές φωτεινότητας της φιλτραρισμένης εικόνας και της φιλτραρισμένης περιοχής ενδιαφέροντος αποθηκεύονται στις μεταβλητές filteredImage και selected_filtered_region_from_image, αντίστοιχα.

Επίσης, ανάλογα με το επιλεγμένο φίλτρο και το μέγεθος της μάσκας του, δημιουργείται η κατάλληλη μάσκα που αποθηκεύεται στη μεταβλητή hFilter, ενώ κατασκευάζεται και η γειτονιά του επιλεγμένου pixel και αποθηκεύεται στη μεταβλητή dataArea.

Η φιλτραρισμένη εικόνα και η φιλτραρισμένη περιοχή ενδιαφέροντος εμφανίζονται στα αντίστοιχα χειριστήρια του βασικού παραθύρου. Επίσης, ενημερώνεται ο πίνακας με τις τιμές φωτεινότητας της γειτονιάς του επιλεγμένου pixel, ο πίνακας με τα βάρη της μάσκας, εμφανίζεται ένας σύντομος υπολογισμός

της φιλτραρισμένης τιμής για το επιλεγμένο pixel, ενώ ενημερώνεται και ο πίνακας με τις τιμές φωτεινότητων της φιλτραρισμένης περιοχής ενδιαφέροντος, όπου φωτίζεται αυτόματα το φιλτραρισμένο pixel στη φιλτραρισμένη περιοχή ενδιαφέροντος. Τέλος, ενεργοποιείται το πλήκτρο «Επεξήγηση», μέσω του οποίου ο χρήστης μπορεί να δει μία αναλυτική περιγραφή της διαδικασίας του φιλτραρίσματος με το επιλεγμένο φίλτρο.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο εκτέλεσης του φιλτραρίσματος `btn_filter_image`:

```
% --- Executes on button press in btn_filter_image.
function btn_filter_image_Callback(hObject, eventdata, handles)
% hObject    handle to btn_filter_image (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global InitialImage;
global filteredImage;
global maskSize;
global filterName;
global hFilter;
global sigma;
global alpha_trimmed_d;
global noiseVarianceWiener;
global sx;

% Λαμβάνω το μέγεθος της γειτονιάς
contents = cellstr(get(handles.popupmenu_filter_mask_size, 'String'));
item = contents{get(handles.popupmenu_filter_mask_size, 'Value')};
switch (item)
    case '3x3'
        maskSize = 3;
    case '5x5'
        maskSize = 5;
end

hFilter=ones(maskSize);

% Λαμβάνω το όνομα του επιλεγμένου φίλτρου και εκτελώ το φιλτράρισμα
val = get(handles.popupmenu_select_filter, 'Value');
switch (val)
    case 1
        filterName = '';
    case 2
        filterName = 'average';
    case 3
        filterName = 'gaussian';
    case 4
        filterName = 'median';
    case 5
        filterName = 'min';
```

```

case 6
    filterName = 'max';
case 7
    filterName = 'wiener';
case 8
    filterName = 'alpha-trimmed';
case 9
    filterName = 'geometrical-mean';
case 10
    filterName = 'sobel';
case 11
    filterName = 'laplacian';
case 12
    filterName = 'Χειροκίνητο';
end

switch (filterName)
case 'average'
    % Φίλτρο μέσης τιμής (average)
    hFilter = fspecial('average',[maskSize maskSize]);
    filteredImage = imfilter(InitialImage, hFilter);
case 'gaussian'
    % Φίλτρο Gauss
    hFilter = fspecial('gaussian', maskSize, sigma);
    filteredImage = imfilter(InitialImage, hFilter);
case 'median'
    % Φίλτρο μεσαίου (median)
    hFilter = ones(maskSize);
    filteredImage = medfilt2(InitialImage,[maskSize maskSize]);
case 'min'
    % Φίλτρο ελαχίστου (min)
    hFilter = ones(maskSize);
    filteredImage = imerode(InitialImage, ones(maskSize));
case 'max'
    % Φίλτρο μεγίστου (max)
    hFilter = ones(maskSize);
    filteredImage = imdilate(InitialImage, ones(maskSize));
case 'wiener'
    % Φίλτρο Wiener
    hFilter = ones(maskSize);
    [filteredImage noiseVarianceWiener] =
wiener2(InitialImage,[maskSize maskSize]);
    %disp(noiseVarianceWiener);
case 'alpha-trimmed'
    % Φίλτρο περικεκομμένου μέσου (alpha-trimmed)
    hFilter = ones(maskSize);
    filteredImage = alpha_trim( InitialImage, maskSize,
alpha_trimmed_d );
case 'geometrical-mean'
    % Φίλτρο γεωμετρικού μέσου (geometrical-mean)
    hFilter = ones(maskSize);
    filteredImage = g_mean( InitialImage, maskSize );
case 'sobel'
    % Φίλτρο όξυνσης sobel
    [filteredImage sx] = sobel( InitialImage, maskSize );

case 'laplacian'
    % Φίλτρο όξυνσης laplacian

```

```

        if (maskSize==3)
            hFilter = [0 -1 0;-1 5 -1;0 -1 0];
        else
            hFilter = [0 0 -1 0 0;0 -1 -2 -1 0;-1 -2 17 -2 -1;0 -1 -2
-1 0;0 0 -1 0 0];
        end
        filteredImage = imfilter(InitialImage, hFilter,'replicate');

        case 'Χειροκίνητο'
            % Φίλτρο ορισμένο από τον χρήστη
            hFilter = get(handles.userTable,'Data');
            filteredImage = uint8( filter2( hFilter, InitialImage));
        end

% Εμφάνιση φιλτραρισμένης εικόνας και της φιλτραρισμένης επιλεγμένης
περιοχής
axes(handles.axes_filtered_image);
imshow(filteredImage);

global position;
global selected_filtered_region_from_image;

[selected_filtered_region_from_image roi_fixed2] =
selectedRoiIndices(position, filteredImage);
axes(handles.axes_filtered_selected_region);
imshow(selected_filtered_region_from_image);

% Εμφάνιση φιλτραρισμένων τιμών των pixel στον πίνακα
tableFilteredPixelValues
cnames = [];
for i = 0:roi_fixed2(3)-1
    cnames = [cnames roi_fixed2(1)+i];
end
rnames = [];
for i = 0:roi_fixed2(4)-1
    rnames = [rnames roi_fixed2(2)+i];
end

set(handles.tableFilteredPixelValues,'Data',selected_filtered_region_
from_image, 'ColumnWidth',{25}, 'ColumnName',cnames,
'RowName',rnames);
jscroll=findobj(handles.tableFilteredPixelValues);
h=jscroll.getComponent;
viewport=h(1);
a=viewport.getComponent;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
jtable.setRowSelectionAllowed(true);
jtable.changeSelection(5-1,5-1, false, false);

% Επιλογή του κεντρικού pixel της αρχικής επιλεγμένης περιοχής στον
πίνακα tablePixelValues
global selected_region_from_image;

```

```

set(handles.tablePixelValues,'Data',selected_region_from_image,
'ColumnWidth',{25}, 'ColumnName',cnames, 'RowName',rnames);
jscroll=findjobj(handles.tablePixelValues);
h=jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
jtable.setRowSelectionAllowed(true);
jtable.changeSelection(5-1,5-1, false, false);

% Ενεργοποίηση του πλήκτρου Επεξήγηση
set(handles.btn_explain_filtering,'Enable', 'on');

% Ενημέρωση λεκτικού με όνομα/παράμετρους φίλτρου, μάσκας/γειτονιάς
set(handles.selectedFilterNameText,'String',strcat('Φίλτρο:', ' ',
filterName, ' Μέγεθος μάσκας: ', num2str(maskSize),'x',
num2str(maskSize)));

% Τοποθετώ σε πίνακα (neighborhoodPixelsTable) τη γειτονιά του
επιλεγμένου pixel
% Κεντρικό pixel είναι αυτό στη μέση της επιλεγμένης περιοχής της
αρχικής εικόνας
global roi_fixed;
global selectedPixelValue;
row = roi_fixed(2) + 5 - 1;
col = roi_fixed(1) + 5 - 1;
selectedPixelValue = InitialImage(row, col);

% Δημιουργώ τον πίνακα - γειτονιά του επιλεγμένου Pixel
(μεγέθους=maskSize)
global dataArea;
dataArea = [];
dataArea = findNeighborhoodTable(InitialImage, row, col, maskSize);

% Εμφανίζω τη γειτονιά και μαρκάρω το κεντρικό της Pixel
cnames = [roi_fixed(1)+(roi_fixed(3)/2 -1) - ((maskSize+1)/2 -1) :
roi_fixed(1)+(roi_fixed(3)/2-1) + ((maskSize+1)/2 -1)];
rnames = [roi_fixed(2)+(roi_fixed(4)/2 -1) - ((maskSize+1)/2 -1) :
roi_fixed(2)+(roi_fixed(4)/2-1) + ((maskSize+1)/2 -1)];
set(handles.neighborhoodPixelsTable,'data',
dataArea,'ColumnWidth',{25}, 'ColumnName',cnames, 'RowName',rnames);
jscroll=findjobj(handles.neighborhoodPixelsTable);
h=jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
jtable.setRowSelectionAllowed(true);
row = (maskSize+1)/2;
col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false);

```

```

% Εμφανίζω τη μάσκα του φίλτρου
switch (filterName)
    case {'average', 'gaussian', 'geometrical-
mean', 'laplacian', 'Χειροκίνητο'}
        set(handles.maskTable, 'visible', 'on');
        set(handles.text20, 'visible', 'on');
        set(handles.text21, 'visible', 'on');
        set(handles.maskTable, 'data', hFilter, 'ColumnWidth', {25},
'ColumnName', [1:maskSize], 'RowName', [1:maskSize]);
        % Ενημέρωση λεκτικού αποτελέσματος χρήσης του φίλτρου
        res = sum(sum(double(dataArea).*hFilter));
        if (strcmp(filterName, 'geometrical-mean'))
            res = prod(prod(double(dataArea).*hFilter));
            resFinal = res^(1/(maskSize*maskSize));
        end
        if (strcmp(filterName, 'geometrical-mean'))
            set(handles.maskTable, 'visible', 'off');
            set(handles.text20, 'visible', 'off');
            set(handles.text21, 'visible', 'off');
            set(handles.maskingText, 'String', strcat('=>
(', num2str(res), ')^(1/', num2str(maskSize*maskSize), ') =>
', num2str(round(resFinal)))));
        elseif (strcmp(filterName, 'Χειροκίνητο') ||
strcmp(filterName, 'laplacian') )
            filteredValue = round(res);
            if (filteredValue < 0)
                filteredValue = 0;
            end
            if (filteredValue > 255)
                filteredValue = 255;
            end
            set(handles.maskingText, 'String', strcat('=>
', num2str(res), ' => ', num2str(filteredValue), ' Πατήστε Επεξήγηση'));
        else
            set(handles.maskingText, 'String', strcat('=>
', num2str(res), ' => ', num2str(round(res))));
        end

        case {'min', 'max', 'median', 'alpha-trimmed', 'wiener', 'sobel'}
            %set(handles.maskTable, 'data',
ones(maskSize), 'ColumnWidth', {25}, 'ColumnName', [1:maskSize],
'RowName', [1:maskSize]);
            set(handles.maskTable, 'visible', 'off');
            set(handles.text20, 'visible', 'off');
            set(handles.text21, 'visible', 'off');
            set(handles.maskingText, 'String', strcat('=> ',
num2str(selected_filtered_region_from_image(5,5)), ' Πατήστε
Επεξήγηση'));
    end

% Τέλος, ενεργοποιώ και τη δυνατότητα επιλογής κάποιου pixel
set(handles.tablePixelValues, 'Enable', 'on');

```

Στο σημείο αυτό είναι χρήσιμο να παρουσιάσουμε τον κώδικα της συνάρτησης `findNeighborhoodTable()`, που βρίσκεται στο αρχείο `findNeighborhoodTable.m` και υπολογίζει την γειτονιά του επιλεγμένου `pixel` της περιοχής ενδιαφέροντος.

Ο κώδικας του αρχείου `findNeighborhoodTable.m` παρατίθεται παρακάτω:

```
function [ neighborhoodTable ] = findNeighborhoodTable( area,
centerPixelRow, centerPixelCol, sizeToSelect )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

% Θα πρέπει να προσθέσω έναν έλεγχο για το αν το επιλεγμένο pixel
βρίσκεται
% στα όρια της εικόνας, έτσι ώστε να προσθέσω μηδενικά και να
σχηματίσω τη
% ζητούμενη γειτονιά μεγέθους sizeToSelect*sizeToSelect

[rows cols] = size(area);
if ((centerPixelRow>=1) & (centerPixelRow<=rows)) &
((centerPixelCol>=1) & (centerPixelCol<=cols))
areaIsValid = true;
k1=1;
for i=-((sizeToSelect+1)/2-1):((sizeToSelect+1)/2-1)
    k2=1;
    for j=-((sizeToSelect+1)/2-1):((sizeToSelect+1)/2-1)
        if ((centerPixelRow+i>=1) & (centerPixelRow+i<=rows)) &
((centerPixelCol+j>=1) & (centerPixelCol+j<=cols))

neighborhoodTable(k1,k2)=area(centerPixelRow+i,centerPixelCol+j);
        else
            neighborhoodTable(k1,k2) = 0;
            areaIsValid = false;
        end
        k2=k2+1;
    end
    k1=k1+1;
end
if (areaIsValid==true)
    %disp('Valid area indexes');
else
    disp('INVALID area indices! Padded with zeros as necessary!');
end
else
    neighborhoodTable=[];
    disp('ERROR: Selected pixel indices are out of bounds!');
end

end
```

3.4.1.9 Επιλογή Pixel Επίδειξης Φιλτραρίσματος

Ο χρήστης έχει τη δυνατότητα να επιλέξει με το ποντίκι κάποιο άλλο pixel της επιλεγμένης περιοχής ενδιαφέροντος, προκειμένου η επεξήγηση του φιλτραρίσματος να γίνει με βάση αυτό το pixel.

Όταν ο χρήστης επιλέξει κάποιο pixel της επιλεγμένης περιοχής, τότε ενημερώνεται η επιλεγμένη γειτονιά, η σύντομη επεξήγηση υπολογισμού της νέας φιλτραρισμένης τιμής, ενώ στον πίνακα φωτεινοτήτων της φιλτραρισμένης περιοχής ενδιαφέροντος φωτίζεται το αντίστοιχο φιλτραρισμένο pixel.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν επιλεγεί κάποιο pixel του πίνακα `tablePixelValues` της επιλεγμένης περιοχής ενδιαφέροντος:

```
--- Executes when selected cell(s) is changed in tablePixelValues.
function tablePixelValues_CellSelectionCallback(hObject, eventdata,
handles)
% hObject    handle to tablePixelValues (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) currently
selecteds
% handles    structure with handles and user data (see GUIDATA)

% Ο παρακάτω κώδικας εκτελείται μόνο όταν έχει γίνει φιλτράρισμα και
% ο πίνακας των φιλτραρισμένων Pixel δεν είναι άδειος
global selected_filtered_region_from_image;
selection = eventdata.Indices(:,:);
if (length(selected_filtered_region_from_image)>0 &&
numel(selection)~=0)

% Χρειάζομαι τις τιμές των γειτονικών pixels από αυτά του selection
% Η γειτονιά θα είναι μεγέθους όσο η maskSize, πχ 3x3 με κέντρο το
% selected pixel

% PX έστω γειτονιά 3x3
% Κρατώ τη γραμμή και τη στήλη του selection (selected pixel)
row=selection(1);
col=selection(2);

% Επιλογή του κατάλληλου κελιού στον πίνακα με τα φιλτραρισμένα
pixels
jscroll=findobj(handles.tableFilteredPixelValues);
h=jscroll.getComponent;
viewport=h(1);
a=viewport.getComponent;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
```

```

jtable.setRowSelectionAllowed(true);
jtable.changeSelection(row-1,col-1, false, false);

% Τοποθετώ σε πίνακα (neighborhoodPixelsTable) τη γειτονιά του
επιλεγμένου pixel
global InitialImage;
global maskSize;
global roi_fixed;
global selectedPixelValue;
row2 = roi_fixed(2) + row - 1;
col2 = roi_fixed(1) + col - 1;
selectedPixelValue = InitialImage(row2, col2);

% Δημιουργώ τον πίνακα - γειτονιά του επιλεγμένου Pixel
(μεγέθους=maskSize)
global dataArea;
dataArea = [];
dataArea = findNeighborhoodTable(InitialImage, row2, col2, maskSize);

% Εμφανίζω τη γειτονιά και μαρκάρω το κεντρικό της Pixel
global cnames2;
global rnames2;
cnames2 = [col2 - ((maskSize+1)/2 - 1) : col2 + ((maskSize+1)/2 -
1)];
rnames2 = [row2 - ((maskSize+1)/2 - 1) : row2 + ((maskSize+1)/2 -
1)];
set(handles.neighborhoodPixelsTable,'data',
dataArea,'ColumnWidth',{25}, 'ColumnName',cnames2,
'RowName',rnames2);
jscroll=findjobj(handles.neighborhoodPixelsTable);
h=jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
jtable.setSelectionMode(javax.swing.ListSelectionModel.MULTIPLE_INTER
VAL_SELECTION);
jtable.setColumnSelectionAllowed(true);
jtable.setRowSelectionAllowed(true);
row3 = (maskSize+1)/2;
col3 = (maskSize+1)/2;
jtable.changeSelection(row3-1,col3-1, false, false);

global filterName;
global hFilter;
% Εμφανίζω τη μάσκα του φίλτρου
switch (filterName)
    case {'average','gaussian','geometrical-
mean','laplacian','Χειροκίνητο'}
        set(handles.maskTable,'data', hFilter,'ColumnWidth',{25},
'ColumnName',[1:maskSize], 'RowName',[1:maskSize]);
        % Ενημέρωση λεκτικού αποτελέσματος χρήσης του φίλτρου
        res = sum(sum(double(dataArea).*hFilter));
        if (strcmp(filterName,'geometrical-mean'))
            res = prod(prod(double(dataArea).*hFilter));
            resFinal = res^(1/(maskSize*maskSize));
        end
        if (strcmp(filterName,'geometrical-mean'))

```



```

        set(handles.maskingText, 'String', strcat('=>
(' , num2str(res), ')^(1/' , num2str(maskSize*maskSize), ') =>
', num2str(round(resFinal))));
        elseif (strcmp(filterName, 'Χειροκίνητο') ||
strcmp(filterName, 'laplacian'))
            filteredValue = round(res);
            if (filteredValue < 0)
                filteredValue = 0;
            end
            if (filteredValue > 255)
                filteredValue = 255;
            end
            set(handles.maskingText, 'String', strcat('=>
', num2str(res), ' => ', num2str(filteredValue), ' Πατήστε Επεξήγηση'));
        else
            set(handles.maskingText, 'String', strcat('=>
', num2str(res), ' => ', num2str(round(res))));
        end

        case {'min', 'max', 'median', 'alpha-trimmed', 'wiener', 'sobel'}
            set(handles.maskTable, 'data',
ones(maskSize), 'ColumnWidth', {25}, 'ColumnName', [1:maskSize],
'RowName', [1:maskSize]);
            set(handles.maskingText, 'String', strcat('=> ',
num2str(selected_filtered_region_from_image(row,col)), ' Πατήστε
Επεξήγηση'));
    end
end

```

3.4.1.10 Εμφάνιση Ιστογραμμάτων

Πατώντας το πλήκτρο «Ιστογράμματα» (btn_show_histograms) ο χρήστης μπορεί να εμφανίσει και να συγκρίνει τα ιστογράμματα της αρχικής και της φιλτραρισμένης εικόνας, καθώς και της περιοχής ενδιαφέροντος πριν και μετά το φιλτράρισμα. Επιπλέον, για λόγους πληρότητας εμφανίζεται σε ένα ξεχωριστό παράθυρο η αρχική και η τελική εικόνα, προκειμένου ο χρήστης να μπορεί να κάνει ευκολότερα την οπτική σύγκριση.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο εμφάνισης ιστογραμμάτων btn_show_histograms:

```

% --- Executes on button press in btn_show_histograms.
function btn_show_histograms_Callback(hObject, eventdata, handles)
% hObject    handle to btn_show_histograms (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global filteredImage;
global InitialImage;
global selected_region_from_image;
global selected_filtered_region_from_image;

```

```
figure('Name','Σύγκριση αρχικής και φιλτραρισμένης
εικόνας','NumberTitle','off');
subplot(1,2,1), imshow(InitialImage);title('Αρχική εικόνα');
subplot(1,2,2), imshow(filteredImage);title('Φιλτραρισμένη εικόνα');
```

```
figure('Name','Σύγκριση ιστογράμματος αρχικής και φιλτραρισμένης
εικόνας','NumberTitle','off');
subplot(2,2,1), imhist(InitialImage), title('Αρχική εικόνα');xlim([-5
260]);ylim('auto');axh = gca;set(axh,'YGrid','on');
subplot(2,2,2), imhist(selected_region_from_image), title('Επιλεγμένη
περιοχή αρχικής εικόνας');xlim([-5 260]);ylim('auto');axh =
gca;set(axh,'YGrid','on');
subplot(2,2,3), imhist(filteredImage), title('Φιλτραρισμένη
εικόνα');xlim([-5 260]);ylim('auto');axh = gca;set(axh,'YGrid','on');
subplot(2,2,4), imhist(selected_filtered_region_from_image),
title('Φιλτραρισμένη επιλεγμένη περιοχή');xlim([-5
260]);ylim('auto');axh = gca;set(axh,'YGrid','on');
```

3.4.1.11 Επεξήγηση Φιλτραρίσματος

Πατώντας το πλήκτρο «Επεξήγηση» (btn_explain_filtering) ο χρήστης μπορεί να εμφανίσει το παράθυρο της αναλυτικής επεξήγησης της διαδικασίας φιλτραρίσματος με τις παραμέτρους που έχει επιλέξει και για το pixel που έχει επιλέξει στην περιοχή ενδιαφέροντος. Ανάλογα με το επιλεγμένο φίλτρο εμφανίζεται το κατάλληλο παράθυρο.

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν πατηθεί το πλήκτρο επεξήγησης btn_explain_filtering:

```
% --- Executes on button press in btn_explain_filtering.
function btn_explain_filtering_Callback(hObject, eventdata, handles)
% hObject handle to btn_explain_filtering (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global filterName;

switch(filterName)
    case {'min', 'max', 'median'}
        explain_order; %opens the subgui
    case {'average', 'gaussian', 'geometrical-mean', 'Χειροκίνητο'}
        explain_linear; %opens the subgui
    case {'alpha-trimmed'}
        explain_alpha_trimmed; %opens the subgui
    case {'wiener'}
        explain_wiener; %opens the subgui
    case {'sobel'}
        explain_sobel; %opens the subgui
    case {'laplacian'}
        explain_laplacian; %opens the subgui
end
```

3.4.2 Προγραμματισμός των Παραθύρων Επεξήγησης

Για την αναλυτική επεξήγηση (επίδειξη) της διαδικασίας φιλτραρίσματος χρησιμοποιούνται έξι (6) παράθυρα (fig-files) και ο προγραμματισμός τους γίνεται στα ομώνυμα M-files. Πιο συγκεκριμένα, έχουμε τα εξής παράθυρα επεξήγησης:

1. Explain_linear.fig, για φιλτράρισμα με τα φίλτρα:
 - μέσης τιμής (average),
 - gauss,
 - γεωμετρικού μέσου και
 - χειροκίνητο, που ορίζει ο χρήστης τα βάρη της μάσκας του φίλτρου
2. Explain_order.fig, για φιλτράρισμα με τα φίλτρα:
 - min,
 - max και
 - median
3. Explain_alpha_trimmed.fig, για φιλτράρισμα με το φίλτρο περικεκομμένου μέσου (alpha trimmed).
4. Explain_wiener.fig, για φιλτράρισμα με το προσαρμοζόμενο φίλτρο wiener.
5. Explain_Sobel.fig, για φιλτράρισμα με το φίλτρο Sobel (1^{ης} παραγώγου).
6. Explain_laplacian.fig, για φιλτράρισμα με το φίλτρο Laplacian (2^{ης} παραγώγου).

3.4.2.1 Το Παράθυρο Επεξήγησης Βασικών Γραμμικών Φίλτρων

Για την επίδειξη φιλτραρίσματος με τα βασικά γραμμικά φίλτρα χρησιμοποιείται το παράθυρο explain_linear.fig και ο προγραμματισμός του γίνεται στο αρχείο explain_linear.m. Τα φίλτρα που υποστηρίζει είναι τα εξής:

1. μέσης τιμής (average),
2. gauss,
3. γεωμετρικού μέσου και
4. χειροκίνητο, που ο χρήστης μπορεί να ορίσει τα βάρη της μάσκας του φίλτρου.

Υπενθυμίζεται ότι το φίλτρο γεωμετρικού μέσου δεν είναι γραμμικό φίλτρο, ωστόσο, ο προγραμματισμός του εντάχθηκε στο συγκεκριμένο αρχείο επειδή οι πράξεις υπολογισμού του είναι παρόμοιες με αυτές των γραμμικών φίλτρων.

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, ανάλογα με το επιλεγμένο φίλτρο, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν `OpeningFcn` του παραθύρου. Επιπλέον, προκειμένου να γίνουν και να παρουσιαστούν οι πράξεις υπολογισμού που απαιτούνται κάθε φορά και να ενημερωθούν τα κατάλληλα χειριστήρια του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις παρακάτω μεταβλητές, που έχουν οριστεί ως καθολικές:

- `maskSize`
- `filterName`
- `hFilter`
- `selectedPixelValue`
- `dataArea`
- `roi_fixed`
- `sigma`

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο `explain_linear`:

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

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

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
```

```

global hFilter;
global selectedPixelValue;
global dataArea;
global roi_fixed;
global sigma;

filterNameText = filterName;
set(handles.filterNameText, 'String', filterNameText);
set(handles.selectedPixelTable, 'Data', selectedPixelValue);

maskSizeText = strcat(num2str(maskSize), 'x', num2str(maskSize));
set(handles.maskSizeText, 'String', maskSizeText);
switch (filterName)
    case 'average'
        shortDescriptionText = 'Υπολογίζεται ο αριθμητικός μέσος των
τιμών έντασης των pixels της γειτονιάς του επιλεγμένου pixel, ο
οποίος αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
        set(handles.sdText, 'Visible', 'off');
        set(handles.sdValueText, 'Visible', 'off');
    case 'gaussian'
        shortDescriptionText = 'Υπολογίζεται το άθροισμα των
γινομένων μεταξύ των τιμών έντασης των pixels της γειτονιάς του
επιλεγμένου pixel και των αντίστοιχων συντελεστών της μάσκας. Το
τελικό αποτέλεσμα αντικαθιστά την τιμή του επιλεγμένου pixel στη
φιλτραρισμένη εικόνα.';
        set(handles.sdText, 'Visible', 'on');
        set(handles.sdValueText, 'Visible', 'on');
        set(handles.sdValueText, 'String', num2str(sigma));
    case 'geometrical-mean'
        shortDescriptionText = 'Υπολογίζεται ο γεωμετρικός μέσος των
τιμών έντασης των pixels της γειτονιάς του επιλεγμένου pixel, ο
οποίος αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
        set(handles.sdText, 'Visible', 'off');
        set(handles.sdValueText, 'Visible', 'off');
    case 'Χειροκίνητο'
        shortDescriptionText = 'Υπολογίζεται το άθροισμα των
επιμέρους γινομένων των τιμών έντασης των pixels της γειτονιάς του
επιλεγμένου pixel με τις αντίστοιχες τιμές της μάσκας του φίλτρου, το
οποίο αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
        set(handles.sdText, 'Visible', 'off');
        set(handles.sdValueText, 'Visible', 'off');
end

set(handles.shortDescriptionText, 'String', shortDescriptionText);

global cnames2;
global rnames2;
set(handles.neighborhoodTable, 'Data', dataArea, 'ColumnWidth', {25},
'ColumnName', cnames2, 'RowName', rnames2);
jscroll=findjobj(handles.neighborhoodTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;

```

```

col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

set(handles.maskTable,'Data',hFilter,'ColumnWidth',{40},
'ColumnName',[1:maskSize], 'RowName',[1:maskSize]);

% Εκτέλεση πράξεων φιλτραρίσματος
% Ενημέρωση του κειμένου με τις οδηγίες εκτέλεσης του φιλτραρίσματος
% Δημιουργία String επίδειξης Εκτέλεσης πράξεων
msg='';
[data_rows data_cols] = size(dataArea);
switch (filterName)
    case {'average', 'gaussian', 'Χειροκίνητο'}
        res = 0.0;
        for i=1:data_rows
            for j=1:data_cols
                res = res + tofloat(dataArea(i,j))*hFilter(i,j);
            end
        end
        res = res*255;
        msg = createFormattedText(dataArea,hFilter,'+',3,6,5);
    case {'geometrical-mean'}
        res=1.0;
        for i=1:data_rows
            for j=1:data_cols
                res = res*tofloat(dataArea(i,j))*hFilter(i,j);
            end
        end
        res=res ^ (1.0/(data_rows*data_cols));
        res=res*255;
        msg = createFormattedText(dataArea,hFilter,'*',3,6,5);
end
% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);

filteredValue = round(res);
if (filteredValue < 0)
    filteredValue = 0;
end
if (filteredValue > 255)
    filteredValue = 255;
end

set(handles.filteredValueTable,'Data',filteredValue);

switch (filterName)
    case {'average', 'gaussian', 'Χειροκίνητο'}
        msg = strcat('=(', msg, ') = ', num2str(res), ' =>
',num2str(filteredValue));
    case {'geometrical-mean'}
        msg = strcat('=(', msg, ')^(1/',
num2str(maskSize*maskSize),') = ', num2str(res), ' =>
',num2str(round(res)));
end

set(handles.mathsText,'String',msg);

```

Στο σημείο αυτό κρίνεται σκόπιμο να αναφέρουμε τη συνάρτηση `g_mean()`, η οποία χρησιμοποιείται στον κώδικα του βασικού παραθύρου `BasicFiltering.m` της εφαρμογής για τον υπολογισμό της φιλτραρισμένης εικόνας με το φίλτρο γεωμετρικού μέσου.

Ο σχετικός κώδικας βρίσκεται στο αρχείο `g_mean.m` και παρατίθεται παρακάτω:

```
function [ filteredImage ] = g_mean( InitialImage, maskSize )
%GMEAN Summary of this function goes here
% Detailed explanation goes here

[InitialImageScaled revertClass] = tofloat(InitialImage);
filteredImage = exp(imfilter(log(InitialImageScaled),
ones(maskSize))) .^ (1/(maskSize*maskSize)));
filteredImage = revertClass(filteredImage);

end
```

3.4.2.2 Το Παράθυρο Επεξήγησης Μη Γραμμικών Φίλτρων Στατιστικής Διάταξης

Για την επίδειξη φιλτραρίσματος με τα βασικά μη γραμμικά φίλτρα στατιστικής διάταξης χρησιμοποιείται το παράθυρο `explain_order.fig` και ο προγραμματισμός του γίνεται στο αρχείο `explain_order.m`. Τα φίλτρα που υποστηρίζει είναι τα εξής:

1. ελάχιστης τιμής (`min`),
2. μέγιστης τιμής (`max`),
3. ενδιάμεσης τιμής (`median`)

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, ανάλογα με το επιλεγμένο φίλτρο, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν `OpeningFcn` του παραθύρου. Επιπλέον, προκειμένου να γίνουν οι πράξεις υπολογισμού που απαιτούνται κάθε φορά και η ενημέρωση των χειριστηρίων του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις εξής μεταβλητές, που έχουν οριστεί ως καθολικές:

- `maskSize`
- `filterName`
- `selectedPixelValue`

- dataArea
- roi_fixed

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο explain_order:

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

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

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
global selectedPixelValue;
global dataArea;
global roi_fixed;

filterNameText = filterName;
set(handles.filterNameText, 'String', filterNameText);
set(handles.selectedPixelTable, 'Data', selectedPixelValue);

maskSizeText = strcat(num2str(maskSize), 'x', num2str(maskSize));
set(handles.maskSizeText, 'String', maskSizeText);
switch (filterName)
    case 'min'
        shortDescriptionText = 'Οι τιμές των pixels, που ανήκουν στην
γειτονιά του επιλεγμένου pixel, ταξινομούνται και η ελάχιστη τιμή
τους αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
    case 'max'
        shortDescriptionText = 'Οι τιμές των pixels, που ανήκουν στην
γειτονιά του επιλεγμένου pixel, ταξινομούνται και η μέγιστη τιμή τους
αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
    case 'median'
        shortDescriptionText = 'Οι τιμές των pixels, που ανήκουν στην
γειτονιά του επιλεγμένου pixel, ταξινομούνται και η διάμεση τιμή τους
αντικαθιστά την τιμή του επιλεγμένου pixel στη φιλτραρισμένη
εικόνα.';
end

set(handles.shortDescriptionText, 'String', shortDescriptionText);
```



```

set(handles.text15, 'String', filterNameText);

global cnames2;
global rnames2;
set(handles.neighborhoodTable, 'Data', dataArea, 'ColumnWidth', {25},
'ColumnName', cnames2, 'RowName', rnames2);
jscroll=findjobj(handles.neighborhoodTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;
col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

% Αναδιάταξη τιμών γειτονιάς σε μία γραμμή
dataAreaOneLine = [];
[rows cols] = size(dataArea);
for i=1:rows
    for j=1:cols
        dataAreaOneLine = [dataAreaOneLine dataArea(i,j)];
    end
end
% Εύρεση τιμών min, max, median
sortedDataAreaOneLine = sort(dataAreaOneLine);
minOfDataArea = sortedDataAreaOneLine(1);
maxOfDataArea = sortedDataAreaOneLine(end);
medianOfDataArea =
sortedDataAreaOneLine(round(length(dataAreaOneLine)/2));

cnames=[];
for i=1:rows*cols
    cnames=[cnames i];
end
set(handles.sortedNeighborhoodValuesTable, 'Data', sortedDataAreaOneLin
e, 'ColumnWidth', {25}, 'ColumnName', cnames, 'RowName', [1]);

switch (filterName)
    case 'min'
        filteredValue = minOfDataArea;
        selectValueInTable = 1;
    case 'max'
        filteredValue = maxOfDataArea;
        selectValueInTable = rows*cols;
    case 'median'
        filteredValue = medianOfDataArea;
        selectValueInTable = round(length(dataAreaOneLine)/2);
end
jscroll=findjobj(handles.sortedNeighborhoodValuesTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
jtable.changeSelection(1-1,selectValueInTable-1, false, false)

set(handles.selectedValueTable, 'Data', filteredValue);

```

3.4.2.3 Το Παράθυρο Επεξήγησης Φίλτρου Περικεκομμένου Μέσου (Alpha Trimmed)

Για την επίδειξη φιλτραρίσματος με το μη γραμμικό φίλτρο στατιστικής διάταξης `alpha trimmed` χρησιμοποιείται το παράθυρο `explain_alpha_trimmed.fig` και ο προγραμματισμός του γίνεται στο αρχείο `explain_alpha_trimmed.m`.

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν `OpeningFcn` του παραθύρου. Επιπλέον, προκειμένου να γίνουν οι πράξεις υπολογισμού που απαιτούνται και η ενημέρωση των χειριστηρίων του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις εξής μεταβλητές, που έχουν οριστεί ως καθολικές:

- `maskSize`
- `filterName`
- `selectedPixelValue`
- `dataArea`
- `roi_fixed`

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο `explain_alpha_trimmed`:

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

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

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
```

```

global selectedPixelValue;
global dataArea;
global roi_fixed;

filterNameText = filterName;
set(handles.filterNameText,'String',filterNameText);
set(handles.selectedPixelTable,'Data',selectedPixelValue);

maskSizeText = strcat(num2str(maskSize),'x',num2str(maskSize));
set(handles.maskSizeText,'String',maskSizeText);
shortDescriptionText = 'Οι τιμές των pixels, που ανήκουν στην
γειτονιά του επιλεγμένου pixel, ταξινομούνται και αφού αφαιρεθούν οι
D/2 μικρότερες και οι D/2 μεγαλύτερες τιμές, υπολογίζεται ο μέσος
όρος των υπολοίπων τιμών, ο οποίος και αντικαθιστά την τιμή του
επιλεγμένου pixel στη φιλτραρισμένη εικόνα.';
set(handles.shortDescriptionText,'String',shortDescriptionText);

global cnames2;
global rnames2;
set(handles.neighborhoodTable,'Data',dataArea,'ColumnWidth',{25},
'ColumnName',cnames2,'RowName',rnames2);
jscroll=findjobj(handles.neighborhoodTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;
col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

% Αναδιάταξη τιμών γειτονιάς σε μία γραμμή
dataAreaOneLine = [];
[rows cols] = size(dataArea);
for i=1:rows
    for j=1:cols
        dataAreaOneLine = [dataAreaOneLine dataArea(i,j)];
    end
end
% Ταξινόμηση τιμών
sortedDataAreaOneLine = sort(dataAreaOneLine);

cnames=[];
for i=1:rows*cols
    cnames=[cnames i];
end
set(handles.sortedNeighborhoodValuesTable,'Data',sortedDataAreaOneLin
e,'ColumnWidth',{25},'ColumnName',cnames,'RowName',[1]);

global alpha_trimmed_d;
alphaTrimmedValuesForAveraging =
sortedDataAreaOneLine(alpha_trimmed_d/2+1:end-(alpha_trimmed_d/2));

set(handles.dText,'String', num2str(alpha_trimmed_d));

```

```

set(handles.step3Text,'String',strcat('Αφαίρεση των D/2 (',
num2str(alpha_trimmed_d/2),' ) μικρότερων και των D/2 (',
num2str(alpha_trimmed_d/2),' ) μεγαλύτερων τιμών'));
set(handles.alphaTrimmedValuesTable,'Data',
alphaTrimmedValuesForAveraging, 'ColumnWidth',{25},
'ColumnName',[alpha_trimmed_d/2+1:length(sortedDataAreaOneLine)-
(alpha_trimmed_d/2)],'RowName',[1]);

% Δημιουργία String επίδειξης Εκτέλεσης πράξεων
msg='';
res=0.0;
for i=1:length(alphaTrimmedValuesForAveraging)
    res = res + tofloat(alphaTrimmedValuesForAveraging(i));
    msg = strcat(' ',msg,
num2str(alphaTrimmedValuesForAveraging(i)), ' + ');
end

res = res / length(alphaTrimmedValuesForAveraging) * 255;

% % % Αφαίρω τους τελευταίους χαρακτήρες ' + '
msg = msg(1:end-1);
msg = strcat('=( ',msg, ')/',
num2str(length(alphaTrimmedValuesForAveraging)), ' = ', num2str(res),
' => ',num2str(round(res)));
set(handles.mathsText,'String',msg);
set(handles.filteredValueTable,'Data',round(res));

```

Επιπλέον, ο υπολογισμός του φιλτραρίσματος με το φίλτρο περικεκομμένου μέσου χρησιμοποιεί τη συνάρτηση `alpha_trim()`, της οποίας ο κώδικας βρίσκεται στο αρχείο `alpha_trim.m` και παρατίθεται παρακάτω:

```

function [ filteredImage ] = alpha_trim( InitialImage, maskSize,
alpha_trimmed_d )
%ALPHA_TRIM Summary of this function goes here
% Detailed explanation goes here

[Initial_Image_Scaled, revertClass] = tofloat(InitialImage);
f = imfilter(Initial_Image_Scaled, ones(maskSize));
for k = 1:(alpha_trimmed_d/2)
    f = f - ordfilt2(Initial_Image_Scaled, k, ones(maskSize));
end
for k = (maskSize*maskSize - (alpha_trimmed_d/2) +
1):(maskSize*maskSize)
    f = f - ordfilt2(Initial_Image_Scaled, k, ones(maskSize));
end
filteredImage = f / (maskSize*maskSize - alpha_trimmed_d);
filteredImage = revertClass(filteredImage);

end

```

3.4.2.4 Το Παράθυρο Επεξήγησης Φίλτρου Wiener

Το φίλτρο Wiener είναι ένα προσαρμοζόμενο φίλτρο μείωσης τοπικού θορύβου, που βασίζεται σε στατιστικά μέτρα της εικόνας και της χρησιμοποιούμενης γειτονιάς καθενός pixel της εικόνας.

Για την επίδειξη φιλτραρίσματος με το φίλτρο wiener χρησιμοποιείται το παράθυρο explain_wiener.fig και ο προγραμματισμός του γίνεται στο αρχείο explain_wiener.m.

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν OpeningFcn του παραθύρου. Επιπλέον, προκειμένου να γίνουν οι πράξεις υπολογισμού που απαιτούνται και η ενημέρωση των χειριστηρίων του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις εξής μεταβλητές, που έχουν οριστεί ως καθολικές:

- maskSize
- filterName
- hFilter
- selectedPixelValue
- dataArea
- roi_fixed
- noiseVarianceWiener

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο explain_wiener:

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

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

```

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
global hFilter;
global selectedPixelValue;
global dataArea;
global roi_fixed;
global noiseVarianceWiener;

filterNameText = filterName;
set(handles.filterNameText, 'String', filterNameText);
set(handles.selectedPixelTable, 'Data', selectedPixelValue);

maskSizeText = strcat(num2str(maskSize), 'x', num2str(maskSize));
set(handles.maskSizeText, 'String', maskSizeText);
shortDescriptionText = 'Χρησιμοποιείται μία προσαρμοζόμενη μέθοδος
Wiener, που βασίζεται σε στατιστικά μέτρα της γειτονιάς του
επιλεγμένου pixel, όπως ο τοπικός μέσος όρος και η τοπική διακύμανση,
καθώς και η διακύμανση του θορύβου που αλλοιώνει την αρχική εικόνα.';
set(handles.noiseVarianceValueText, 'String', num2str(noiseVarianceWiener));
set(handles.shortDescriptionText, 'String', shortDescriptionText);

global cnames2;
global rnames2;
set(handles.neighborhoodTable, 'Data', dataArea, 'ColumnWidth', {25},
'ColumnName', cnames2, 'RowName', rnames2);
jscroll=findjobj(handles.neighborhoodTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;
col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

% Εκτέλεση πράξεων φιλτραρίσματος
% Ενημέρωση του κειμένου με τις οδηγίες εκτέλεσης του φιλτραρίσματος
% Δημιουργία String επίδειξης πράξεων υπολογισμού τοπικού μέσου όρου
msg='';
[data_rows data_cols] = size(dataArea);
localMean = 0.0;
for i=1:data_rows
    for j=1:data_cols
        localMean = localMean + tofloat(dataArea(i,j));
        msg = strcat(msg,num2str(dataArea(i,j)), '+');
    end
end
localMean = localMean / (maskSize*maskSize);
%localMean = localMean*255;
%msg = createFormattedText(dataArea,hFilter, '+', 3,6,5);

```

```

% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);
msg = strcat('=( ', msg, ')/( ', num2str(maskSize*maskSize), ') = ',
num2str(localMean*255));
set(handles.localMeanText, 'String', msg);

% Δημιουργία String επίδειξης πράξεων υπολογισμού τοπικής διακύμανσης
msg='';
localVariance = 0.0;
for i=1:data_rows
    for j=1:data_cols
        localVariance = localVariance + (tofloat(dataArea(i,j))^2 -
(localMean)^2);
        msg = strcat(msg, '(' , num2str(dataArea(i,j)), '^2 -
', num2str(localMean*255), '^2) + ');
    end
end
localVariance = localVariance / (maskSize*maskSize);
%localVariance = localVariance*255;

% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);
msg = strcat('=( ', msg, ')/( ', num2str(maskSize*maskSize), ') = ',
num2str(localVariance));
set(handles.localVarianceText, 'String', msg);

set(handles.noiseVarianceValue2Text, 'String',
num2str(noiseVarianceWiener));

% Υπολογισμός φιλτραρισμένης τιμής Pixel
res = localMean + (max(0, localVariance -
noiseVarianceWiener)/max(localVariance, noiseVarianceWiener)) *
(tofloat(selectedPixelValue) - localMean);
res = res * 255;
filteredValue = round(res);
% Δημιουργία String επίδειξης πράξεων υπολογισμού φιλτραρισμένης
τιμής Pixel
msg='';
msg = strcat(num2str(localMean*255), '+max(0
', num2str(localVariance), '-
', num2str(noiseVarianceWiener), ')/max(', num2str(localVariance), ', ', nu
m2str(noiseVarianceWiener), ') * ( ', num2str(selectedPixelValue), '-
', num2str(localMean*255), ')');
msg = strcat('=( ', msg, ') = ', num2str(res), ' =>
', num2str(filteredValue));
set(handles.mathsText, 'String', msg);
set(handles.filteredValueTable, 'Data', filteredValue);

```

3.4.2.5 Το Παράθυρο Επεξήγησης Φίλτρου Sobel (1^{ης} Παραγώγου)

Το φίλτρο Sobel χρησιμοποιεί δύο μάσκες για τον υπολογισμό του μέτρου της κλίσης στο επιλεγμένο pixel. Η μία μάσκα αφορά την οριζόντια και η άλλη μάσκα την κατακόρυφη κατεύθυνση. Στη συνέχεια, το μέτρο της κλίσης στο επιλεγμένο pixel υπολογίζεται προσεγγιστικά ως το άθροισμα των μέτρων των κλίσεων στις δύο

κατευθύνσεις. Για την επίδειξη φιλτραρίσματος με το φίλτρο Sobel (1^{ης} παραγώγου) χρησιμοποιείται το παράθυρο `explain_sobel.fig` και ο προγραμματισμός του γίνεται στο αρχείο `explain_sobel.m`.

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν `OpeningFcn` του παραθύρου. Επιπλέον, προκειμένου να γίνουν οι πράξεις υπολογισμού που απαιτούνται και η ενημέρωση των χειριστηρίων του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις εξής μεταβλητές, που έχουν οριστεί ως καθολικές:

- `maskSize`
- `filterName`
- `selectedPixelValue`
- `dataArea`
- `roi_fixed`
- `sx` (είναι η μάσκα του φίλτρου Sobel στην οριζόντια κατεύθυνση)

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο `explain_sobel`:

```
% --- Executes just before explain_sobel is made visible.
function explain_sobel_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
global selectedPixelValue;
global dataArea;
global roi_fixed;
```



```

global sx;
sy=sx';

filterNameText = filterName;
set(handles.filterNameText,'String',filterNameText);
set(handles.selectedPixelTable,'Data',selectedPixelValue);

maskSizeText = strcat(num2str(maskSize),'x',num2str(maskSize));
set(handles.maskSizeText,'String',maskSizeText);
shortDescriptionText = 'Υπολογίζεται το μέτρο M του διανύσματος
κλίσης στο επιλεγμένο pixel, υπολογίζοντας το άθροισμα των μέτρων της
πρώτης παραγώγου στην οριζόντια διεύθυνση Gx και στην κατακόρυφη
διδεύθυνση Gy.';
set(handles.shortDescriptionText,'String',shortDescriptionText);

global cnames2;
global rnames2;
set(handles.neighborhoodTable1,'Data',dataArea,'ColumnWidth',{25},
'ColumnName',cnames2,'RowName',rnames2);
jscroll=findjobj(handles.neighborhoodTable1);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;
col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

set(handles.SobelXTable,'Data',sx,'ColumnWidth',{25},
'ColumnName',cnames2,'RowName',rnames2);
% Εκτέλεση πράξεων φιλτραρίσματος
% Ενημέρωση του κειμένου με τις οδηγίες εκτέλεσης του φιλτραρίσματος
% Δημιουργία String επίδειξης πράξεων υπολογισμού της Gx
msg='';
[data_rows data_cols] = size(dataArea);
res = 0.0;
for i=1:data_rows
    for j=1:data_cols
        res = res + tofloat(dataArea(i,j))*sx(i,j);
    end
end
Gx = round(res*255);
msg = createFormattedText(dataArea,sx,'+',3,6,5);

% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);
msg = strcat('=( ', msg, ')= ', num2str(Gx));
set(handles.GxText,'String',msg);

% Δημιουργία String επίδειξης πράξεων υπολογισμού της Gy
set(handles.neighborhoodTable2,'Data',dataArea,'ColumnWidth',{25},
'ColumnName',cnames2,'RowName',rnames2);
jscroll=findjobj(handles.neighborhoodTable2);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;

```

```

col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

set(handles.SobelYTable, 'Data', sy, 'ColumnWidth', {25},
'ColumnName', cnames2, 'RowName', rnames2);
% Εκτέλεση πράξεων φιλτραρίσματος
% Ενημέρωση του κειμένου με τις οδηγίες εκτέλεσης του φιλτραρίσματος
% Δημιουργία String επίδειξης πράξεων υπολογισμού της Gy
msg='';
[data_rows data_cols] = size(dataArea);
res = 0.0;
for i=1:data_rows
    for j=1:data_cols
        res = res + tofloat(dataArea(i,j))*sy(i,j);
    end
end
Gy = round(res*255);
msg = createFormattedText(dataArea,sy, '+', 3,6,5);

% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);
msg = strcat('=( ', msg, ') = ', num2str(Gy));
set(handles.GyText, 'String', msg);

% Υπολογισμός φιλτραρισμένης τιμής Pixel
res = abs(Gx)+abs(Gy);
filteredValue = res;
if (filteredValue<0)
    filteredValue = 0;
end
if (filteredValue > 255)
    filteredValue = 255;
end
% Δημιουργία String επίδειξης πράξεων υπολογισμού φιλτραρισμένης
τιμής Pixel
msg='';
msg = strcat('|', num2str(Gx), '| + |', num2str(Gy), '| =
', num2str(res), '| => ', num2str(filteredValue));
set(handles.MText, 'String', msg);
set(handles.filteredValueTable, 'Data', filteredValue);

```

Στο σημείο αυτό κρίνεται σκόπιμο να αναφέρουμε τη συνάρτηση `sobel()`, η οποία χρησιμοποιείται στον κώδικα του βασικού παραθύρου `BasicFiltering.m` της εφαρμογής για τον υπολογισμό των δύο (2) μασκών του φίλτρου και για το φιλτράρισμα της εικόνας.

Ο σχετικός κώδικας βρίσκεται στο αρχείο `sobel.m` και παρατίθεται παρακάτω:

```

function [ filteredImage, maskX, maskY ] = sobel( initialImage,
maskSize )
%UNTITLED Summary of this function goes here

```

```

% Detailed explanation goes here
if (maskSize==3)
    maskX = -fspecial('sobel');
elseif (maskSize==5)
    maskX = [2 2 4 2 2;1 1 2 1 1;0 0 0 0 0;-1 -1 -2 -1 -1;-2 -2 -4 -2
-2];
end
maskY = maskX';
[initialImageScaled revertClass] = tofloat(initialImage);

Gx = imfilter(initialImageScaled, maskX, 'replicate');
Gy = imfilter(initialImageScaled, maskY, 'replicate');
%Gx = conv2(initialImage, maskX, 'replicate');
%Gy = conv2(initialImage, maskY, 'replicate');
M = abs(Gx) + abs(Gy);
filteredImage = M;
filteredImage = revertClass(filteredImage);
end

```

3.4.2.6 Το Παράθυρο Επεξήγησης Φίλτρου Laplace (2^{ης} Παραγώγου)

Για την επίδειξη φιλτραρίσματος με το φίλτρο Laplace (2^{ης} παραγώγου) χρησιμοποιείται το παράθυρο `explain_laplacian.fig` και ο προγραμματισμός του γίνεται στο αρχείο `explain_laplacian.m`. Η επεξήγηση περιλαμβάνει τον υπολογισμό της δεύτερης παραγώγου για το επιλεγμένο `pixel` της αρχικής εικόνας και την πρόσθεσή του στην αρχική του τιμή. Η διαδικασία αυτή γίνεται προκειμένου να οξυνθούν οι λεπτομέρειες στην αρχική εικόνα.

Με το άνοιγμα του παραθύρου εκτελείται αυτόματα η διαδικασία του φιλτραρίσματος, οπότε όλος ο απαιτούμενος κώδικας είναι γραμμένος στο συμβάν `OpeningFcn` του παραθύρου. Επιπλέον, προκειμένου να γίνουν οι πράξεις υπολογισμού που απαιτούνται και η ενημέρωση των χειριστηρίων του παραθύρου, είναι απαραίτητη η πρόσβαση του παραθύρου στις εξής μεταβλητές, που έχουν οριστεί ως καθολικές:

- `maskSize`
- `filterName`
- `selectedPixelValue`
- `dataArea`
- `roi_fixed`

Ακολουθεί ο σχετικός κώδικας Matlab, που εκτελείται όταν φορτώνεται το παράθυρο explain_laplacian:

```
% --- Executes just before explain_laplacian is made visible.
function explain_laplacian_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%           command line (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

global maskSize;
global filterName;
global selectedPixelValue;
global dataArea;
global roi_fixed;

filterNameText = filterName;
set(handles.filterNameText, 'String', filterNameText);
set(handles.selectedPixelTable, 'Data', selectedPixelValue);

maskSizeText = strcat(num2str(maskSize), 'x', num2str(maskSize));
set(handles.maskSizeText, 'String', maskSizeText);

shortDescriptionText = 'Υπολογίζεται η δεύτερη παράγωγος (Λαπλασιανή)
της αρχικής εικόνας και στη συνέχεια προστίθεται στην αρχική εικόνα,
προκειμένου να προκαλέσουμε όξυνση των λεπτομερειών.';
set(handles.shortDescriptionText, 'String', shortDescriptionText);

% Κατασκευή της μάσκας Laplace με βάση τις διαστάσεις του φίλτρου
if (maskSize==3)
    hFilter = [0 1 0;1 -4 1;0 1 0];
else
    hFilter = [0 0 1 0 0;0 1 2 1 0;1 2 -16 2 1;0 1 2 1 0;0 0 1 0 0];
end

global cnames2;
global rnames2;
set(handles.neighborhoodTable, 'Data', dataArea, 'ColumnWidth', {25},
'ColumnName', cnames2, 'RowName', rnames2);
jscroll=findjobj(handles.neighborhoodTable);
h = jscroll.getComponents;
viewport=h(1);
a=viewport.getComponents;
jtable=a(1);
row = (maskSize+1)/2;
```

```

col = (maskSize+1)/2;
jtable.changeSelection(row-1,col-1, false, false)

set(handles.maskTable, 'Data', hFilter, 'ColumnWidth', {40},
'ColumnName', [1:maskSize], 'RowName', [1:maskSize]);

% Εκτέλεση πράξεων φιλτραρίσματος
% Ενημέρωση του κειμένου με τις οδηγίες εκτέλεσης του φιλτραρίσματος
% Δημιουργία String επίδειξης Εκτέλεσης πράξεων
msg='';
[data_rows data_cols] = size(dataArea);
res = 0.0;
for i=1:data_rows
    for j=1:data_cols
        res = res + tofloat(dataArea(i,j))*hFilter(i,j);
    end
end
res = res*255;
msg = createFormattedText(dataArea,hFilter,'+',3,6,5);

% Αφαιρώ τους επιπλέον χαρακτήρες '*' ή '+'
msg = msg(1:end-1);

filteredValue = round(res);
msg = strcat('=', msg, ') = ', num2str(filteredValue));
set(handles.mathsText, 'String', msg);

set(handles.selectedPixelTable2, 'Data', selectedPixelValue);
set(handles.filteredValueText, 'String', strcat('(',
num2str(filteredValue), ')'));

res = double(selectedPixelValue) - double(filteredValue);
finalValue = res;
if (finalValue < 0)
    finalValue = 0;
end
if (finalValue > 255)
    finalValue = 255;
end
set(handles.finalValueText, 'String', strcat(num2str(res), ' =>
', num2str(finalValue)));
set(handles.filteredValueTable, 'Data', finalValue);

```

3.5 Βασικό Σενάριο Χρήσης Εφαρμογής

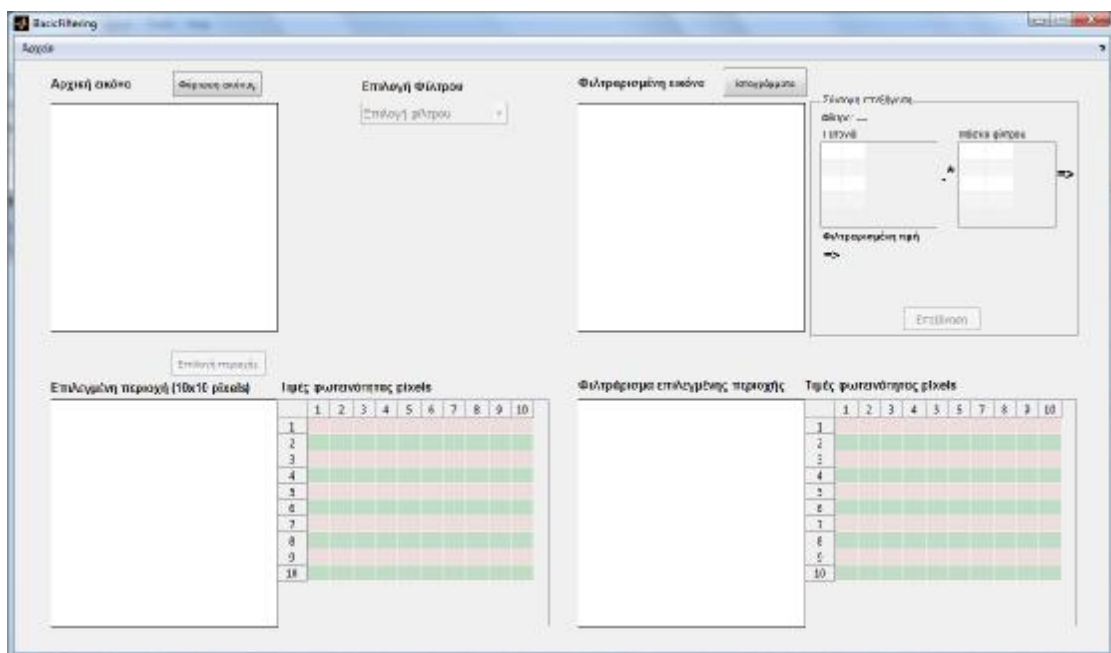
Στην υποενότητα αυτή παρουσιάζεται το βασικό σενάριο χρήσης της εφαρμογής, το οποίο συνοψίζεται στα εξής βήματα:

1. Φόρτωση ασπρόμαυρης εικόνας (αποχρώσεων του γκρι)
2. Επιλογή περιοχής της εικόνας για επίδειξη φιλτραρίσματος
3. Επιλογή φίλτρου

4. Επιλογή παραμέτρων φίλτρου, όπως μέγεθος μάσκας κλπ.
5. Εκτέλεση φιλτραρίσματος
6. Εμφάνιση σύγκρισης ιστογραμμάτων αρχικής και φιλτραρισμένης εικόνας και της επιλεγμένης περιοχής
7. Επεξήγηση διαδικασίας φιλτραρίσματος με το επιλεγμένο φίλτρο
8. Αποθήκευση της φιλτραρισμένης εικόνας

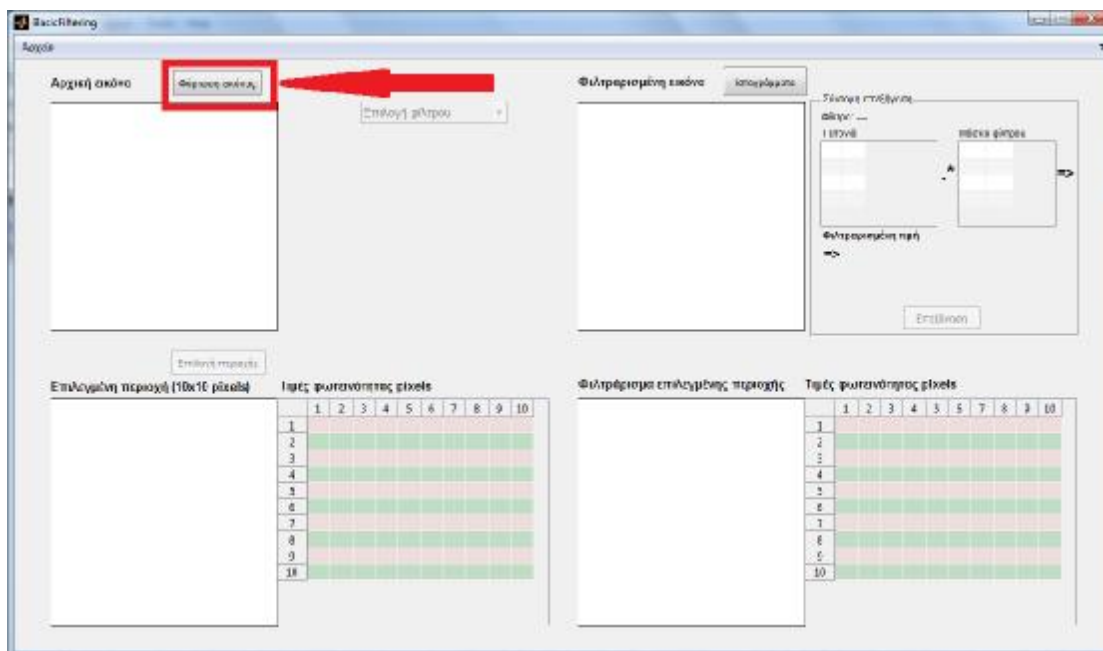
Ακολουθεί η αναλυτική περιγραφή του καθενός βήματος.

Έχοντας εκτελέσει την εφαρμογή μέσα από το περιβάλλον GUIDE του Matlab, ο χρήστης έχει πρόσβαση στην αρχική οθόνη της εφαρμογής. Στην επόμενη εικόνα (Εικόνα 37) φαίνεται η αρχική οθόνη της εφαρμογής.

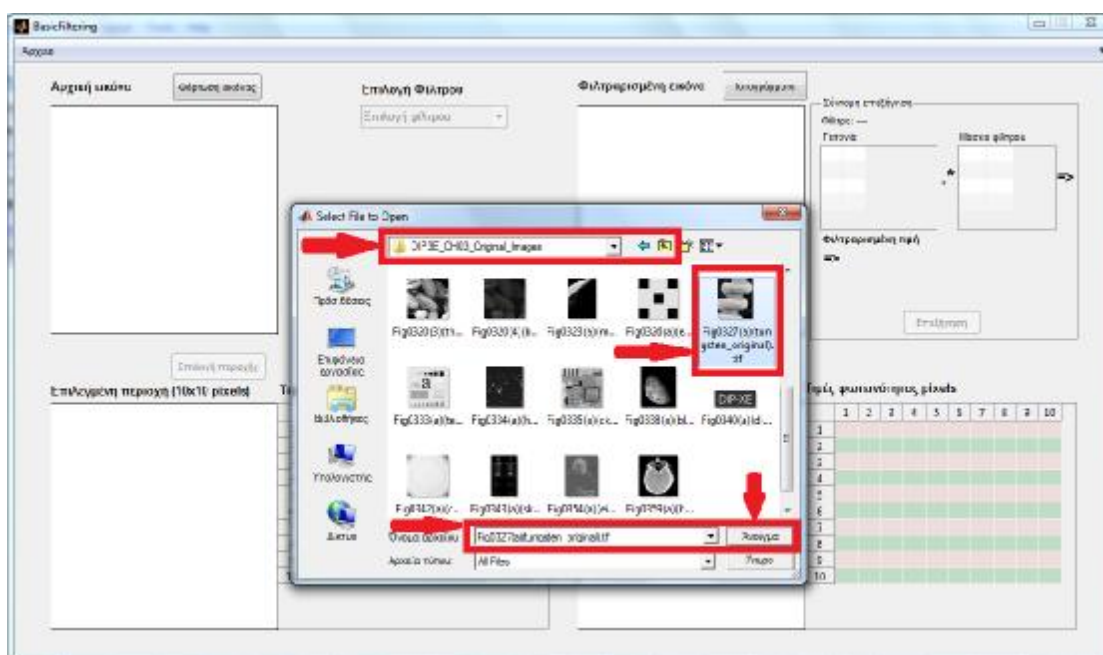


Εικόνα 37: Αρχική οθόνη εφαρμογής

Ακολούθως, πατώντας το πλήκτρο «Φόρτωση εικόνας» εμφανίζεται ένα παράθυρο διαλόγου, όπου ο χρήστης μπορεί να πλοηγηθεί στους φακέλους του, να επιλέξει την εικόνα που επιθυμεί και πατώντας το πλήκτρο «Άνοιγμα» να φορτώσει την επιλεγμένη εικόνα στην εφαρμογή, προκειμένου να την επεξεργαστεί. Στις ακόλουθες εικόνες (Εικόνα 38 και Εικόνα 39) φαίνεται η διαδικασία αυτή.

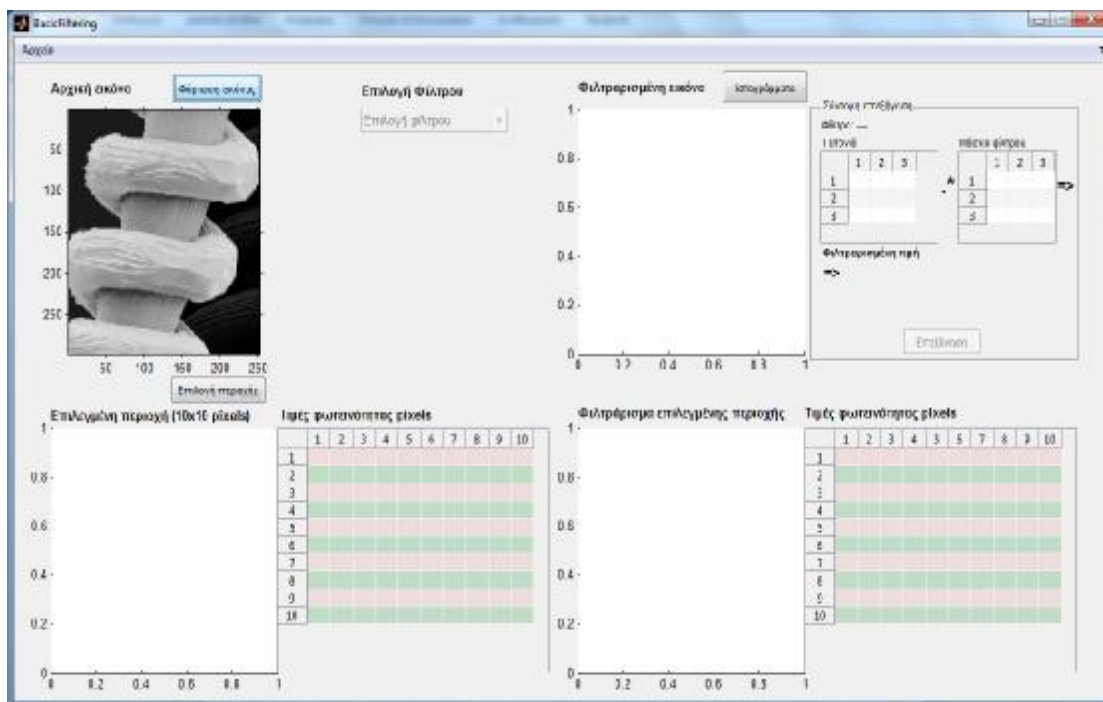


Εικόνα 38: Πλήκτρο «Φόρτωση εικόνας»



Εικόνα 39: Παράθυρο επιλογής και φόρτωσης εικόνας

Μόλις ο χρήστης επιλέξει το πλήκτρο «Άνοιγμα», τότε φορτώνεται η εικόνα και ενημερώνεται η αρχική οθόνη της εφαρμογής εμφανίζοντας την επιλεγμένη εικόνα, όπως φαίνεται στην Εικόνα 40.



Εικόνα 40: Ενημέρωση αρχικής οθόνης εφαρμογής μετά την φόρτωση της εικόνας

Στη συνέχεια, ο χρήστης πρέπει να πατήσει το πλήκτρο «Επιλογή περιοχής» προκειμένου να επιλέξει μία περιοχή ενδιαφέροντος της αρχικής εικόνας, η οποία θα χρησιμοποιηθεί ως αναφορά για την επίδειξη της διαδικασίας του φιλτραρίσματος (Εικόνα 41).

Η εφαρμογή αποκρίνεται εμφανίζοντας την εικόνα σε ένα νέο παράθυρο, στο οποίο ο χρήστης χρησιμοποιώντας το ποντίκι μπορεί να επιλέξει την περιοχή που τον ενδιαφέρει (Εικόνα 42). Επίσης, εφόσον το επιθυμεί ο χρήστης, μπορεί να εστιάσει σε κάποια συγκεκριμένη περιοχή της εικόνας, χρησιμοποιώντας τα χειριστήρια Zoom του παραθύρου.

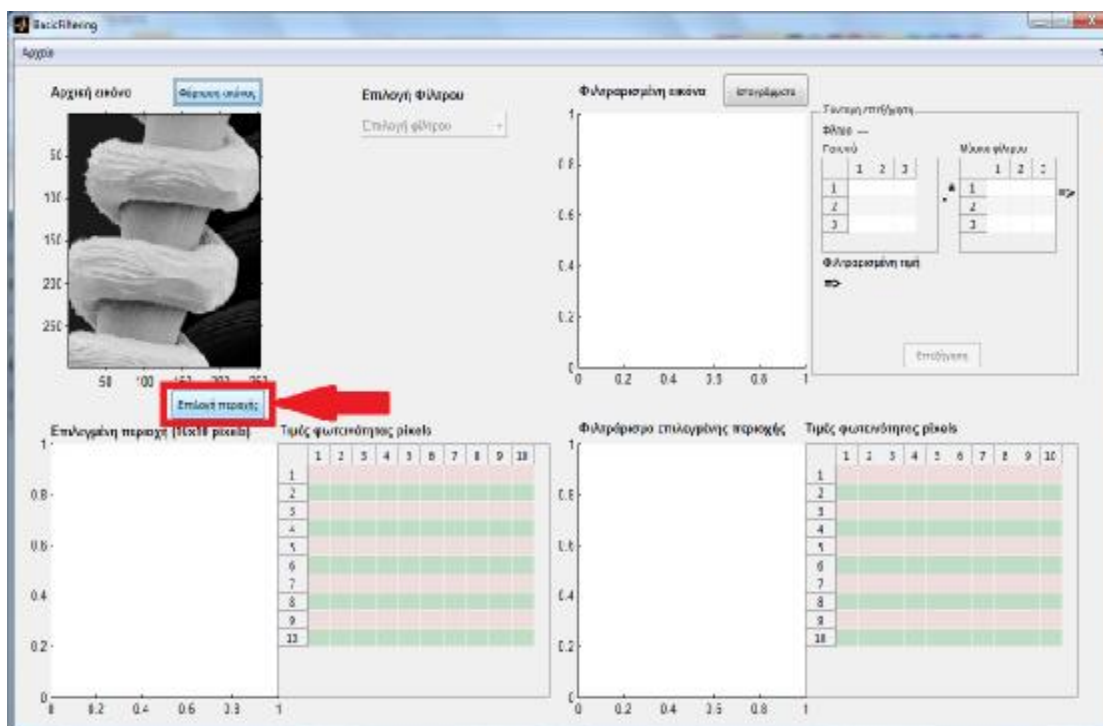
Η επιλογή της περιοχής ενδιαφέροντος γίνεται κάνοντας αριστερό κλικ σε κάποιο σημείο και σέρνοντας το ποντίκι (έχοντας πατημένο το αριστερό κλικ) μέχρι να δημιουργηθεί η περιοχή ενδιαφέροντος. Για να επιλεγεί οριστικά η περιοχή, ο χρήστης πρέπει να κάνει διπλό αριστερό κλικ πάνω στην επιλεγμένη περιοχή.

Δεδομένου ότι η εφαρμογή για λόγους χώρου, μπορεί να εμφανίσει περιοχές ενδιαφέροντος με μέγιστο μέγεθος 10x10 ριξελς, αν ο χρήστης επιλέξει μεγαλύτερη

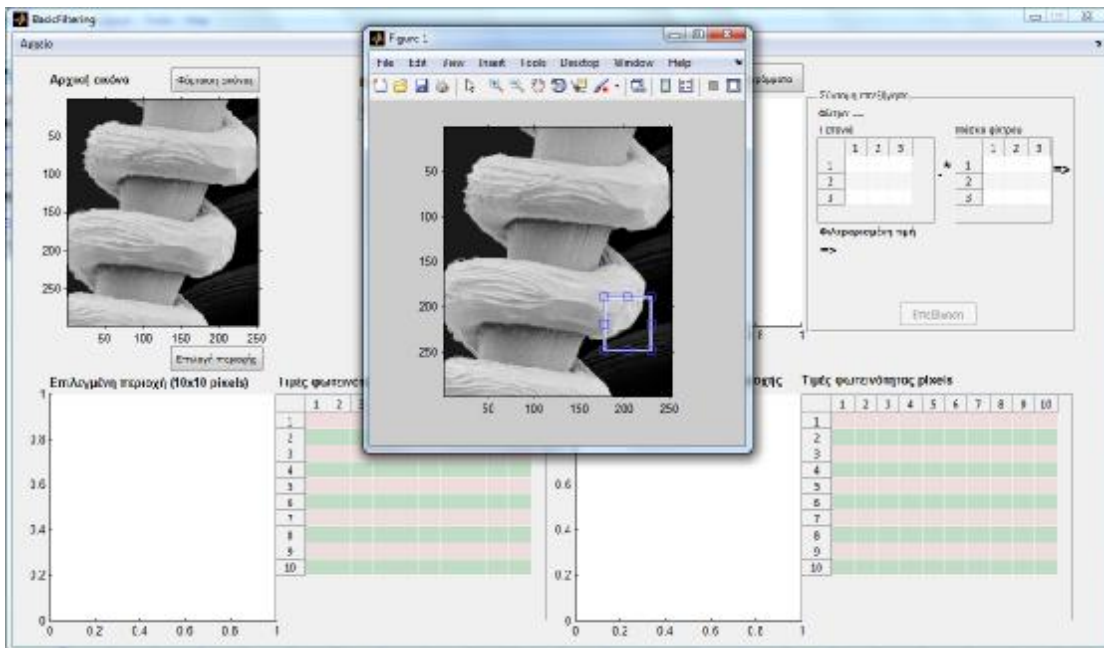
περιοχή, τότε αυτομάτως η περιοχή θα προσαρμοστεί κατάλληλα στο μέγεθος των 10x10 pixels.

Μετά την επιλογή της περιοχής ενδιαφέροντος, ενημερώνεται η αρχική οθόνη της εφαρμογής, εμφανίζοντας τις τιμές έντασης φωτεινότητας των pixels της επιλεγμένης περιοχής, ενώ ταυτόχρονα επιλέγεται (φωτίζεται) η τιμή του κεντρικού pixel της περιοχής ενδιαφέροντος. Επίσης, στην επάνω και στην αριστερή πλευρά του πίνακα με τις τιμές φωτεινότητας των pixels της επιλεγμένης περιοχής φαίνεται η αρίθμηση της γραμμής και της στήλης των pixels (Εικόνα 43).

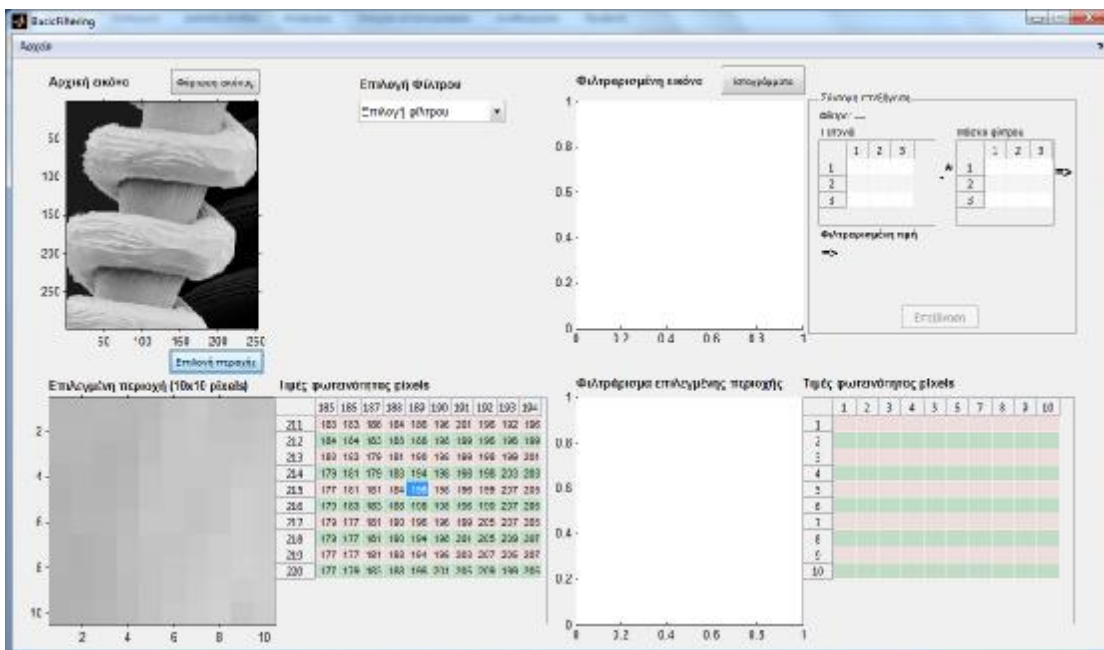
Επισημαίνεται ότι εφόσον έχει φορτωθεί μία εικόνα στην εφαρμογή, ο χρήστης μπορεί ανά πάσα στιγμή να επιλέξει μία διαφορετική περιοχή ενδιαφέροντος πατώντας ξανά το πλήκτρο «Επιλογή περιοχής». Όλα τα προηγούμενα φαίνονται στις παρακάτω εικόνες (Εικόνα 41 ως Εικόνα 43).



Εικόνα 41: Πλήκτρο «Επιλογή περιοχής»



Εικόνα 42: Παράθυρο επιλογής της περιοχής ενδιαφέροντος της αρχικής εικόνας

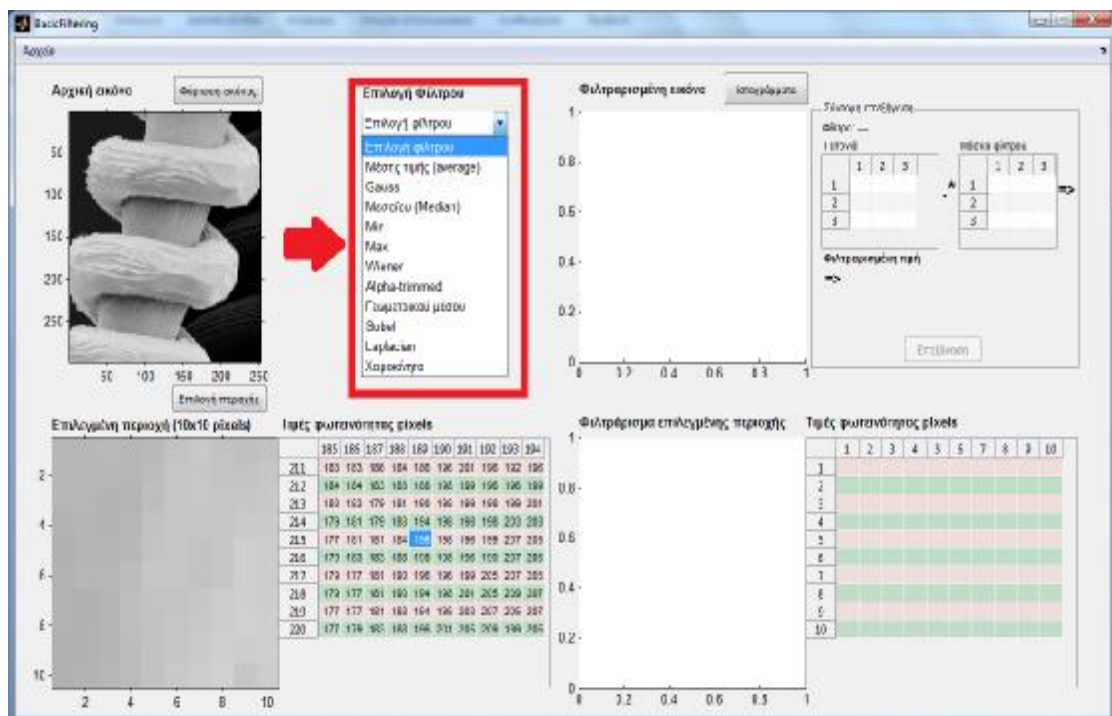


Εικόνα 43: Ενημέρωση αρχικής οθόνης εφαρμογής με τις τιμές φωτεινότητας των pixels της επιλεγμένης περιοχής ενδιαφέροντος

Αφού επιλεγεί η περιοχή ενδιαφέροντος, ενεργοποιείται το χειριστήριο επιλογής φίλτρου. Ο χρήστης μπορεί από τη λίστα επιλογής φίλτρου να επιλέξει το φίλτρο που τον ενδιαφέρει. Τα φίλτρα που υποστηρίζονται είναι τα εξής:

- Μέσης τιμής (Average)
- Gauss
- Μεσαίου (Median)
- Min
- Max
- Wiener
- Περικεκομένου μέσου (alpha - trimmed)
- Γεωμετρικού μέσου
- Sobel
- Laplacian
- Χειροκίνητο, όπου ο χρήστης μπορεί να κατασκευάσει τη δική του μάσκα.

Στην Εικόνα 44 φαίνεται η λίστα με όλα τα φίλτρα που υποστηρίζει η εφαρμογή.

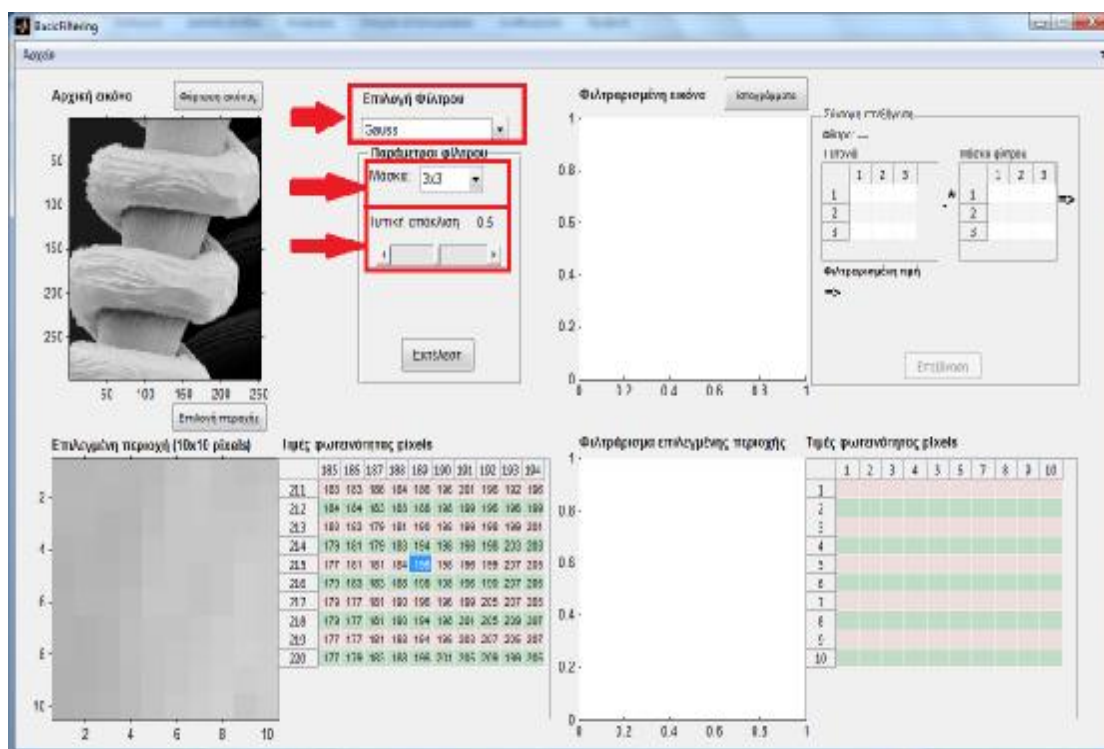


Εικόνα 44: Λίστα επιλογής χωρικού φίλτρου

Μόλις ο χρήστης επιλέξει από τη λίστα το φίλτρο που επιθυμεί, τότε ενεργοποιείται το πλαίσιο με τις παραμέτρους του φίλτρου, καθώς και το πλήκτρο «Εκτέλεση», το οποίο τελικά εκτελεί το φιλτράρισμα. Όλα τα φίλτρα έχουν ως

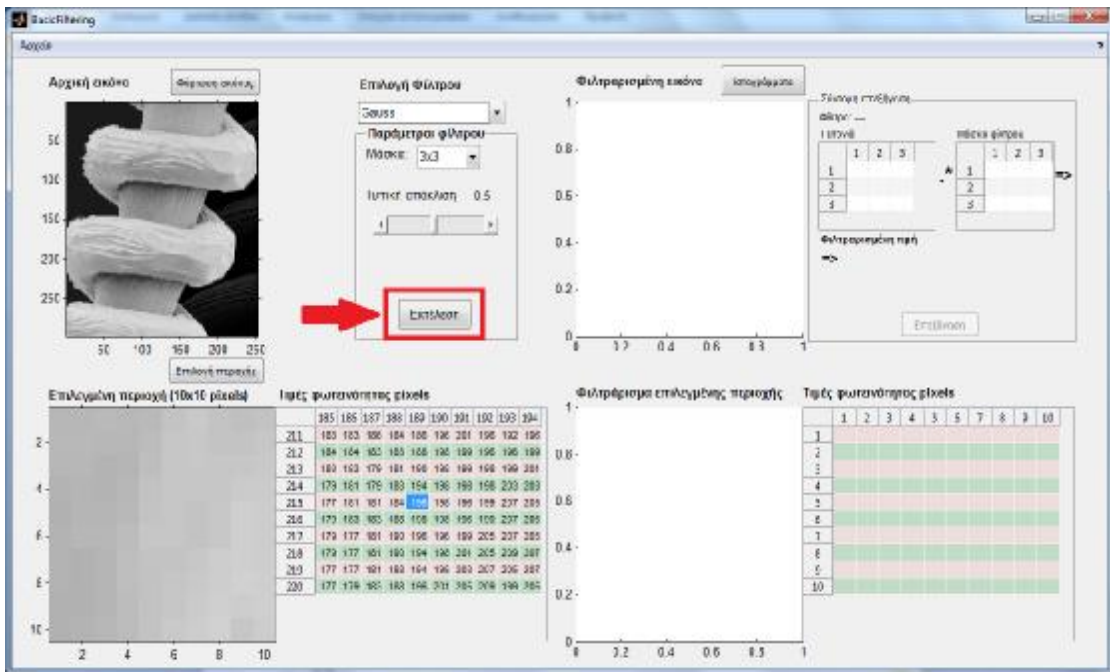
βασική παράμετρο το μέγεθος της μάσκας. Υποστηρίζονται οι μάσκες μεγέθους 3x3 και 5x5, όπου το μέγεθος 3x3 είναι προεπιλεγμένο.

Στην Εικόνα 45 φαίνονται οι παράμετροι του φίλτρου Gauss, όπου η μάσκα έχει το προεπιλεγμένο μέγεθος 3x3 και η τυπική απόκλιση έχει ως προεπιλεγμένη τιμή το 0.5.



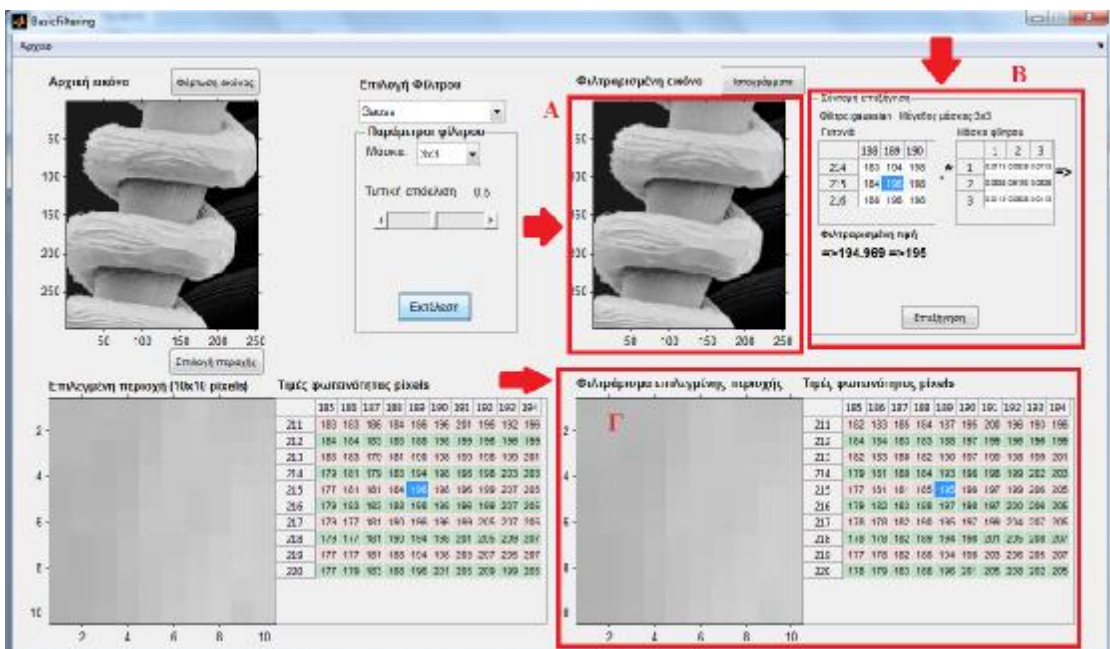
Εικόνα 45: Παράμετροι φίλτρου Gauss

Εφόσον ο χρήστης επιλέξει τις τιμές των παραμέτρων του φίλτρου, μπορεί ακολούθως να πατήσει το πλήκτρο «Εκτέλεση» (Εικόνα 46), το οποίο θα εκτελέσει το φιλτράρισμα της αρχικής εικόνας, καθώς και της επιλεγμένης περιοχής ενδιαφέροντος.



Εικόνα 46: Πλήκτρο «Εκτέλεση»

Τα αποτελέσματα του φιλτραρίσματος εμφανίζονται στο δεξιό τμήμα της οθόνης της εφαρμογής, όπως φαίνεται στην Εικόνα 47.



Εικόνα 47: Ενημέρωση αρχικής οθόνης εφαρμογής με τα αποτελέσματα του φιλτραρίσματος

Παρατηρώντας την Εικόνα 47:

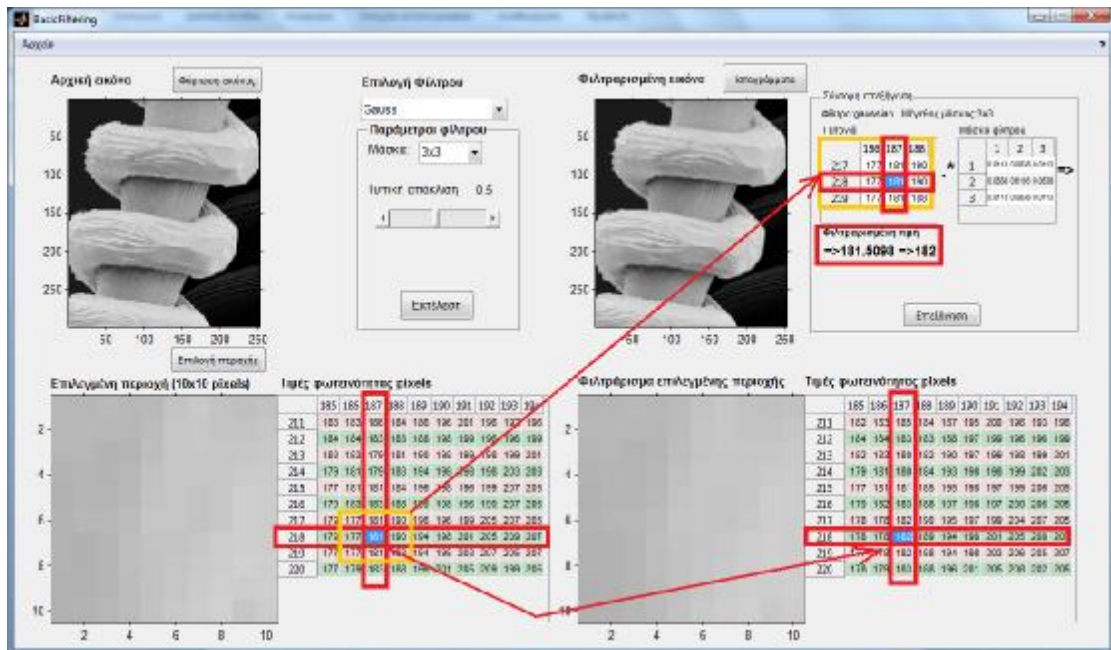
- Στο τμήμα (Α) φαίνεται η συνολική φιλτραρισμένη εικόνα.
- Στο τμήμα (Β) φαίνεται μία σύντομη επεξήγηση της διαδικασίας του χωρικού φιλτραρίσματος. Η επεξήγηση αυτή περιλαμβάνει την εμφάνιση της γειτονιάς του επιλεγμένου *rixel* της περιοχής ενδιαφέροντος, τη μάσκα του φίλτρου (εφόσον ο χρήστης έχει επιλέξει κάποιο γραμμικό φίλτρο) και την φιλτραρισμένη τιμή του επιλεγμένου *rixel*.
- Στο τμήμα (Γ) φαίνεται η φιλτραρισμένη περιοχή ενδιαφέροντος, καθώς και οι τιμές φωτεινότητας των *rixels* της. Στην επάνω και στην αριστερή πλευρά του πίνακα με τις τιμές φωτεινότητας των *rixels* της φιλτραρισμένης επιλεγμένης περιοχής φαίνεται η αρίθμηση της γραμμής και της στήλης των *rixels*. Στον πίνακα των φιλτραρισμένων φωτεινοτήτων αυτομάτως επιλέγεται (φωτίζεται) το *rixel* που αντιστοιχεί στο επιλεγμένο *rixel* της περιοχής ενδιαφέροντος.

Αναφορικά με τον τρόπο χειρισμού των φιλτραρισμένων τιμών ισχύουν τα εξής:

- Αν οι φιλτραρισμένες τιμές είναι δεκαδικές, τότε στρογγυλοποιούνται στον πλησιέστερο ακέραιο.
- Αν οι φιλτραρισμένες τιμές είναι μικρότερες του 0 ή μεγαλύτερες του 255, τότε λαμβάνουν τις τιμές 0 και 255 αντίστοιχα.

Στο σημείο αυτό επισημαίνεται ότι ο χρήστης έχει τη δυνατότητα να επιλέξει κάποιο άλλο από τα *rixels* της περιοχής ενδιαφέροντος και να παρατηρήσει πως αλλάζει η τιμή φωτεινότητας του *rixel* μετά το φιλτράρισμά του. Η επιλογή ενός *rixel* της περιοχής ενδιαφέροντος γίνεται κάνοντας αριστερό κλικ με το ποντίκι πάνω στο επιθυμητό *rixel* στο πίνακα φωτεινοτήτων της περιοχής ενδιαφέροντος. Στην περίπτωση αυτή, θα ενημερωθεί η γειτονιά του επιλεγμένου *rixel* και θα υπολογιστεί η φιλτραρισμένη τιμή του, ενώ επιπλέον θα επιλεγεί (φωτιστεί) το αντίστοιχο *rixel* στον πίνακα φωτεινοτήτων της φιλτραρισμένης περιοχής ενδιαφέροντος.

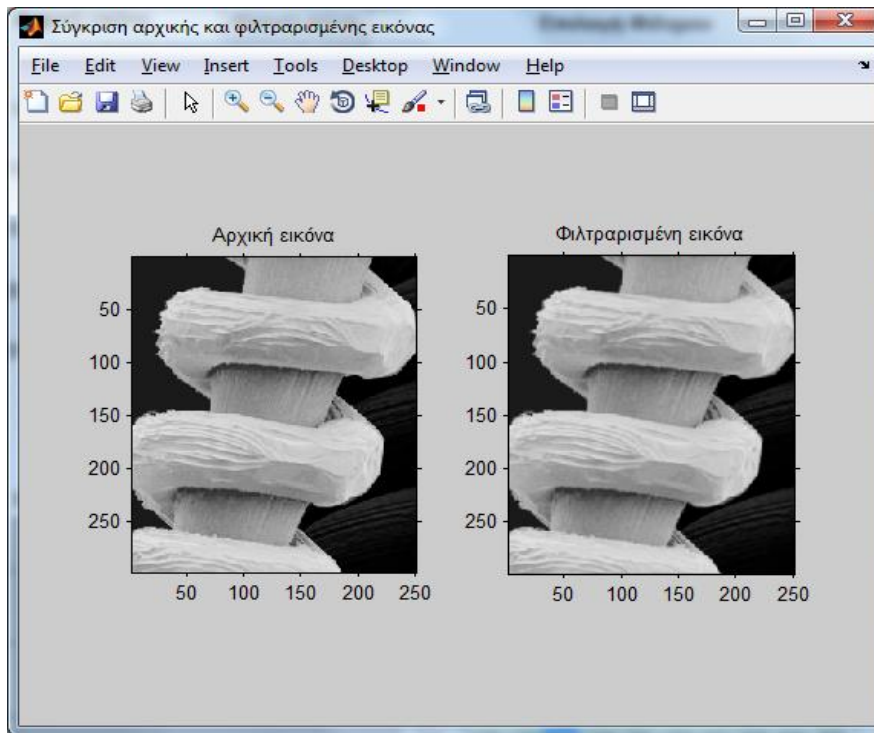
Στην Εικόνα 48 φαίνεται η επιλογή του pixel με συντεταγμένες (218, 187) στον πίνακα φωτεινότητας της περιοχής ενδιαφέροντος, καθώς και οι μεταβολές που συνοδεύουν τη συγκεκριμένη επιλογή.



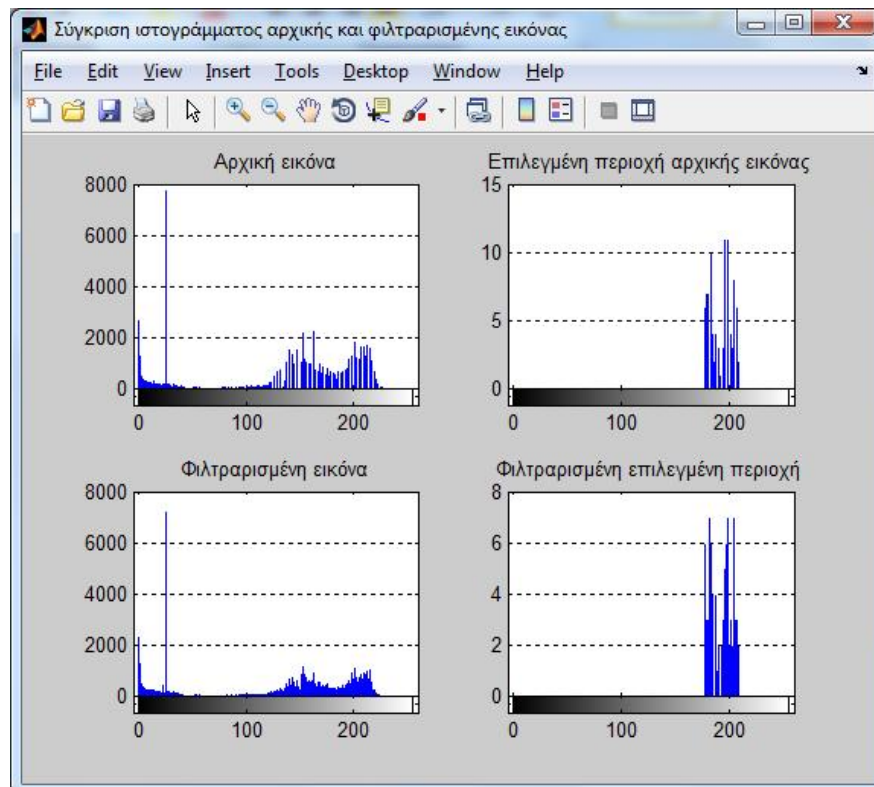
Εικόνα 48: Επιλογή του pixel (218, 187) της περιοχής ενδιαφέροντος και ενημέρωση της αρχικής οθόνης της εφαρμογής

Τώρα πλέον ο χρήστης μπορεί πατώντας το πλήκτρο «Ιστογράμματα» να δει και να συγκρίνει τα ιστογράμματα της αρχικής και της φιλτραρισμένης εικόνας, καθώς και της περιοχής ενδιαφέροντος πριν και μετά το φιλτράρισμα. Επιπλέον, για λόγους πληρότητας εμφανίζεται σε ένα ξεχωριστό παράθυρο η αρχική και η τελική εικόνα, προκειμένου ο χρήστης να μπορεί να κάνει ευκολότερα οπτική σύγκριση.

Στις παρακάτω δύο (2) εικόνες (Εικόνα 49 και Εικόνα 50) φαίνονται τα σχετικά παράθυρα, που διαθέτουν χειριστήρια επεξεργασίας, όπως εστίασης, αποθήκευσης κλπ.

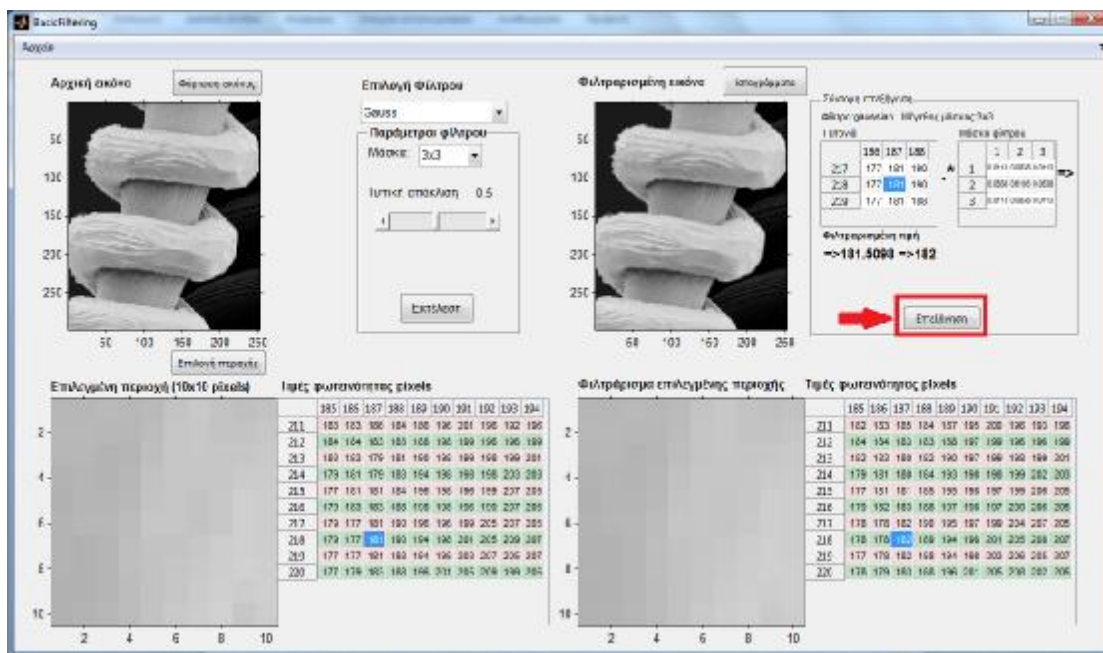


Εικόνα 49: Σύγκριση εικόνας πριν και μετά το φιλτράρισμα



Εικόνα 50: Σύγκριση ιστογραμμάτων πριν και μετά το φιλτράρισμα

Επιπλέον, εφόσον έχει εκτελεστεί το φιλτράρισμα κάποιας εικόνας, ο χρήστης έχει τη δυνατότητα να επιλέξει να δει σε ένα νέο παράθυρο την αναλυτική επεξήγηση της διαδικασίας του φιλτραρίσματος για το επιλεγμένο φίλτρο, πατώντας το πλήκτρο «Επεξήγηση» που φαίνεται στην Εικόνα 51.

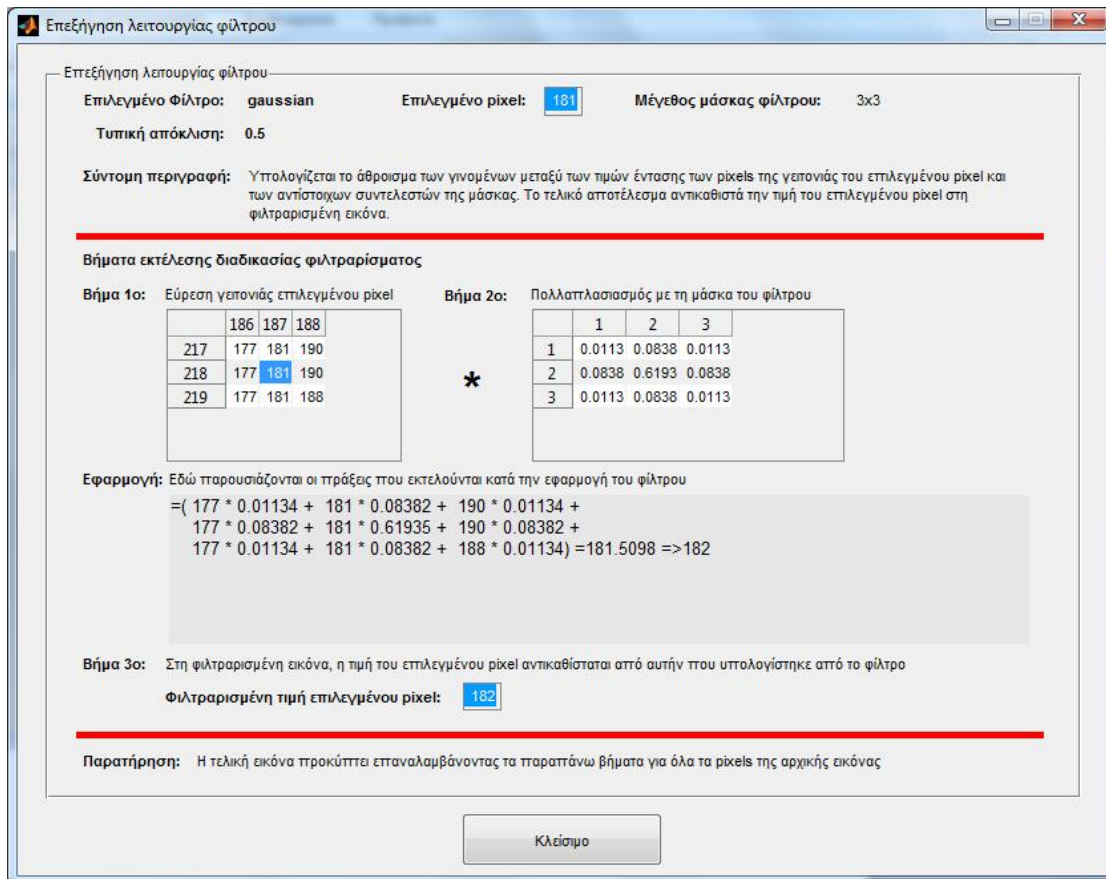


Εικόνα 51: Πλήκτρο «Επεξήγηση»

Το παράθυρο της αναλυτικής περιγραφής της διαδικασίας φιλτραρίσματος για ένα επιλεγμένο φίλτρο (Εικόνα 52), περιλαμβάνει τα εξής στοιχεία:

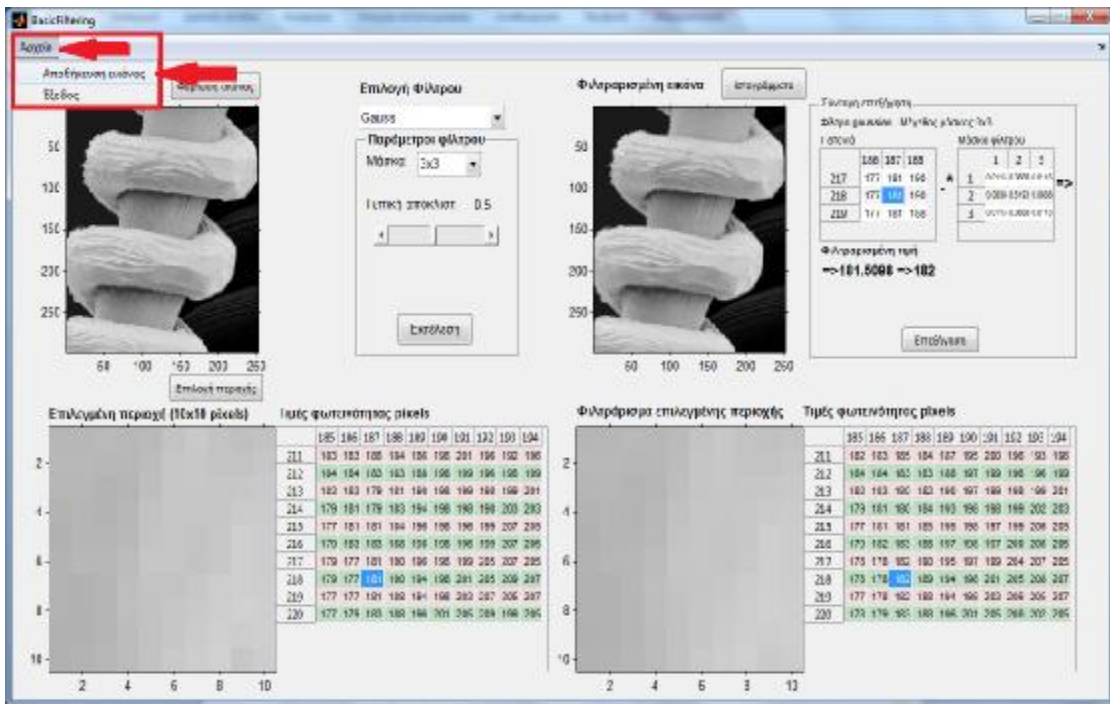
- Όνομα φίλτρου
- Τιμή φωτεινότητας επιλεγμένου ρixel
- Μέγεθος μάσκας φίλτρου
- Λοιπές παράμετροι φίλτρου
- Σύντομη επεξήγηση
- Αναλυτική περιγραφή σε μορφή αριθμημένων βημάτων της διαδικασίας του φιλτραρίσματος, που περιλαμβάνει:
 - Εύρεση της γειτονιάς του επιλεγμένου ρixel
 - Μάσκα φίλτρου (εφόσον πρόκειται για γραμμικό φίλτρο)

- Αναλυτική περιγραφή και παρουσίαση όλων των πράξεων υπολογισμού που απαιτεί το φίλτρο
- Τελική φιλτραρισμένη τιμή

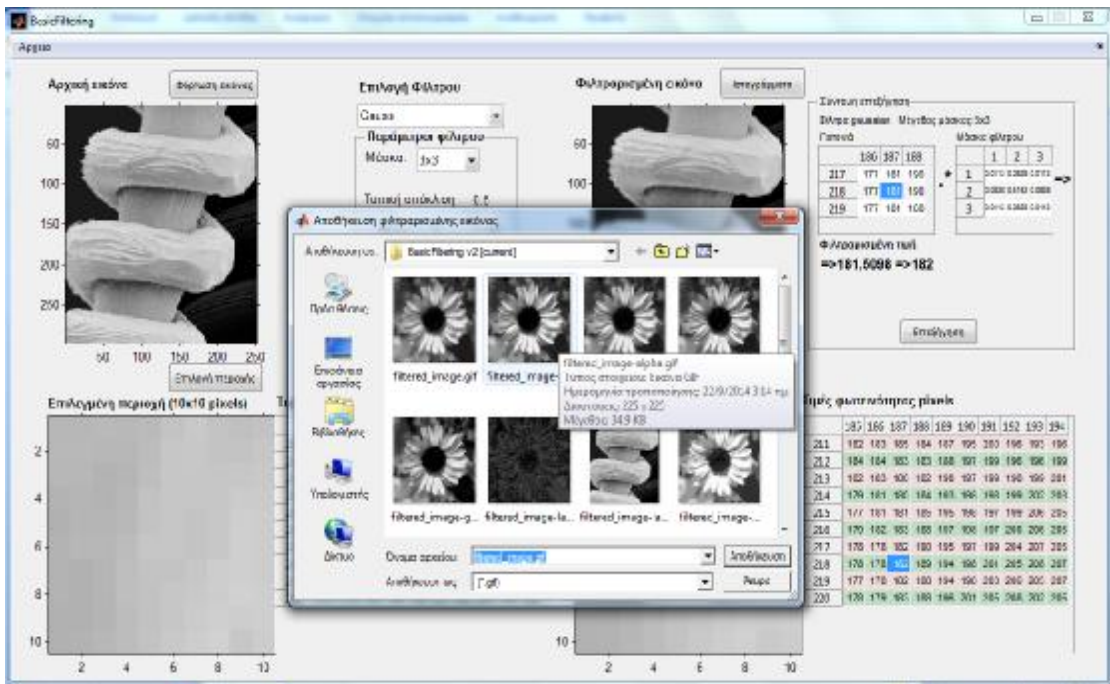


Εικόνα 52: Παράθυρο επεξήγησης διαδικασίας φιλτραρίσματος με το φίλτρο Gauss

Από το μενού της αρχικής οθόνης της εφαρμογής, ο χρήστης μπορεί να αποθηκεύσει τη φιλτραρισμένη εικόνα, επιλέγοντας «Αρχείο» \Rightarrow «Αποθήκευση». Στην περίπτωση αυτή, εμφανίζεται ένα παράθυρο διαλόγου, που προτρέπει τον χρήστη να επιλέξει τη θέση και το όνομα αποθήκευσης του αρχείου. Αφού συμπληρωθούν τα στοιχεία αυτά, πατώντας το πλήκτρο «Αποθήκευση» ολοκληρώνεται η αποθήκευση της φιλτραρισμένης εικόνας σε αρχείο. Η διαδικασία αυτή φαίνεται στις δύο (2) παρακάτω εικόνες (Εικόνα 53 και Εικόνα 54).



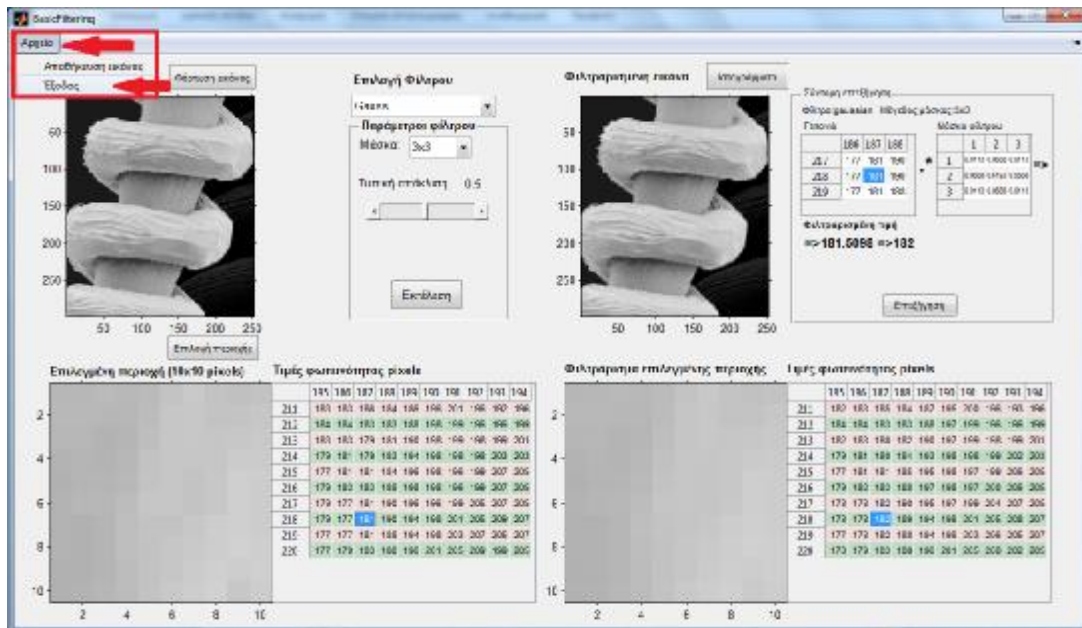
Εικόνα 53: Επιλογή «Αποθήκευση εικόνας»



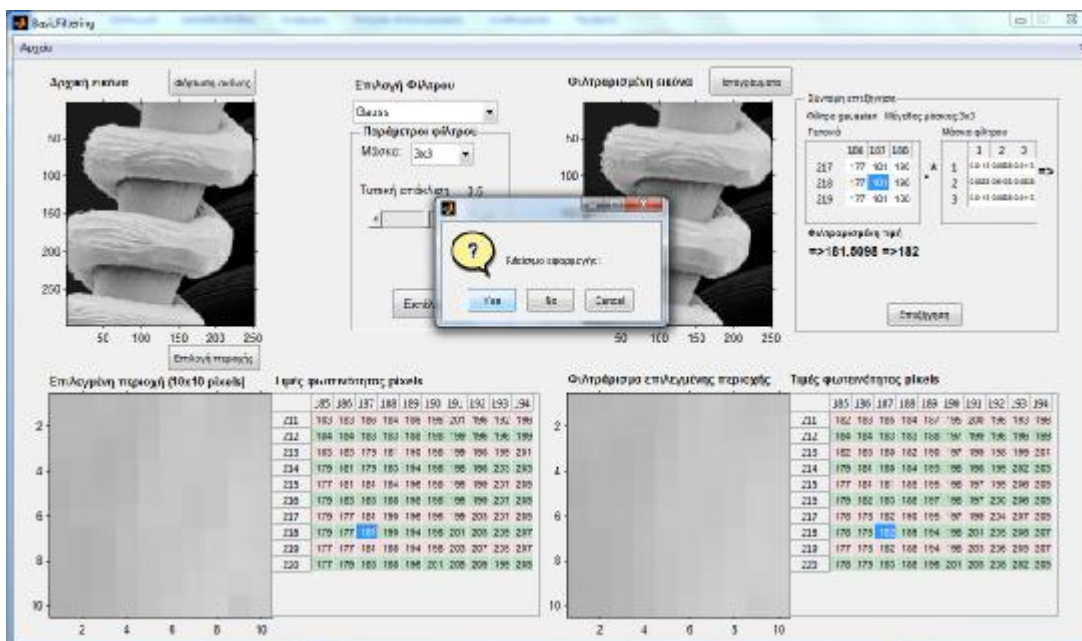
Εικόνα 54: Παράθυρο αποθήκευσης εικόνας

Ο χρήστης μπορεί να φορτώσει μία νέα εικόνα ή να επιλέξει μία άλλη περιοχή ενδιαφέροντος ή να επιλέξει ένα διαφορετικό φίλτρο και να επαναλάβει τις παραπάνω διαδικασίες.

Σε περίπτωση που επιθυμεί να τερματίσει την εφαρμογή, μπορεί να επιλέξει από την αρχική οθόνη της εφαρμογής το μενού «Αρχείο» à «Εξόδος». Στο μήνυμα επιβεβαίωσης κλεισίματος της εφαρμογής που θα εμφανιστεί θα πρέπει να επιλέξει «Ναι», οπότε και θα τερματιστεί η εφαρμογή. Στις επόμενες εικόνες (Εικόνα 55 και Εικόνα 56) φαίνεται η διαδικασία τερματισμού της εφαρμογής.



Εικόνα 55: Επιλογή «Εξόδος»



Εικόνα 56: Επιβεβαίωση εξόδου από την εφαρμογή

4 Συμπεράσματα

Παρουσιάστηκε το βασικό θεωρητικό πλαίσιο της εργασίας, εστιάζοντας στην τεχνική του χωρικού φιλτραρίσματος και στα βασικά χωρικά φίλτρα. Επίσης, παρουσιάστηκαν σύντομα κάποιες εκπαιδευτικές εφαρμογές που χρησιμοποιούνται για την εκπαίδευση μέσω υπολογιστή εννοιών και τεχνικών της ψηφιακής επεξεργασίας εικόνας.

Ακολούθως, παρουσιάστηκαν τα βασικά χαρακτηριστικά του Matlab, του περιβάλλοντος GUIDE που διαθέτει για τη δημιουργία γραφικών διεπαφών, καθώς και της εργαλειοθήκης επεξεργασίας εικόνας (Image Processing Toolbox) του Matlab.

Επίσης, παρουσιάστηκε αναλυτικά η σχεδίαση και η ανάπτυξη εφαρμογής με τα εργαλεία αυτά, με σκοπό να χρησιμοποιηθεί ως εκπαιδευτικό βοήθημα από σπουδαστές για την αναλυτικότερη παρουσίαση και επεξήγηση βασικών εννοιών της ψηφιακής επεξεργασίας εικόνας. Στην τρέχουσα έκδοσή της η εφαρμογή καλύπτει εκτενέστατα την τεχνική του χωρικού φιλτραρίσματος εικόνων αποχρώσεων του γκρι, ενσωματώνοντας τα βασικότερα χωρικά φίλτρα και παρέχοντας τη δυνατότητα δημιουργίας και πειραματισμού με νέες χωρικές μάσκες, που μπορεί να ορίσει ο χρήστης. Η βασικότερη καινοτομία της εφαρμογής είναι η αναλυτικότερη παρουσίαση και επεξήγηση της τεχνικής του χωρικού φιλτραρίσματος σε επίπεδο pixel.

5 Προοπτικές - Μελλοντικές Βελτιώσεις

Η εφαρμογή που υλοποιήθηκε καλύπτει κυρίως το πεδίο του χωρικού φιλτραρίσματος εικόνων αποχρώσεων του γκρι, χρησιμοποιώντας τα βασικότερα χωρικά φίλτρα. Ωστόσο, η εφαρμογή έχει ανοικτή αρχιτεκτονική και είναι εύκολα επεκτάσιμη.

Στις μελλοντικές βελτιώσεις της εφαρμογής μπορούν να ενταχθούν οι εξής:

- η διαδικασία του φιλτραρίσματος να μπορεί να εκτελεστεί επαναληπτικά, όπου ο χρήστης να ορίζει τον επιθυμητό αριθμό επαναλήψεων και σε κάθε επανάληψη ο χρήστης να μπορεί να συγκρίνει τα αποτελέσματα της εφαρμογής του φίλτρου.
- Να υποστηρίζονται οι έγχρωμες (RGB) εικόνες
- Ανάλυση ιστογράμματος
- Τεχνικές επεξεργασίας ιστογράμματος
- Μετασχηματισμοί σημείου
- Φιλτράρισμα στο πεδίο των συχνοτήτων
- Μετασχηματισμοί wavelets
- Προσθήκη διαφόρων τύπων θορύβου στην αρχική εικόνα
- Γεωμετρικοί μετασχηματισμοί
- Ανίχνευση αντικειμένων
- Τεχνικές συμπίεσης

Βιβλιογραφία

- CS425 Lab: *Intensity Transformations and Spatial Filtering*. (2011). Ανάκτηση 5 10, 2014, από Department of Computer Science, University of Regina: <http://www.cs.uregina.ca/Links/class-info/425/Lab3/index.html>
- Gonzalez, R. C., & Woods, R. E. (2007). *Digital Image Processing* (3rd ed.). Prentice Hall.
- Kekre, H. B., & Gharge, S. M. (2010). Image Segmentation using Extended Edge Operator for Mammographic Images. (*IJCSE*) *International Journal on Computer Science and Engineering*, 02 (04), pp. 1086-1091.
- Kim, S. (2013, August 20). Applications of Convolution in Image Processing with Matlab. University of Washington.
- Krasula, L., Klima, M., Rogard, E., & Jeanblanc, E. (2011, December). MATLAB-based Applications for Image Processing and Image Quality Assessment – Part I: Software Description., 20 (4), pp. 1009-1015.
- Krasula, L., Klima, M., Rogard, E., & Jeanblanc, E. (2012, April). MATLAB-based Applications for Image Processing and Image Quality Assessment Part II: Experimental Results. *Radioengineering*, 21 (1), pp. 154-161.
- Lent, C. S. (2013). *Learning To Program With Matlab: Building GUI Tools*. Wiley & Sons.
- Marchand, P., & Holland, T. O. (2003). *Graphics And GUIs With Matlab* (3rd ed.). Washington DC: Chapman & Hall/CRC.
- MathWorks.com. (2012). *Image Processing Toolbox*. Ανάκτηση 5 21, 2014, από MathWorks: <http://www.mathworks.com/help/images/index.html>
- Rajashekar, U., Bovik, A. C., Sage, D., Unser, M., Karam, L. J., & Lagendijk, R. L. (2004). Image Processing Education.

ROBOTIX. (2014). *ROBOTIX Image Tutorials*. Ανάκτηση 3 2014, 3, από ROBOTIX Tutorials:

<http://www.robotix.in/tutorials/categ/imageprocessing/imagetypes>

Weisstein, E. W. (2014). *Convolution*. Ανάκτηση 5 7, 2014, από Wolfram MathWorld:

<http://mathworld.wolfram.com/Convolution.html>

Βαρσάμης, Δ. (2006). *MATLAB Graphical User Interfaces: Σημειώσεις GUI για το εργαστήριο του μαθήματος Ασαφή Συστήματα*. Σέρρες: ΤΕΙ Σερρών - Τμήμα Πληροφορικής και Επικοινωνιών.

Στρουθόπουλος, Χ. Π. (2009, Δεκέμβριος). *Σημειώσεις Μαθήματος: Ψηφιακή Επεξεργασία Εικόνας*. Σέρρες: ΤΕΙ Σερρών - Τμήμα Πληροφορικής και Επικοινωνιών.