



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ  
ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

**Μίξη πολυεστιακών εικόνων με χρήση κατάτμησης περιοχών  
και χωρικής συχνότητας**

---

**Πτυχιακή εργασία του**

Τόντζου Φωτίου (2896)

Επιβλέπων: Δρ. Αθανάσιος Νικολαΐδης, Αναπληρωτής Καθηγητής

**ΣΕΡΡΕΣ, ΣΕΠΤΕΜΒΡΙΟΣ 2017**

**Υπεύθυνη Δήλωση** : Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής Τ.Ε., του Τ.Ε.Ι. Κεντρικής Μακεδονίας.

## ΠΕΡΙΛΗΨΗ

Συχνά, οι ψηφιακές εικόνες που λαμβάνονται έχουν περιοχές με μη καλή εστίαση. Αυτό μπορεί να οφείλεται είτε σε περιβαλλοντικούς παράγοντες ή στο ότι δόθηκε σημασία στην καλή εστίαση άλλων περιοχών. Στην παρούσα πτυχιακή εργασία μελετάται και υλοποιείται μία μέθοδος μίξης πολυεστιακών εικόνων που στόχο έχει να παραχθεί μια εικόνα που να είναι εξ ολοκλήρου καλά εστιασμένη. Αρχικά, οι πολυεστιακές εικόνες συνδυάζονται χρησιμοποιώντας τον μέσο όρο τους. Στη συνέχεια η εικόνα που προκύπτει κατατέμνεται με χρήση της μεθόδου των normalized cuts. Κατόπιν, οι αρχικές εικόνες κατατέμνονται βάσει του αποτελέσματος κατάτμησης της ενδιάμεσης εικόνας. Τέλος, οι αντίστοιχες κατατετμημένες περιοχές των εικόνων πηγής συνδυάζονται σύμφωνα με τις χωρικές τους συχνότητες. Η τεχνική εφαρμόστηκε σε διάφορα ζεύγη πολυεστιακών εικόνων προκειμένου να επαληθευτεί η αποδοτικότητα της. Τα αποτελέσματα έδειξαν ότι με τη μέθοδο αυτή, η τελική εικόνα είναι εστιακά καλύτερη από τις δύο αρχικές, ωστόσο υπάρχουν περιθώρια βελτιστοποίησης.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	2
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	4
ΚΕΦΑΛΑΙΟ 2: Παρουσίαση της μεθόδου .....	6
2.1 Πίνακας Βαρών.....	6
2.2 Κατάτμηση με normalized cut.....	7
2.3 Παρουσίαση της μεθόδου .....	8
2.4 Χωρικές συχνότητες .....	10
2.5 Μίξη πολυεστιακών εικόνων .....	12
ΚΕΦΑΛΑΙΟ 3: Εφαρμογή της μεθόδου σε ζεύγη πολυεστιακών εικόνων .....	14
3.1 Αναλυτική παρουσίαση μίξης ενός ζεύγους πολυεστιακών εικόνων.....	14
3.2 Συγκεντρωτικά αποτελέσματα για τα υπόλοιπα ζεύγη πολυεστιακών εικόνων.....	16
ΚΕΦΑΛΑΙΟ 4: Συμπεράσματα.....	20
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	22
ΠΑΡΑΡΤΗΜΑ 1: Εφαρμογή του αλγορίθμου με Matlab Graphical User Interface .....	23
ΠΑΡΑΡΤΗΜΑ 2: Κώδικες Matlab .....	26

## ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Οι εικόνες που λαμβάνονται, συχνά δεν εστιάζουν καλά σε όλες τις περιοχές της εικόνας. Ένα τέτοιο παράδειγμα φαίνεται στην Εικόνα 1.1 όπου μόνο το πίσω μέρος της απεικόνισης φαίνεται καθαρά.



Εικόνα 1.1 : Εστίαση στο πίσω μέρος της εικόνας

Η διόρθωση αυτού του προβλήματος με χρήση μόνο αυτής της εικόνας, δηλαδή η απομάκρυνση του θορύβου από τη μη καλά εστιασμένη περιοχή, δεν είναι εύκολη. Αν, όμως, υπάρχουν δύο εικόνες που απεικονίζουν την ίδια σκηνή αλλά εστιάζουν σε διαφορετικά μέρη η κάθε μία, όπως φαίνεται στην Εικόνα 1.2, τότε υπάρχουν διάφορες μεθοδολογίες που προσφέρουν λύσεις. Κάποιες από αυτές κάνουν χρήση παλμικά συζευγμένων νευρωνικών δικτύων (pulse coupled neural networks) (Huang & Jing, 2007; Wang, Ma, & Gu, 2010), πυραμίδας κλίσης (gradient pyramid) (Adelson, Anderson, Bergen, Burt, & Ogden, 1984) ή χρήση μετασχηματισμού κυματιδίων (Pu & Ni, 2000).

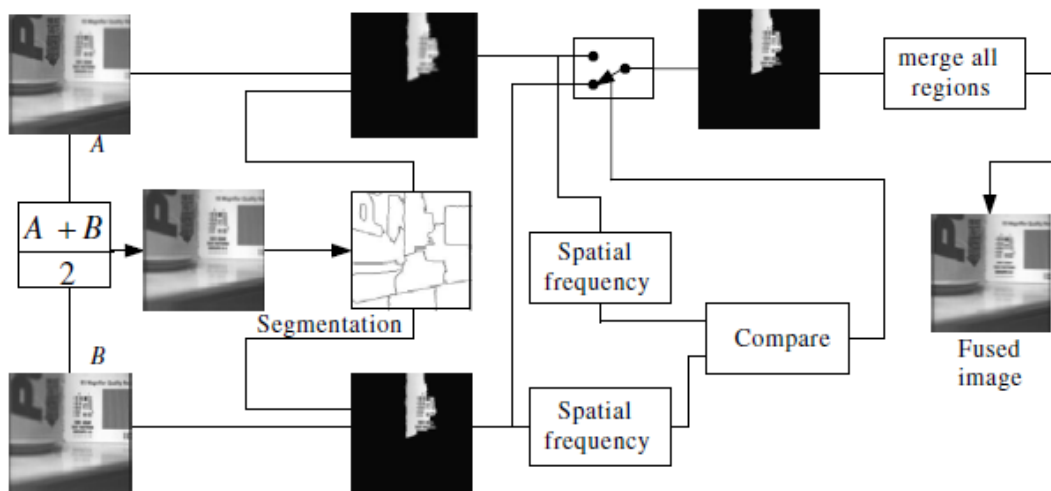


Εικόνα 1.2: Δύο αντίγραφα της ίδιας σκηνής που εστιάζουν σε διαφορετικές περιοχές

Στην παρούσα εργασία, γίνεται χρήση μίας άλλης μεθόδου, που συνδυάζει δύο μεθοδολογίες: μία για να κόψει τη σκηνή σε διαφορετικά μέρη με χρήση normalized cuts

(Shi & Malik, 2000) και μια δεύτερη για να συνδυάσει τα μέρη που προέκυψαν από το πρώτο βήμα με χρήση χωρικών συχνοτήτων (Li & Yang, 2008).

Τα βασικά βήματα της μεθοδολογίας φαίνονται στο Σχήμα 1.1 ξεκινώντας από αριστερά. Αρχικά υπολογίζεται ο μέσος όρος των δύο αρχικών εικόνων. Η μεθοδολογία των normalized cuts υλοποιείται πάνω στο μέσο όρο και ο προηγούμενος μέσος όρος των εικόνων κατατέμνεται. Στη συνέχεια, οι αρχικές εικόνες κατατέμνονται όμοια με την κατάτμηση του μέσου όρου (Σχήμα 1.1, στο κέντρο). Τέλος, οι αντίστοιχες περιοχές των εικόνων συγκρίνονται με τη μέθοδο των χωρικών συχνοτήτων και επιλέγεται κάθε φορά αυτή με τη μεγαλύτερη τιμή χωρικής συχνότητας. Με τον τρόπο αυτό επιλέγονται οι «καλύτερες» περιοχές για να σχηματιστεί το τελικό παζλ, που είναι η τελική μιξαρισμένη εικόνα.



Σχήμα 1.1: Μεθοδολογία για τη μίξη πολυεστιακών εικόνων με χρήση περιοχών κατάτμησης και χωρικές συχνότητες. Ανατυπώθηκε από "Multifocus image fusion using region segmentation and spatial frequency" των S. Li και B. Yang, 2008, *Image and Vision Computing*, 26 (σελ. 973).

Η παραπάνω μεθοδολογία υλοποιήθηκε στο γραφικό περιβάλλον του Matlab (έκδοση R2016a). Οι μεθοδολογίες για την κατάτμηση με normalized cuts καθώς και για τη χρήση χωρικών συχνοτήτων παρουσιάζονται αναλυτικά στο Κεφάλαιο 2. Στο Κεφάλαιο 3 παρουσιάζονται τα αποτελέσματα όλης της μεθόδου, όταν αυτή εφαρμόστηκε σε διάφορα ζεύγη πολυεστιακών εικόνων επιπέδων του γκρίζου. Τα συμπεράσματα της εργασίας παρουσιάζονται στο Κεφάλαιο 4. Τέλος, στα Παραρτήματα παρουσιάζεται όλος ο κώδικας Matlab της εργασίας.

## ΚΕΦΑΛΑΙΟ 2: Παρουσίαση της μεθόδου

### 2.1 Πίνακας Βαρών

Πριν προχωρήσουμε στην παρουσίαση της μεθόδου και του αλγορίθμου, θα αναφερθούμε σε ένα πολύ σημαντικό μαθηματικό εργαλείο, γνωστό στη βιβλιογραφία ως *normalized cut*, το οποίο έχει σαν αποτέλεσμα ένα βέλτιστο τρόπο κατάτμησης της εικόνας (Shi & Malik, 1998).

Αρχικά θα αναπαραστήσουμε μια εικόνα με έναν γράφο  $G(\mathbf{V}, \mathbf{E})$  όπου  $\mathbf{V}$  το σύνολο των κόμβων (εδώ, τα εικονοστοιχεία) και  $\mathbf{E}$  οι ακμές του. Κάθε ακμή που συνδέει δύο κόμβους έχει και ένα βάρος  $w(i,j)$  που εξαρτάται από τη διαφορά των εντάσεων των κόμβων  $F(i) - F(j)$  και από την απόσταση τους  $X(i) - X(j)$ . Μεγάλη τιμή βάρους μεταξύ δύο κόμβων σημαίνει ότι οι εντάσεις τους διαφέρουν λίγο και η απόστασή τους είναι μικρή. Η σχέση του βάρους δίνεται από τον τύπο.

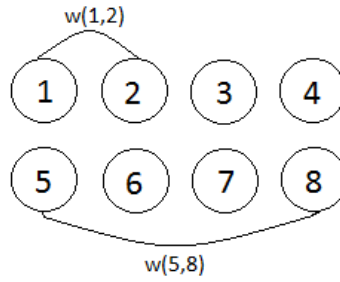
$$W(i, j) = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_f^2}} \cdot \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_x^2}}, & \text{όταν } \|X(i) - X(j)\| < r \\ 0 & , \text{ σε άλλη περίπτωση} \end{cases} \quad (2.1)$$

όπου  $X(i) - X(j)$  είναι η απόσταση των κόμβων  $i$  και  $j$  και ισούται με την Ευκλείδεια απόσταση (Gonzalez & Woods, 2011, σ. 112):

$$\|X(i) - X(j)\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Στην παραπάνω σχέση τα  $x_i, x_j$  είναι οι στήλες και τα  $y_i, y_j$  είναι οι γραμμές στις οποίες βρίσκονται οι κόμβοι στην εικόνα. Η παράμετρος  $\sigma_x$  ισούται με το 10% της συνολικής συνάρτησης απόστασης.

Στο Σχήμα 2.1 φαίνεται μια απλή αναπαράσταση μιας εικόνας  $2 \times 4$  σε μορφή γράφου, όπου κάθε εικονοστοιχείο είναι ένας κόμβος και όλοι οι κόμβοι συνδέονται μεταξύ τους με μια τιμή βάρους  $w(i,j)$ . Στο σχήμα φαίνονται αντιπροσωπευτικά μόνο 2 από τα συνολικά 28 βάρη που υπάρχουν.



Σχήμα 2.1: Αναπαράσταση μιας εικόνας 2×4 σε μορφή γράφου.

## 2.2 Κατάτμηση με normalized cut

Συνεχίζοντας με τον όρο του normalized cut, θεωρούμε αρχικά ότι η εικόνα χωρίζεται σε δύο μέρη A και B (δοκιμαστική κατάτμηση). Δηλαδή, χωρίζοντας την εικόνα στα δύο μέρη, κόβουμε ουσιαστικά τις ακμές που συνδέουν όλα τα στοιχεία του A μέρους με όλα τα στοιχεία του B μέρους. Επειδή όμως σε κάθε ακμή αντιστοιχεί και ένα βάρος, όπως είπαμε και προηγουμένως, το άθροισμα όλων των βαρών μεταξύ του A και B μέρους, θα είναι:

$$cut(A, B) = \sum_{u \in A, t \in B} w(u, t)$$

Έτσι, τελικά, η βέλτιστη κατάτμηση θα είναι αυτή για την οποία το παραπάνω άθροισμα γίνεται ελάχιστο (Shi & Malik, Normalized Cuts and Image Segmentation, 2000). Επειδή, όμως, το παραπάνω άθροισμα ελαχιστοποιείται χωρίζοντας μόνο ένα εικονοστοιχείο από όλα τα υπόλοιπα, για το λόγο αυτό πρέπει πρώτα να κανονικοποιηθεί. Δηλαδή, υπολογίζεται το άθροισμα των λόγων των αθροισμάτων των βαρών μεταξύ A και B προς τα αθροίσματα των βαρών μεταξύ του κάθε μέρους και του συνολικού γράφου (Shi & Malik, 2000).

Έτσι, η σχέση του normalized cut δίνεται από τον παρακάτω τύπο:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (2.2)$$

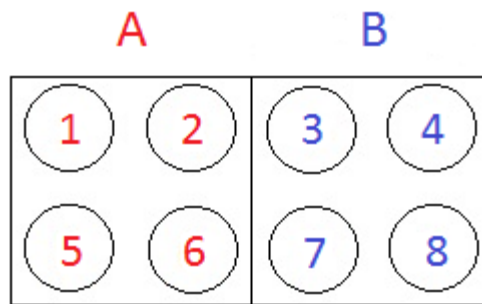
όπου  $cut(A, B) = \sum_{u \in A, t \in B} w(u, t)$ ,

$assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ ,

$assoc(B, V) = \sum_{u \in B, t \in V} w(u, t)$ .



Στο Σχήμα 2.2 φαίνεται ο γράφος του Σχήματος 2.1 χωρισμένος σε δύο δοκιμαστικά μέρη A και B.



Σχήμα 2.2: Κατάτμηση της εικόνας σε δύο μέρη A και B.

Για τον γράφο αυτόν, οι τιμές των  $\text{cut}(A, B)$ ,  $\text{assoc}(A, V)$  και  $\text{assoc}(B, V)$  που βρίσκονται στη Σχέση 2.2 παίρνουν τη μορφή:

$$\begin{aligned} \text{cut}(A, B) &= w(1,3) + w(1,4) + w(1,7) + w(1,8) + w(2,3) + w(2,4) + w(2,7) \\ &\quad + w(2,8) \\ &\quad + w(5,3) + w(5,4) + w(5,7) + w(5,8) + w(6,3) + w(6,4) + w(6,7) + w(6,8) \end{aligned}$$

$$\begin{aligned} \text{assoc}(A, V) &= w(1,2) + w(1,3) + w(1,4) + w(1,5) + w(1,6) + w(1,7) + w(1,8) \\ &\quad + w(2,3) + w(2,4) + w(2,5) + w(2,6) + w(2,7) + w(2,8) + w(5,3) \\ &\quad + w(5,4) + w(5,6) + w(5,7) + w(5,8) + w(6,3) + w(6,4) + w(6,7) \\ &\quad + w(6,8) \end{aligned}$$

$$\begin{aligned} \text{assoc}(B, V) &= w(3,1) + w(3,2) + w(3,4) + w(3,5) + w(3,6) + w(3,7) + w(3,8) \\ &\quad + w(4,1) + w(4,2) + w(4,5) + w(4,6) + w(4,7) + w(4,8) + w(7,1) \\ &\quad + w(7,2) + w(7,5) + w(7,6) + w(7,8) + w(8,1) + w(8,2) + w(8,5) \\ &\quad + w(8,6) \end{aligned}$$

Σύμφωνα με τη συγκεκριμένη μεθοδολογία, η βέλτιστη κατάτμηση της εικόνας θα προκύψει όταν βρεθεί η κατάτμηση για την οποία ελαχιστοποιείται η τιμή του  $N_{\text{cut}}$ .

### 2.3 Παρουσίαση της μεθόδου

Αφού παρουσιάσαμε αναλυτικά τη μέθοδο των normalized cuts θα περάσουμε στη συνέχεια στα βήματα που ακολουθούνται για την κατάτμηση μιας εικόνας σε επιμέρους

περιοχές. Οι Shi και Malik προτείνουν έναν πολύ καλά δομημένο αλγόριθμο, τον οποίο και θα υιοθετήσουμε στην παρούσα εργασία (Li & Yang, 2008).

Για τις ανάγκες του συγκεκριμένου προβλήματος, έχουμε κατά νου την αντιστοιχία που περιγράφηκε αναλυτικά στην προηγούμενη παράγραφο, ότι, δηλαδή, μια εικόνα διαστάσεων  $k \times 1$ , αναπαρίσταται από  $k \times 1$  κόμβους, που στον καθένα αντιστοιχεί μια τιμή έντασης  $F(i)$  σε κλίμακα του γκρι ( $0 < F(i) < 255$ ) και κάθε κόμβος συνδέεται με όλους τους υπόλοιπους με μια τιμή βάρους  $W(i,j)$  (Σχήμα 2.1).

Οι τιμές των εντάσεων των κόμβων  $F(i)$ , κανονικοποιούνται στην κλίμακα από 0 έως 1, όπου το 0 αντιστοιχεί στο 0 της κλίμακας του γκρι και το 1 αντιστοιχεί στο 255 (Shi & Malik, 2000).

1. Βρίσκουμε τον μέσο όρο των δυο αρχικών εικόνων προσθέτοντας τις τιμές των εντάσεων των αντίστοιχων κόμβων και στη συνέχεια διαιρώντας με το δύο.
2. Υπολογίζουμε τον πίνακα βαρών

$$W(i,j) = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_f^2}} \cdot \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_x^2}}, & \text{όταν } \|X(i) - X(j)\| < r \\ 0 & , \text{ σε άλλη περίπτωση} \end{cases} \quad (2.1)$$

Ο πίνακας των βαρών είναι συμμετρικός και έχει διαστάσεις ( $k \cdot 1 \times k \cdot 1$ ), μία σειρά για κάθε κόμβο. Παρατηρούμε ότι το βάρος μεταξύ δύο κόμβων μηδενίζεται όταν η απόσταση τους γίνει μεγαλύτερη μιας προκαθορισμένης απόστασης  $r$ .

3. Υπολογίζουμε έναν διαγώνιο πίνακα  $D$  που προκύπτει ύστερα από πρόσθεση όλων των στοιχείων κάθε γραμμής (ή στήλης) του πίνακα  $W$ .
4. Λύνουμε την εξίσωση ιδιοτιμών:

$$(D - W)x = \lambda Dx. \quad (2.3)$$

Η κατάτμηση της εικόνας θα βασιστεί στο ιδιοδιάνυσμα που αντιστοιχεί στη δεύτερη μικρότερη ιδιοτιμή. Από τις τιμές του ιδιοδιανύσματος αυτού, βρίσκουμε 10 τιμές (splitting points) που ισαπέχουν μεταξύ τους. Για κάθε ένα splitting point χωρίζουμε όλους τους κόμβους της εικόνας σε δύο μέρη. Το πρώτο μέρος A είναι τα σημεία που αντιστοιχούν σε τιμές μικρότερες του splitting point και το δεύτερο μέρος B είναι τα σημεία που αντιστοιχούν σε τιμές μεγαλύτερες του splitting point. Για τον κάθε διαχωρισμό της εικόνας σε δύο μέρη A και B, υπολογίζουμε την τιμή του normalized

cut που περιγράφηκε στην προηγούμενη παράγραφο (Σχέση 2.2). Η διαδικασία αυτή πραγματοποιείται 10 φορές (μία για κάθε ένα splitting point) και επιλέγουμε να κόψουμε, την εικόνα σύμφωνα με εκείνο το splitting point που αντιστοιχεί στη μικρότερη τιμή του αντίστοιχου  $N_{cut}$ .

5. Στη συνέχεια πρέπει να διερευνήσουμε αν το ιδιοδιάνυσμα που επιλέχθηκε έχει τιμές που μεταβάλλονται ομαλά, δηλαδή να ελέγξουμε την σταθερότητα (stability) του ιδιοδιανύσματος. Για το σκοπό αυτό, κάνουμε ένα ιστόγραμμα των τιμών του και βρίσκουμε τον λόγο της μικρότερης ράβδου προς τη μεγαλύτερη. Θέτουμε ένα άνω όριο στην τιμή του λόγου αυτού. Στην παρούσα εργασία το όριο αυτό τέθηκε ίσο με 0.06.

Στη συνέχεια, εφόσον τηρείται το παραπάνω κριτήριο σταθερότητας και εφόσον η τιμή του  $N_{cut}$  δεν υπερβαίνει ένα προκαθορισμένο επίπεδο (στην παρούσα εργασία 0.08), τα βήματα 3-6 επαναλαμβάνονται για κάθε ένα από τα δύο μέρη.

## 2.4 Χωρικές συχνότητες

Η χωρική συχνότητα αναφέρεται στο επίπεδο των λεπτομερειών που υπάρχουν σε μια εικόνα ανά βαθμό οπτικής γωνίας (Hauser). Μια εικόνα, δηλαδή, με πολλές λεπτομέρειες και αρκετή ευκρίνεια θα έχει υψηλή τιμή συνολικής χωρικής συχνότητας σε σχέση με μια θολή εικόνα που θα έχει, αντίστοιχα, χαμηλή τιμή χωρικής συχνότητας.

Ο μαθηματικός υπολογισμός της χωρικής συχνότητας μιας εικόνας έχει ως εξής. Θεωρούμε μια εικόνα  $M \times N$  σε κλίμα του γκρι, όπου  $F(m, n)$  η τιμή της έντασης του εικονοστοιχείου που βρίσκεται στην  $m$  σειρά και  $n$  στήλη. Η τιμή αυτή της έντασης, είναι η ίδια με την τιμή που αναφέρθηκε στην παράγραφο 2.1  $F(i)$ , όπου εκεί το  $i$  παρίστανε τον αντίστοιχο κόμβο. Η χωρική συχνότητα της εικόνας αυτής ορίζεται ως εξής (Li, Kwok, & Wang, 2001):

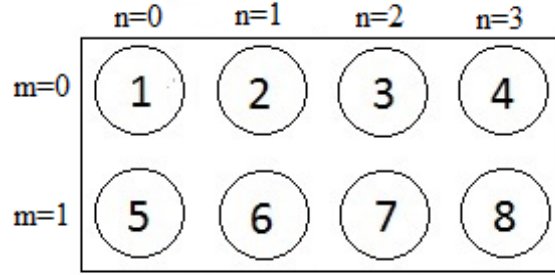
$$SF = \sqrt{(RF)^2 + (CF)^2} \quad (2.4)$$

όπου RF και CF η συχνότητα γραμμής (Row Frequency) και στήλης (Column Frequency), αντίστοιχα που δίνονται από τις εξής σχέσεις:

$$RF = \sqrt{\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} [F(m, n) - F(m, n-1)]^2} \quad (2.5)$$

$$CF = \sqrt{\frac{1}{MN} \sum_{n=0}^{N-1} \sum_{m=1}^{M-1} [F(m, n) - F(m-1, n)]^2} \quad (2.6)$$

Στο Σχήμα 2.3 φαίνεται μια εικόνα με τη μορφή γράφου  $\mathbf{G}(\mathbf{V}, \mathbf{E})$ , όπως αυτή που περιγράφηκε στο Σχήμα 2.1 με τις αριθμήσεις των γραμμών και των στηλών που απαιτούνται για τον υπολογισμό των συχνοτήτων RF και CF των Σχέσεων 2.5 και 2.6.



Σχήμα 2.3: Αναπαράσταση της εικόνας σε γράφο, με αριθμήσεις γραμμών και στηλών.

Για την εικόνα αυτή, κάνοντας εφαρμογή της Σχέσης 2.5, η συχνότητα γραμμής θα είναι:

$$\begin{aligned}
 RF &= \sqrt{\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} [F(m, n) - F(m, n-1)]^2} = \sqrt{\frac{1}{8} \sum_{m=0}^1 \sum_{n=1}^3 [F(m, n) - F(m, n-1)]^2} \\
 &= \sqrt{\frac{1}{8} [(F(2) - F(1))^2 + (F(3) - F(2))^2 + (F(4) - F(3))^2 + (F(6) - F(5))^2 + \\
 &\quad + (F(7) - F(6))^2 + (F(8) - F(7))^2]}
 \end{aligned}$$

Παρατηρούμε ότι η τιμή της έντασης  $F(0,1)$  (αρίθμηση γραμμής: 0, αρίθμηση στήλης: 1) ισοδυναμεί με το συμβολισμό  $F(2)$  (δεύτερος κόμβος κατά απόλυτη σειρά), όπως φαίνεται και στο Σχήμα 2.3.

Ομοίως, σύμφωνα με τη Σχέση 2.6 η συχνότητα στήλης θα είναι:

$$CF = \sqrt{\frac{1}{MN} \sum_{n=0}^{N-1} \sum_{m=1}^{M-1} [F(m, n) - F(m-1, n)]^2} = \sqrt{\frac{1}{8} \sum_{n=0}^3 \sum_{m=1}^1 [F(m, n) - F(m-1, n)]^2}$$

$$= \sqrt{\frac{1}{8} [(F(5) - F(1))^2 + (F(6) - F(2))^2 + (F(7) - F(3))^2 + (F(8) - F(4))^2]}$$

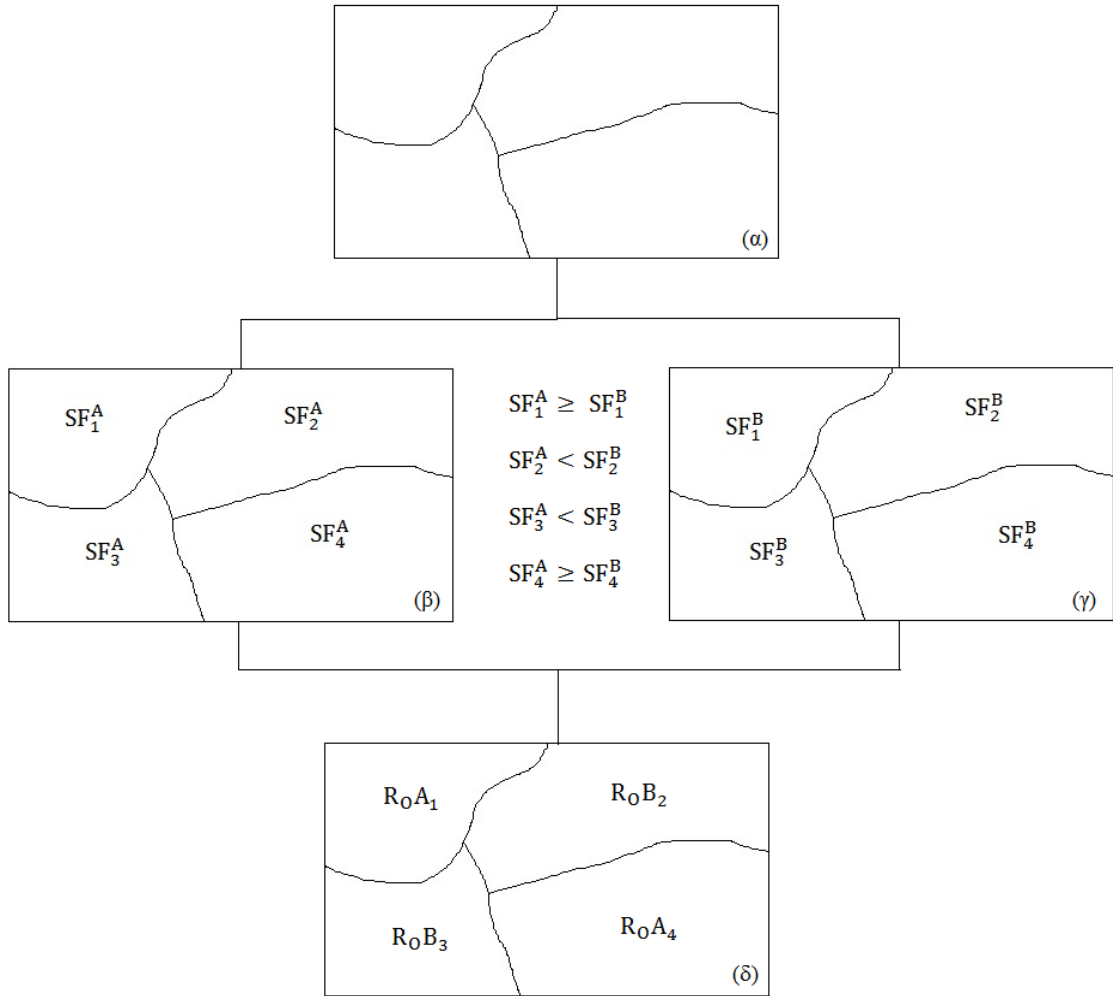
Η χωρική συχνότητα SF προκύπτει κάνοντας χρήση της Σχέσης 2.4.

## 2.5 Μίξη πολυεστιακών εικόνων

Στη συνέχεια, έχοντας δει τη μεθοδολογία για την κατάτμηση της εικόνας (παράγραφος 2.2), καθώς και την έννοια της χωρικής συχνότητας (παράγραφος 2.4), θα παρουσιάσουμε το συνδυασμό των δύο αυτών τεχνικών για να πάρουμε ως τελικό αποτέλεσμα μια εικόνα, που θα είναι η μίξη των αρχικών πολυεστιακών εικόνων.

Αρχικά, υποθέτουμε ότι έχουμε ολοκληρώσει τη μεθοδολογία της κατάτμησης του μέσου όρου των δύο αρχικών πολυεστιακών εικόνων. Ας θεωρήσουμε, λοιπόν, ότι οι περιοχές που προκύπτουν ύστερα από την κατάτμηση έχουν τη μορφή που φαίνεται στο Σχήμα 2.4(α). Στη συνέχεια χωρίζουμε τις αρχικές εικόνες (αυτές που εστιάζουν σε διαφορετικές περιοχές) σύμφωνα με τις περιοχές αυτές. Δηλαδή, στα Σχήματα 2.4(β) και 2.4(γ) φαίνεται η αναπαράσταση της κατάτμησης των αρχικών εικόνων ακριβώς με τον τρόπο που υποδεικνύει το Σχήμα 2.4(α). Στη συνέχεια, θεωρούμε κάθε περιοχή σαν μια μεμονωμένη εικόνα διαστάσεων  $M \times N$  (όπου φυσικά τα  $M, N$  διαφέρουν από περιοχή σε περιοχή) και για κάθε περιοχή της κάθε εικόνας, υπολογίζεται η χωρική συχνότητα σύμφωνα με τις Σχέσεις της παραγράφου 2.3.

Τέλος, συγκρίνουμε τις αντίστοιχες τιμές των χωρικών συχνοτήτων και επιλέγουμε την περιοχή εκείνης της εικόνας της οποίας η χωρική συχνότητα έχει μεγαλύτερη τιμή από τη χωρική συχνότητα της αντίστοιχης περιοχής της άλλης εικόνας. Στο Σχήμα 2.4 φαίνεται μια τυχαία σχέση χωρικών συχνοτήτων και το αποτέλεσμα της μίξης είναι αυτό που παρουσιάζεται στο Σχήμα 2.4(δ).



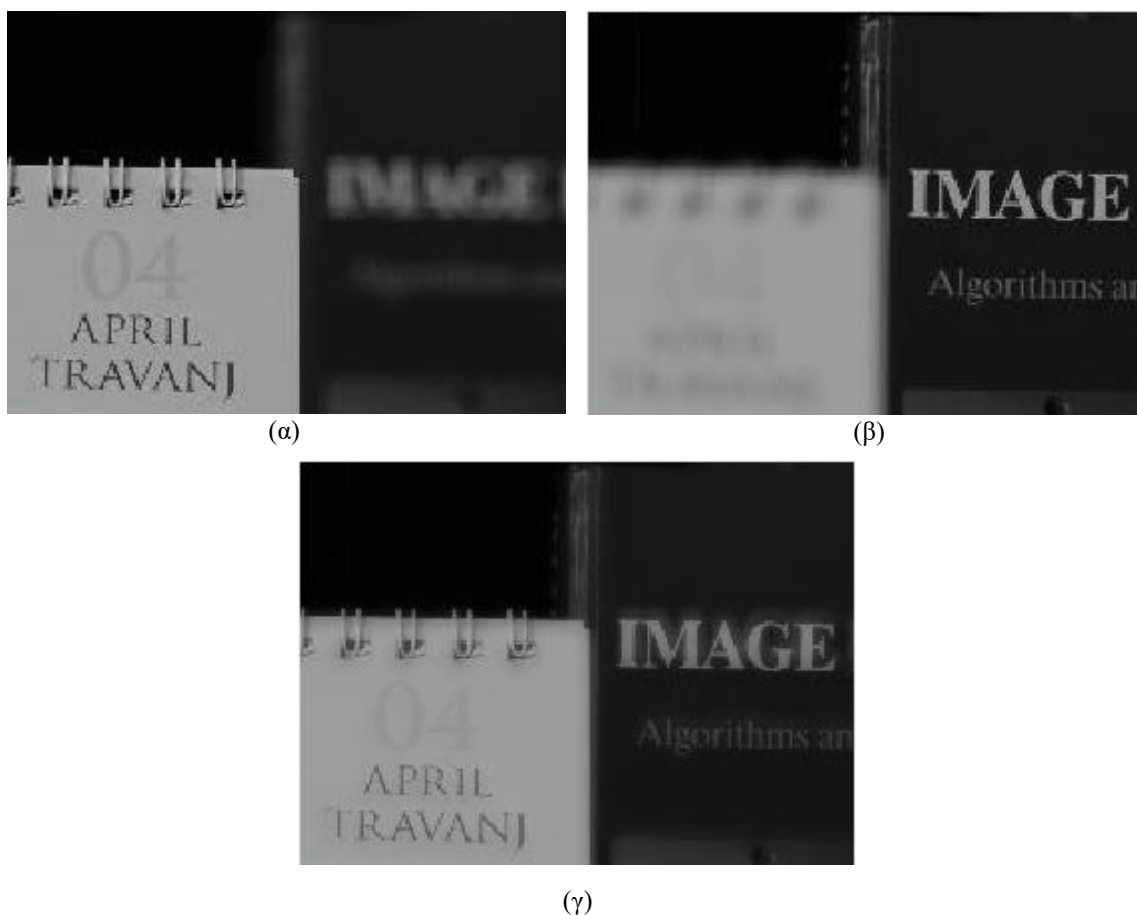
Σχήμα 2.4: Μίξη πολυεστιακών εικόνων: (α) Αποτέλεσμα κατάτμησης (μεθοδολογία της παραγράφου 2.2). (β), (γ) Οι αρχικές εικόνες χωρισμένες σύμφωνα με το αποτέλεσμα (α) και η εύρεση των χωρικών συχνοτήτων για κάθε περιοχή τους. (δ) Επιλογή περιοχής από την εικόνα A ή B, ανάλογα με το αποτέλεσμα της σύγκρισης των αντίστοιχων χωρικών συχνοτήτων και σύνθεση της νέας εικόνας.

## ΚΕΦΑΛΑΙΟ 3: Εφαρμογή της μεθόδου σε ζεύγη πολυεστιακών εικόνων

### 3.1 Αναλυτική παρουσίαση μίξης ενός ζεύγους πολυεστιακών εικόνων

Η μεθοδολογία του Κεφαλαίου 2 εφαρμόστηκε σε αρκετά ζεύγη εικόνων με στόχο, πάντα, να πάρουμε μια τελική εικόνα με μεγαλύτερη ευκρίνεια ανά περιοχή.

Στην Εικόνα 3.1(α) και 3.1(β) φαίνονται δείγματα δύο πολυεστιακών εικόνων. Οι αρχικές διαστάσεις των εικόνων ήταν 162x225 και έπρεπε να σμικρυνθούν, ώστε να μπορέσει το πρόγραμμα να ανταπεξέλθει στις μεγάλες διαστάσεις του πίνακα βαρών  $\mathbf{W}$ . Στη συγκεκριμένη περίπτωση, λοιπόν οι εικόνες προς επεξεργασία που εστιάζουν σε διαφορετικές περιοχές είχαν διαστάσεις 79x109. Στη συνέχεια, μετατρέψαμε τις εντάσεις  $\mathbf{F}(i)$  των δύο εικόνων σε κλίμακα από 0 έως 1 όπως υποδεικνύει το βήμα 1 της μεθοδολογίας της παραγράφου 2.2. Στην Εικόνα 3.1(γ) φαίνεται ο μέσος όρος των δύο αρχικών εικόνων (βήμα 2).



Εικόνα 3.1: (α), (β): Αρχικές εικόνες που εστιάζουν σε διαφορετικές περιοχές  
(γ): Μέσος όρος των αρχικών εικόνων

Στη συνέχεια υπολογίστηκε ο πίνακας βαρών  $\mathbf{W}$  όπως αναφέρεται στη Σχέση 2.1. Ο πίνακας αυτός θα έχει διαστάσεις  $(79 \times 109) \times (79 \times 109)$ , μία σειρά για τα βάρη του κάθε κόμβου. Στη Σχέση 2.1 οι παράμετροι  $\sigma_1$ ,  $\sigma_x$  και  $\gamma$  πήραν τις τιμές 0.05, 4 και 10 αντίστοιχα. Οι τιμές αυτές των παραμέτρων υιοθετήθηκαν και σε όλα τα επόμενα παραδείγματα του Κεφαλαίου 3.

Μετά τον υπολογισμό του διαγώνιου πίνακα  $\mathbf{D}$  (βήμα 3) και τη λύση της εξίσωσης των ιδιοτιμών (Σχέση 2.3), χρησιμοποιήσαμε τη δεύτερη στήλη του πίνακα των ιδιοδιανυσμάτων  $\mathbf{V}$  (η στήλη αυτή αντιστοιχεί στη δεύτερη μικρότερη ιδιοτιμή). Υπολογίσαμε 10 splitting points δηλαδή 10 τιμές που ισαπέχουν μεταξύ τους και μεταξύ της μικρότερης και μεγαλύτερης τιμής της δεύτερης στήλης του πίνακα  $\mathbf{V}$ . Για κάθε ένα splitting point χωρίσαμε όλους τους κόμβους της εικόνας σε δύο μέρη. Το πρώτο μέρος A είναι τα σημεία που αντιστοιχούν σε τιμές μικρότερες του splitting point και το δεύτερο μέρος B είναι τα σημεία που αντιστοιχούν σε τιμές μεγαλύτερες του splitting point. Για τον κάθε διαχωρισμό της εικόνας σε δύο μέρη A και B, υπολογίσαμε την τιμή του normalized cut που περιεγράφηκε στην παράγραφο 2.1 (Σχέση 2.2).

Στη συνέχεια, έγινε έλεγχος για το κριτήριο σταθερότητας. Έτσι φτιάξαμε το ιστόγραμμα των τιμών της δεύτερης στήλης του πίνακα  $\mathbf{V}$  και υπολογίσαμε το λόγο της μικρότερης τιμής στις ράβδους προς τη μεγαλύτερη.

Η κατάτμηση μιας περιοχής επιτράπηκε μόνο στις περιπτώσεις που η μικρότερη από τις 10 τιμές  $N_{cut}$  ήταν μικρότερη από 0.08 και ο λόγος από το ιστόγραμμα ήταν μικρότερος από 0.06. Η διαδικασία αυτή επαναλήφθηκε τόσες φορές όσες το επέτρεψαν οι παραπάνω συνθήκες σταθερότητας και μέγιστης τιμής  $N_{cut}$  (εδώ, 0.06).

Το αποτέλεσμα της κατάτμησης φαίνεται στην Εικόνα 3.2.

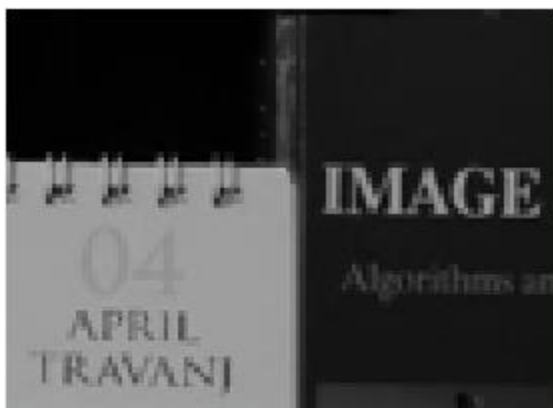


Εικόνα 3.2: Οι περιοχές της Εικόνας 3.1



Αφού βρήκαμε τις περιοχές της εικόνας, κόψαμε τις αρχικές εικόνες (Εικόνα 3.1(α) και 3.1(β)) σύμφωνα με τις περιοχές που φαίνονται στην Εικόνα 3.2. Για κάθε μία περιοχή και των δύο εικόνων υπολογίζουμε τις χωρικές συχνότητες (Σχέση 2.4). Έπειτα, για τη δημιουργία του τελικού αποτελέσματος, επιλέγουμε την περιοχή εκείνης της εικόνας για την οποία η χωρική συχνότητα είναι μεγαλύτερη από τη χωρική συχνότητα της αντίστοιχης περιοχής της άλλης εικόνας (δες Σχήμα 2.4).

Το αποτέλεσμα της μίξης των πολυεστιακών εικόνων με χρήση της μεθοδολογίας κατάτμησης περιοχών και χωρικών συχνοτήτων φαίνεται στην Εικόνα 3.3.



Εικόνα 3.3: Το αποτέλεσμα της μίξης των πολυεστιακών εικόνων

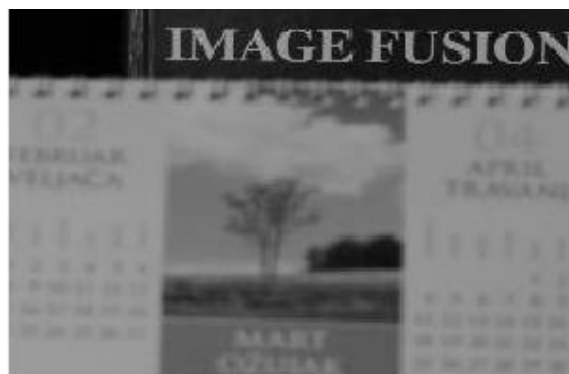
Σχολιάζοντας το αποτέλεσμα, βλέπουμε ότι η θολή περιοχή της αρχικής εικόνας του Σχήματος 3.1(α) είναι ευκρινής και ευανάγνωστη στο τελικό αποτέλεσμα της μίξης, και, αντίστοιχα, το ίδιο συμβαίνει για τη θολή περιοχή της αρχικής εικόνας του Σχήματος 3.1(β).

### **3.2 Συγκεντρωτικά αποτελέσματα για τα υπόλοιπα ζεύγη πολυεστιακών εικόνων**

Στη συνέχεια, παραθέτουμε συγκεντρωτικά διάφορα αποτελέσματα της μίξης πολυεστιακών εικόνων, παραθέτοντας επιπλέον τις περιοχές κατάτμησης.



(α)



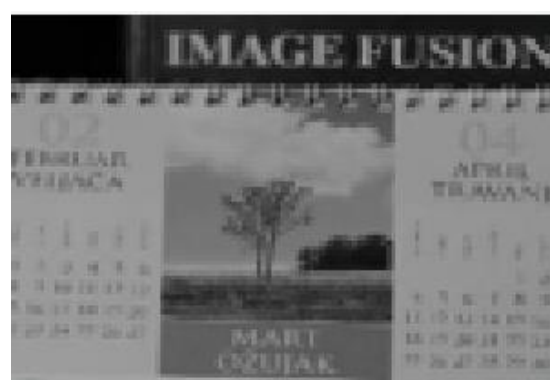
(β)



(γ)



(δ)

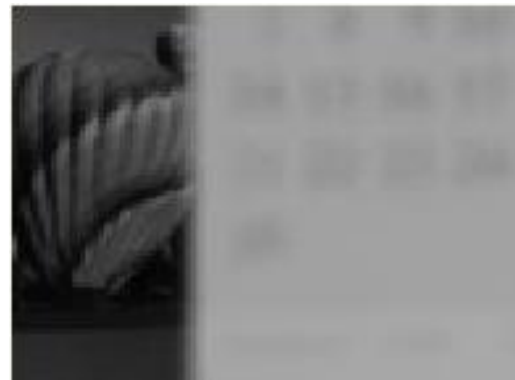


(ε)

Εικόνα 3.4: (α), (β) Αρχικές πολυεστιακές εικόνες διαστάσεων 81x124 (γ) Μέσος όρος των πολυεστιακών εικόνων (δ) Κατάτμηση σύμφωνα με normalized cuts όπου  $\sigma_1 = 0.05$ ,  $\sigma_x = 4$ ,  $r = 10$  (ε) Τελικό αποτέλεσμα ύστερα από χρήση χωρικών συχνοτήτων



(α)



(β)



(γ)



(δ)



(ε)

Εικόνα 3.5: (α), (β) Αρχικές πολυεστιακές εικόνες διαστάσεων 87x116 (γ) Μέσος όρος των πολυεστιακών εικόνων (δ) Κατάτμηση σύμφωνα με normalized cuts όπου  $\sigma_1 = 0.1$ ,  $\sigma_x = 4$ ,  $r = 10$  (ε) Τελικό αποτέλεσμα ύστερα από χρήση χωρικών συχνοτήτων



(α)



(β)



(γ)



(δ)



(ε)

Εικόνα 3.6 (α), (β) Αρχικές πολυεστιακές εικόνες διαστάσεων 102x99 (γ) Μέσος όρος των πολυεστιακών εικόνων (δ) Κατάτμηση σύμφωνα με normalized cuts όπου  $\sigma_1 = 0.05$ ,  $\sigma_x = 4$ ,  $r = 10$  (ε) Τελικό αποτέλεσμα ύστερα από χρήση χωρικών συχνοτήτων

## ΚΕΦΑΛΑΙΟ 4: Συμπεράσματα

Σκοπός της παρούσας εργασίας είναι η μελέτη και η εφαρμογή μιας τεχνικής επεξεργασίας εικόνων που εστιάζουν σε διαφορετικά μέρη με απώτερο σκοπό η τελική εικόνα της μίξης να έχει τις καλύτερα εστιασμένες περιοχές από κάθε εικόνα. Αρχικά, οι πολυεστιακές εικόνες συνδυάστηκαν χρησιμοποιώντας το μέσο όρο τους. Στη συνέχεια ο μέσος όρος χωρίστηκε σε περιοχές με χρήση της μεθόδου των *normalized cuts*. Κατόπιν, οι αρχικές εικόνες χωρίστηκαν σε περιοχές βάσει του αποτελέσματος κατάτμησης του μέσου όρου. Τέλος, έγινε σύγκριση των αντίστοιχων περιοχών των εικόνων πηγής σύμφωνα με τις χωρικές τους συχνότητες και επιλέχθηκε η περιοχή εκείνης της εικόνας με τη μεγαλύτερη τιμή χωρικής συχνότητας. Η τεχνική εφαρμόστηκε σε διάφορα ζεύγη πολυεστιακών εικόνων προκειμένου να επαληθευτεί η αποδοτικότητα της.

Τα αποτελέσματα της παρούσας εργασίας δείχνουν ότι η μέθοδος κατάτμησης πολυεστιακών εικόνων με *normalized cuts* και χωρικές συχνότητες δουλεύει ικανοποιητικά· η τελική εικόνα έχει καλύτερη ευκρίνεια από τον απλό μέσο όρο και σίγουρα είναι καλύτερη από τις περιοχές των αρχικών εικόνων που δεν εστίαζαν καλά. Παρόλα αυτά, κάποια από τα αποτελέσματα δεν είναι σε συμφωνία με αυτά της βιβλιογραφίας. Η Εικόνα 3.6, π.χ., έχει χωριστεί σε οχτώ περιοχές (Εικόνα 3.6 (δ)), σε αντίθεση με τους Li και Yang (2008) οι οποίοι χρησιμοποίησαν τον ίδιο αλγόριθμο και στην ίδια απεικόνιση βρήκαν 13 περιοχές. Πιστεύουμε ότι ο λόγος της διαφωνίας μπορεί να οφείλεται είτε στη διαφορετική επιλογή των παραμέτρων  $\sigma_i$  και  $\sigma_x$  είτε σε πιθανή διαφορετική επιλογή του ορίου για την ελάχιστη τιμή του  $N_{cut}$ . Συγκριμένα, με την επιλογή τιμών 0.1 και 0.3 των Li και Yang για τα  $\sigma_i$  και  $\sigma_x$  αντίστοιχα, καταλήγαμε μόνο σε δύο περιοχές και έτσι επιλέξαμε να μείνουμε σταθεροί στις τιμές 0.05 και 4 αντίστοιχως. Όσον αφορά το όριο της ελάχιστης τιμής του  $N_{cut}$ , δεν αναφέρεται στους Li και Yang και έτσι δεν μπορούμε να σχολιάσουμε συγκριτικά.

Στην παρούσα εργασία, οι παράμετροι  $\sigma_i$  και  $\sigma_x$  είχαν σταθερές τιμές για όλες τις εικόνες (0.05 και 4 αντίστοιχα). Πιστεύουμε, παρόλα αυτά, ότι θα έπρεπε να υπολογίζονται για κάθε απεικόνιση ξεχωριστά. Αυτή αποτελεί μία από τις προτάσεις για βελτίωση του παρόντος αλγορίθμου στο μέλλον.

Επιπλέον, η μέθοδος δυστυχώς βασιζόταν σε περιορισμένο πλήθος εικονοστοιχείων (μέγιστο 10000) όταν η διαθέσιμη μνήμη στον υπολογιστή ήταν 8GB RAM. Οι αρχικές εικόνες, δηλαδή, έπρεπε να σμικρυνθούν με την εντολή *resize* ώστε στο τέλος οι υπό επεξεργασία εικόνες να έχουν πλήθος εικονοστοιχείων 10000. Αυτό οφείλεται στο μέγεθος

του πίνακα  $\mathbf{W}$ . Για 10000 εικονοστοιχεία, ο πίνακας  $\mathbf{W}$  είχε διαστάσεις 10000x10000. Πιστεύουμε ότι ο αλγόριθμος θα ήταν πιο αποδοτικός εάν ο πίνακας βαρών  $\mathbf{W}$ , μπορούσε να αντιπροσωπευτεί μόνο από το άνω τμήμα του, αφού είναι εξ' ορισμού συμμετρικός. Παρόλα αυτά, σε αυτήν την περίπτωση η επίλυση της εξίσωσης ιδιοτιμών θα ήταν αδύνατη. Αυτό αποτελεί μια ακόμη πρόταση για μελλοντική βελτίωση του αλγορίθμου.

Ακόμη, αξίζει να αναφέρουμε ότι η μέθοδος θα μπορούσε να εφαρμοστεί σε αντίγραφα εικόνων που εστιάζουν σε περισσότερες από δύο διαφορετικές περιοχές. Παρόμοια αποτελέσματα, π.χ. για εικόνες που εστιάζουν σε τρεις περιοχές, παρουσιάζονται στους Li και Yang.

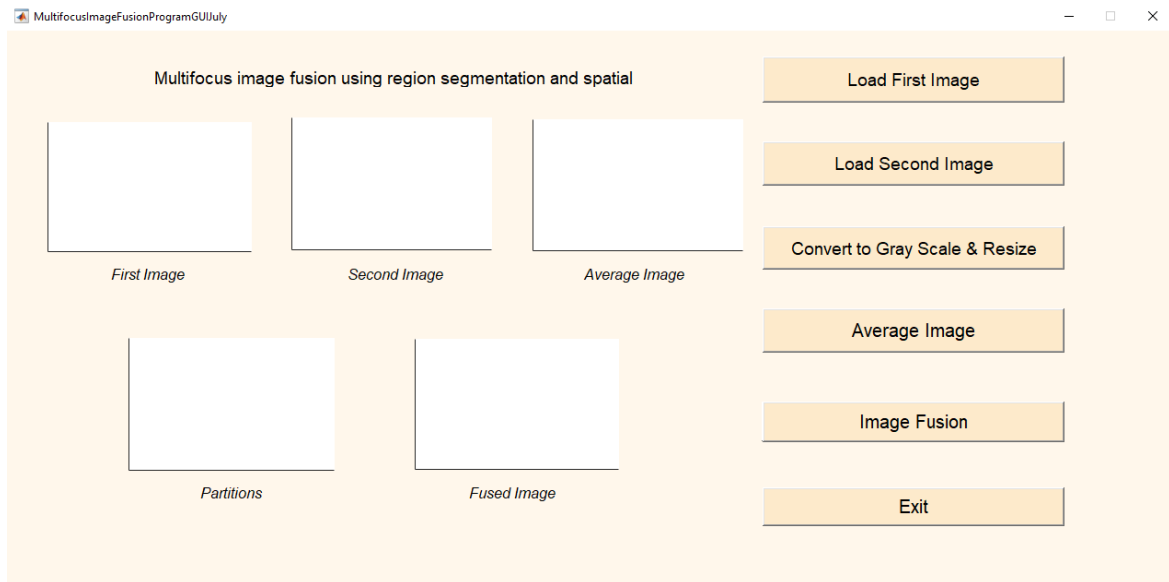
Τέλος, ενδιαφέρον θα αποτελούσε η εφαρμογή του αλγορίθμου και σε έγχρωμες πολυεστιακές εικόνες. Θυμίζουμε ότι στον αλγόριθμο που υλοποιήθηκε στην παρούσα εργασία οι αρχικά έγχρωμες απεικονίσεις μετατρέπονταν πρώτα σε κλίμακα του γκρι και στη συνέχεια ο αλγόριθμος εφαρμοζόταν στις εικόνες κλίμακας του γκρι.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., & Ogden, J. M. (1984). Pyramid methods in image processing. *RCA Engineer*, 29, σσ. 33-41.
- Gonzalez, R. C., & Woods, R. E. (2011). *Ψηφιακή Επεξεργασία Εικόνας* (3η εκδ.). Θεσσαλονίκη: Τζιόλα.
- Hauser, L. (n.d.). *University of Calgary*. Ανάκτηση Σεπτέμβριος 11, 2016, από <http://www.ucalgary.ca/pip369/mod4/spatial/frequency1>
- Huang, W., & Jing, Z. (2007). Multi-focus image fusion using pulse coupled neural network. *Pattern Recognition Letters*, 28, σσ. 1123-1132.
- Li, S., & Yang, B. (2008). Multifocus image fusion using region segmentation and spatial frequency. *Image and Vision Computing*, 26(7), σσ. 971-979.
- Li, S., Kwok, J. T., & Wang, Y. (2001). Combination of images with diverse focuses using the spatial frequency. *Information Fusion*, 2(3), σσ. 169-176.
- Pu, T., & Ni, G. (2000). Contrast-based image fusion using the discrete wavelet transform. *Optical Engineering*, 39, σσ. 2075-2082.
- Shi, J., & Malik, J. (1998). Motion Segmentation and Tracking Using Normalized Cuts. *International Conference on Computer Vision*, (σσ. 1154-1160).
- Shi, J., & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 888-905.
- Wang, Z., Ma, Y., & Gu, J. (2010). Multi-focus image fusion using PCNN. *Pattern Recognition*, 43, σσ. 2003-2016.

## ΠΑΡΑΡΤΗΜΑ 1: Εφαρμογή του αλγορίθμου με Matlab Graphical User Interface

Σε αυτή την ενότητα θα σχολιάσουμε τη λειτουργία των πλήκτρων που χρησιμοποιούμε στο Graphical User Interface που φαίνεται στην παρακάτω Εικόνα.



Τα πρώτα δύο πλήκτρα *Load First Image* και *Load Second Image* φορτώνουν και δείχνουν τις δυο αρχικές εικόνες στους άξονες *First Image* και *Second Image*, σύμφωνα με τις απλές εντολές του Matlab:

```
[imdirectionP1,user_cancel]=imgetfile();  
P1=imread(imdirectionP1);  
axes(handles.axes1);  
imshow(P1);  
  
[imdirectionP2,user_cancel]=imgetfile();  
P2=imread(imdirectionP2);  
axes(handles.axes2);  
imshow(P2);
```



Το πλήκτρο *Convert to Gray Scale & Reseize* μετατρέπει τις αρχικές εικόνες σε απόχρωση γκρι και κάνει τη σμίκρυνση με την εντολή `resize` έτσι ώστε το γινόμενο των δύο διαστάσεων να μην υπερβαίνει τα 10000 εικονοστοιχεία:

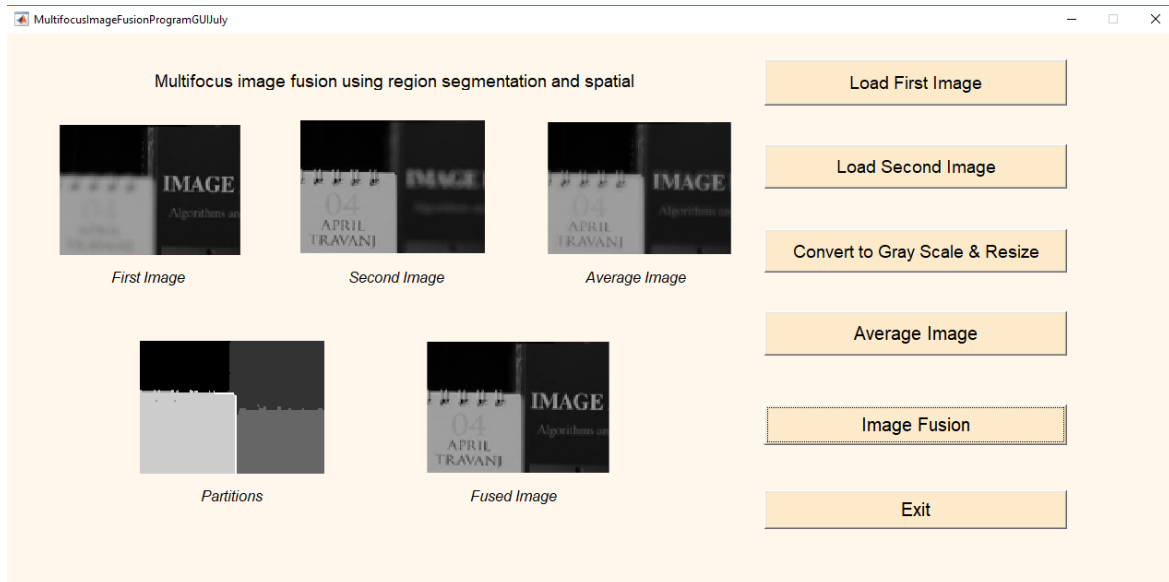
```
P1=rgb2gray(P1); % Metatropi apo rgb se gray tw n arxikwn
eikonwn
P2=rgb2gray(P2);
[a,b]=size(P1);
if a*b>10000
    P1=imresize(P1,(sqrt(10000/(a*b)))); %Smikrinsi tw n
eikonon gia na min einai poly megalos o pinakas varon W
    P2=imresize(P2,(sqrt(10000/(a*b))));
end
I1=im2double(P1); % Metatropi apo uint8 se double me klimaka
apo 0-1 (Shi Malik, sel. 894)
I2=im2double(P2);
axes(handles.axes1);
imshow(I1);
axes(handles.axes2);
imshow(I2);
```

Το πλήκτρο *Average Image* υπολογίζει το μέσο όρο των δύο αρχικών εικόνων και τον εμφανίζει στους ομόνυμους άξονες:

```
TMI=(I1+I2)/2;
axes(handles.axes3);
imshow(TMI);
```

Το πλήκτρο *Image Fusion* κόβει την εικόνα σε περιοχές σύμφωνα με τη μεθοδολογία των *normalized cuts* και κάνει χρήση των χωρικών συχνοτήτων για την τελική μίξη των εικόνων. Με το πλήκτρο αυτό καλείται η συνάρτηση *Total\_Cut\_Image* η οποία με τη σειρά της καλεί την αναδρομική συνάρτηση *Cut\_Image*. Τα αποτελέσματα παρουσιάζονται στους άξονες *Partitions* και *Fused Image*. Στο πρώτο απεικονίζονται οι κατατμήσεις της εικόνας του

μέσου όρου όπου κάθε περιοχή χρωματίζεται με διαφορετική απόχρωση του γκρι ενώ στο δεύτερο απεικονίζεται το τελικό αποτέλεσμα της μίξης των δύο πολυεστιακών εικόνων. Οι δύο κώδικες, *Total\_Cut\_Image* και *Cut\_Image* παρουσιάζονται αναλυτικά στο Παράρτημα 2.



## ΠΑΡΑΡΤΗΜΑ 2: Κώδικες Matlab

### 1. Graphical User Interface

```
function                                varargout                                =
MultifocusImageFusionProgramGUIJuly(varargin)
%   MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY   MATLAB   code   for
MultifocusImageFusionProgramGUIJuly.fig
%   MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY, by itself, creates
a new MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY or raises the
existing
%   singleton*.
%
%   H = MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY returns the
handle to a new MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY or the
handle to
%   the existing singleton*.
%
%
MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY('CALLBACK',hObject,event
Data,handles,...) calls the local
%   function named CALLBACK in
MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY.M with the given input
arguments.
%
%
MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY('Property','Value',...)
creates a new MULTIFOCUSIMAGEFUSIONPROGRAMGUIJULY or raises
the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before
MultifocusImageFusionProgramGUIJuly_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes
property application
%   stop. All inputs are passed to
MultifocusImageFusionProgramGUIJuly_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
MultifocusImageFusionProgramGUIJuly

% Last Modified by GUIDE v2.5 07-Aug-2017 21:26:04
```

```

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @MultifocusImageFusionProgramGUIJuly_OpeningFcn, ...
                  'gui_OutputFcn',  @MultifocusImageFusionProgramGUIJuly_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before MultifocusImageFusionProgramGUIJuly
is made visible.
function
MultifocusImageFusionProgramGUIJuly_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
MultifocusImageFusionProgramGUIJuly (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes MultifocusImageFusionProgramGUIJuly wait for
user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout =
MultifocusImageFusionProgramGUIJuly_OutputFcn(hObject,
eventdata, handles)

```

```

% varargin    cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Loadlim.
function Loadlim_Callback(hObject, eventdata, handles)
% hObject    handle to Loadlim (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global P1
[imdirectionP1,user_cancel]=imgetfile();
if user_cancel
    msgbox(sprintf('Error'),'Error','Error');
    return
end
P1=imread(imdirectionP1);
axes(handles.axes1);
imshow(P1);

% --- Executes on button press in Load2im.
function Load2im_Callback(hObject, eventdata, handles)
% hObject    handle to Load2im (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global P2 imdirectionP2
[imdirectionP2,user_cancel]=imgetfile();
if user_cancel
    msgbox(sprintf('Error'),'Error','Error');
    return
end
P2=imread(imdirectionP2);
axes(handles.axes2);
imshow(P2);

% --- Executes on button press in GrayScale.
function GrayScale_Callback(hObject, eventdata, handles)
% hObject    handle to GrayScale (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global P1 P2 I1 I2

```

```

P1=rgb2gray(P1); % Metatropi apo rgb se gray tw n arxikwn
eikonwn
P2=rgb2gray(P2);
[a,b]=size(P1);
if a*b>10000
    P1=imresize(P1,(sqrt(10000/(a*b)))); %Smikrinsi tw n
eikonon gia na min einai poly megalos o pinakas varon W
    P2=imresize(P2,(sqrt(10000/(a*b))));
end
I1=im2double(P1); % Metatropi apo uint8 se double me klimaka
apo 0-1 (Shi Malik, sel. 894)
I2=im2double(P2);
axes(handles.axes1);
imshow(I1);
axes(handles.axes2);
imshow(I2);

% --- Executes on button press in Average.
function Average_Callback(hObject, eventdata, handles)
% hObject handle to Average (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
global I1 I2 TMI
TMI=(I1+I2)/2;
axes(handles.axes3);
imshow(TMI);

% --- Executes on button press in fusion.
function fusion_Callback(hObject, eventdata, handles)
global imdirectionP2 imdirection I1 I2 TMI
[y,tmi,tmi_final] = Total_Cut_Image(I1,I2,TMI);
axes(handles.axes4);
imshow(tmi);
axes(handles.axes5);
imshow(tmi_final);

% --- Executes on button press in Exit.
function Exit_Callback(hObject, eventdata, handles)
% hObject handle to Exit (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
msgbox('Thanks for using Multifocus Image fusion
Program','Message',20)
%pause(2);
close all
close();
close();

```

## 2. Συνάρτηση Total\_Cut\_Image:

```
function [y,tmi,tmi_final] = Total_Cut_Image(I1,I2,TMI)

[a,b]=size(TMI);

% Dimiourgia enos voithitikou pinaka intelmatrix me
diastaseis a*b,4 (a kai
% b oi diastaseis tis eikonas). I proti stili periexei
akeraious arithmous.
% Idioi arithmoi antistoixoun se pixel pou anikoun sto idio
partiion.
% I deuteri stili periexei tis times tou gray
% scale. I triti kai tetarti stili periexoun tis
syntetagmenes x kai y
% (grammi kai stili) tou kathe simiou sthn eikona TMI.
% Sto eksis oles oi epeksergasies tha ginontai me vasi
tetoious pinakes
% opos o intelmatrix.
intelmatrix=zeros(a*b,4);
intelmatrix(:,1) = 1;
intelmatrix(:,2)=reshape(TMI', [],1);
for x=1:a
    for y=1:b
        intelmatrix(b*(x-1)+y,3)=x;
        intelmatrix(b*(x-1)+y,4)=y;
    end
end
iteration = 1;
[y] = Cut_Image(intelmatrix,iteration);

%% Show the partitions in grey shades %%
un = unique(y(:,1));
s = length(un);
strogg=floor(255/(s-1));
grey_shades=0:strogg:255;
y_to_show = y;
for i = 1:s
    y_to_show(find(y_to_show(:,1) == un(i)), 2) =
grey_shades(i);

end
tmi = reshape(y_to_show(:,2), [b,a])' ;
tmi=uint8(tmi);

%%Spatial Frequencies %%

%H proti stili tou y(:,4) periexei arnitikes times oi opoies
symvolizoun
%tis katatmiseis. Ola ta stoixia pou exoun idia arnitiki
timi, anikoun sto
```

```

%idio partition. Ti stili auti ti metatrepoume stis
diastaseis tw n arxikwn
% eikonon (a x b). O pinakas autos einai o Partitions.

Partitions = reshape (y(:,1), b,a)';
I = zeros(a,b,2); % Oi arxikes polyestiakes eikones I1 kai
I2 ensomatomenes se enan pinaka.
I(:,:,1) = I1; %H proti polyestiaki eikona
I(:,:,2) = I2; %H deuteri polyestiaki eikona
RF = zeros(2,s); %Row frequencies
CF = zeros(2,s); %Column frequencies
for image = 1:2
    for i=1:s
        [m,n] = find(Partitions ==un(i)); % Theseis (m,n)
        tw n pixel pou anikoun sto idio partition
        %Ta parkato akolouthoun to tipo RF (Li & Yang)
        for j = min(m) : max(m)
            for k = min(n) +1 : max(n)
                % Sinthikes: a) Ta geitonika pixel na
                anoikoun sto idio
                % Partition b) Akomi ki an einai idia, tha
                prepei na
                % anikoun sto partition un(i)
                if (Partitions(j,k) == Partitions(j,k-1) )
                    && (Partitions(j,k) == un(i) )
                        RF (image, i) = RF(image,i) +
                        (I(j,k,image) - I(j,k-1,image))^2;
                    end
                end
            end
        end
        M = max(m) - min(m) +1; %Plithos seiron tou
        partition i
        N = max(n) - min(n) +1; %Plithos stilon tou
        partition i
        RF(image,i) = sqrt( RF(image,i)/(M*N) );
    end
end

%Omoios gia to CF...
for image = 1:2
    for i=1:s
        [m,n] = find(Partitions ==un(i));
        for k = min(n) : max(n)
            for j = min(m) +1 : max(m)
                if (Partitions(j,k) == Partitions(j-1,k) )
                    && (Partitions(j,k) == un(i) )
                        CF (image, i) = CF(image,i) +
                        (I(j,k,image) - I(j-1,k,image))^2;
                    end
                end
            end
        end
    end
end

```



```

        end
        M = max(m) - min(m) +1;
        N = max(n) - min(n) +1;
        CF(image,i) = sqrt( CF(image,i)/(M*N) );
    end
end
% Spatial Frequency
SF = sqrt(RF.^2 + CF.^2);

%Teliki miksarismeni eikona:
tmi_final= zeros(a,b);
for i=1:s
    [m,n] = find(Partitions ==un(i));
    if SF(1,i) >= SF(2,i)
        for t = 1: length(m)
            tmi_final(m(t) ,n(t)) = I1(m(t) ,n(t));
        end
    else
        for t = 1: length(m)
            tmi_final(m(t) ,n(t)) = I2(m(t) ,n(t));
        end
    end
end
end

tmi_final = round(tmi_final*256);
tmi_final= uint8(tmi_final);
end

```

3. Cut\_Image, αναδρομική συνάρτηση που καλείται από την προηγούμενη κύρια συνάρτηση Total\_Cut\_Image:

```
function [y] = Cut_Image(intelmatrix,iteration)

% I proti stili tou intelmatrix, periexei akeraious
arithmetic pou antistoixoun
% sta partitions (pou exou dimiourgithei eos auti ti
stigma).
% P.x. An i proti stili periexei 1 kai 2 auto shmainei
% oti mexri stigmis h eikona exei kopei se dyo meri (to
proto meros einai oi
% seires tou intelmatrix pou exoun 1 stin proti stili kai to
deutero meros einai
% oi seires tou intelmatrix pou exoun 2 stin proti stili).
An oi akeraioi
% einai arititikoι, shmainei oti to sygkekrimeno partition de
tha ksanakopei
% (logo kapoiou treshold). H anadromiki synartisi Cut_Image
kovei panta
% to partition pou antistoixei sto megalytero thetiko
akeraio.

si = 0.05; %Shi Malik
sx = 4; %Shi Malik
r = 10; %apostasi r pano apo ton pixel pano apo tin opoia to
w midenizetai
threshold = 0.1; % ratio threshold (Shi Malik)
number_of_partitions = max(intelmatrix(:,1)); %Poios einai o
megalyteros akeraios.

if number_of_partitions >0 % Tha kopsei mono an o
megalyteros akeraios einai thetikos

    A = find (intelmatrix(:,1)== number_of_partitions) ; %
    Oi antistoixes theseis ston

                                                                    %
intelmatrix
    if length(A) > 100 %Tha kopsei mono an to sygkekrimeno
partition den einai poly mikro

        %To parakato dimiourgei ton symmetriko pinaka varon
w
w=zeros (length(A),length(A));
for ki=1:length(A)
    for kj=ki:length(A)
        d=sqrt( (intelmatrix(A(ki),3)-
intelmatrix(A(kj),3))^2+(intelmatrix(A(ki),4)-
intelmatrix(A(kj),4))^2 );
        if d>r
```

```

        w(ki,kj) = 0;
    else
        F=exp(-((intelmatrix(A(ki),2)-
intelmatrix(A(kj),2))^2)/(si^2));
        X=exp(-(d^2)/(sx^2));
        w(ki,kj)=F*X;
        w(kj,ki)=F*X; %O pinakas varon einai
symmetrikos
    end
end
end

%Diagonios pinakas D pou periexei ta athrismata ton
seiron tou w
%(Shi, Malik)
D=zeros(length(A),length(A));
SumofW=sum(w,2); % athrizei orizontia tis grammes
tou W
for n=1:length(A)
    D(n,n)=SumofW(n);
end

% Evresi idiodianismaton (pinakas V) kai idiotimon
(pinakas L)

[V,L]=eigs((D-w),D,2,'sm'); % Lynei thn eksisosi
idiotimon kratontas
                                % mono tis dyo
mikroteres ('sm')
                                % idiotimes kai ta
antistoixa
                                % idiodianysmata
%To parakato vriskei ti megaliteri apo tis dyo
mikroteres idiotimes:
second_smallest = find(diag(L) == max(diag(L)));

maxv=max(V(:,second_smallest)); %Megisti timi tou
idiodianismatos
minv=min(V(:,second_smallest)); %Elaxisti timi tou
idiodianismatos
pithana_partitions=ones(length(A),10); % pithana
partitionas simfona me 10 pithana splitting points

%To parakato vriskei gia kathe pithano kopsimo, mia
timi ncut symfona
%me ti methodology tou normalized cuts.
ncut=ones(10,1);
for f=1:10; %10 epanalipsisei gia kathe splitting
point
    splitting_point=minv+((maxv-minv)/11)*f; % f:
splitting point

```

```

        pithana_partitions(:,f)=(V(:,second_smallest) <
splitting_point);% ean isxeiei i synthiki tote opou isxeiei
bale 1 allivw bake 0
        cutA_B=0;
        cutB_B=0;
        cutA_A=0;
        for ki=1:length(A)
            for li=ki:length(A)
                if
(pithana_partitions(ki,f)==1)&&(pithana_partitions(li,f)==1)
;
                    cutB_B=cutB_B+w(ki,li);
                elseif
(pithana_partitions(ki,f)==0)&&(pithana_partitions(li,f)==0)
;
                    cutA_A=cutA_A+w(ki,li);
                else
                    cutA_B=cutA_B+w(ki,li);
                end
            end
        end
        end

ncut(f)=(cutA_B/(cutA_B+cutA_A))+(cutA_B/(cutB_B+cutA_B));
        end

        %To parakato ypologizei ena apo ta kritiria gia na
pragmatopoiithei
        %to kopsimo (ratio<threshold).
        bins=hist(V(:,second_smallest)); %Istogramma timon
tis deuteris stilis tou V.
        ratio=min(bins)/max(bins); %Auto einai to threshold.
        if ratio<threshold && min(ncut(:))<0.08
%synthiki gia na ksanakopsei
        % Vriskei to mikrotero ncut.
        %a: I thesi (ara kai i stili tou pithana
partitions pou tha xrisimopoisoume) pou vrisketai to
mikrotero ncut
        a=find(ncut==min(ncut(:)));
        m=(pithana_partitions(:,a(1))==1); % m: oi
theseis sto DIANISMA A pou antistoixoun sto proto meros tou
kopsimatos
        o=(pithana_partitions(:,a(1))==0); % o: oi
theseis sto DIANISMA A pou antistoixoun sto deutero meros
tou kopsimatos

        %Oi theseis A tha moirastoun se dyo partitions.
Kathe
        %partition tha lavei enan akeraio.
        intelmatrix (A(m),1) = number_of_partitions; %
A(m): oi theseis tou partition ston intelmatrix.
        intelmatrix (A(o),1) = number_of_partitions+1;

```

```

        iteration = iteration + 1;
        intelmatrix = Cut_Image(intelmatrix,iteration);
%Anadromi

    else
        intelmatrix(A,1) = -iteration; %Ean den kopei
tote stis theseis A topotheteitai enas arnitikos arithmos
        %I timi tou
arnitikou den exei
        %tosi simasia.
        iteration = iteration + 1;
        intelmatrix = Cut_Image(intelmatrix,iteration);

    end
else
    intelmatrix(A,1) = -iteration;%Ean den kopei tote
stis theseis A topotheteitai enas arnitikos arithmos
        %I timi tou arnitikou
den exei
        %tosi simasia.
        iteration = iteration + 1;
        intelmatrix = Cut_Image(intelmatrix,iteration);
end
y =intelmatrix;
end
y =intelmatrix;

```