



Σχολή Μηχανικών

**Τμήμα Μηχανικών Πληροφορικής Υπολογιστών
και Τηλεπικοινωνιών**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανίχνευση εξεχουσών περιοχών εικόνων
βασισμένη σε ολική αντίθεση**

Αντώνιος Πατσιούρας

A.E.M.: 2159

Επιβλέπων καθηγητής: Αθανάσιος Νικολαΐδης

2021

Πίνακας Περιχομένων

Πρόλογος.....	ΣΕΛ.3
Εισαγωγή.....	ΣΕΛ.4
Σχετικές Εργασίες.....	ΣΕΛ.7
Παραγωγή τιμών εξεχουσών.....	ΣΕΛ.9
περιοχών μέσω αντίθεσης βασισμένης σε ιστόγραμμα	
Παραγωγή τιμών εξεχουσών.....	ΣΕΛ.13
περιοχών μέσω αντίθεσης βασισμένης σε περιοχές	
Παρουσίαση εφαρμογής.....	ΣΕΛ.17
Πειραματικές συγκρίσεις.....	ΣΕΛ.21
Επίλογος και μελλοντικές εργασίες.....	ΣΕΛ.24
Βιβλιογραφία.....	ΣΕΛ.25
Παράρτημα.....	ΣΕΛ.28

Πρόλογος

Στην παρούσα εργασία παρουσιάζεται η μελέτη και υλοποίηση δύο μεθόδων υπολογισμού εξεχουσών περιοχών μιας εικόνας μέσω κατάλληλης επεξεργασίας χωρίς να απαιτείται προηγούμενη γνώση του περιεχόμενου της εικόνας.

Αυτό είναι πολύ σημαντικό σε πολλές εφαρμογές όπως η κατάτμηση εικόνας ή η αναγνώριση αντικειμένων ή η προσαρμοστική συμπίεση εικόνας.

Παρουσιάζεται μια μέθοδος υπολογισμού των εξεχουσών περιοχών μιας εικόνας με βάση την αντίθεση ανάμεσα σε περιοχές της εικόνας η οποία ταυτόχρονα υπολογίζει τις διαφορές αντίθεσης και τη χωρική συνοχή σε ολόκληρη την εικόνα.

Ο αλγόριθμος είναι απλός, αποδοτικός και παράγει χάρτες των εξεχουσών περιοχών με πλήρη ανάλυση. Ο αλγόριθμος παρέχει αποτελέσματα μεγαλύτερης ακρίβειας και καλύτερα ποσοστά ανάκλησης από παλιότερες μεθόδους χρησιμοποιώντας ένα από τα πιο γνωστά δημοσίως διαθέσιμα σύνολα δεδομένων.

Επίσης παρουσιάζεται πώς ο χάρτης των εξεχουσών περιοχών μπορεί να χρησιμοποιηθεί για την παραγωγή υψηλής ποιότητας μάσκας κατάτμησης της εικόνας για μεταγενέστερη επεξεργασία της εικόνας.

Εισαγωγή

Οι άνθρωποι συνήθως χωρίς πολύ κόπο κρίνουν το πόσο σημαντική είναι μια περιοχή της εικόνας και επικεντρώνουν την προσοχή τους στα σημαντικά μέρη της.

Η υπολογιστική ανίχνευση τέτοιων εξεχουσών περιοχών παραμένει ένας σημαντικός στόχος καθώς επιτρέπει την προνομιακή κατανομή των υπολογιστικών πόρων και στη μετέπειτα ανάλυση και σύνθεση της εικόνας.

Οι εξέχουσες περιοχές προέρχονται από την οπτική μοναδικότητα ή σπανιότητα και αποδίδονται σε μεταβολές στα χαρακτηριστικά της εικόνας όπως για παράδειγμα το χρώμα, οι ακμές και τα όρια.

Η οπτική ανίχνευση εξεχουσών περιοχών σχετίζεται αρκετά με το πώς ο άνθρωπος αντιλαμβάνεται κι επεξεργάζεται τα οπτικά ερεθίσματα, κάτι που διερευνάται από πολλούς κλάδους συμπεριλαμβανόμενων της γνωστικής ψυχολογίας και τη νευροβιολογίας.

Διάφορες θεωρίες σχετικά με την ανθρώπινη οπτική εστίαση υποθέτουν ότι η ανθρώπινη όραση επεξεργάζεται μόνο κάποια μέρη μιας εικόνας με λεπτομέρεια ενώ τα υπόλοιπα δεν τα επεξεργάζεται σχεδόν καθόλου.

Ένα πρώιμο έργο των Treisman και Gelade[25], Koch και Ullman[17] αλλά και μεταγενέστερες θεωρίες για την ανθρώπινη εστίαση προτάθηκαν από τους Itti [15], Wolfe [26] και άλλους.

Αυτοί προτείνουν δυο τρόπους οπτικής εστίασης: 1. γρήγορη, από κάτω προς τα πάνω οδηγούμενη από τα δεδομένα και 2. πιο αργή από πάνω προς τα κάτω εξαρτώμενη από τη διεργασία ανίχνευσης εξεχουσών περιοχών οδηγούμενη από το στόχο.

Στην παρούσα εργασία εστιάζουμε στην οδηγούμενη από τα δεδομένα ανίχνευση εξεχουσών περιοχών από κάτω προς τα πάνω χρησιμοποιώντας την αντίθεση της εικόνας.

Προτείνεται μια ανάλυση της αντίθεσης για εξαγωγή οδηγούμενη από τα δεδομένα υψηλής ανάλυσης βασισμένη στις ακόλουθες παρατηρήσεις:

- Μια μέθοδος βασισμένη στην ολική αντίθεση που διαχωρίζει ένα αντικείμενο μεγάλης κλίμακας από τα τριγύρω του είναι προτιμότερη από μεθόδους τοπικής αντίθεσης που παράγουν υψηλές τιμές για τις εξέχουσες περιοχές κοντά ή ακριβώς πάνω στην περίμετρο του αντικείμενου.

- Καθολικές εκτιμήσεις επιτρέπουν τη σύγκριση τιμών των εξεχουσών περιοχών που ανήκουν σε όμοιες περιοχές και μπορούν να αναδείξουν ομοιόμορφα τα αντικείμενα που περιέχονται στις συγκεκριμένες περιοχές.

- Οι εξέχουσες περιοχές μιας εικόνας εξαρτώνται κυρίως από την αντίθεση των γειτονικών περιοχών, όσο οι περιοχές είναι πιο μακριά από την περιοχή που βρισκόμαστε τόσο πιο ασήμαντες είναι οι τιμές της αντίθεσής τους.

- Οι χάρτες των εξεχουσών περιοχών πρέπει να είναι γρήγοροι κι εύκολοι στην κατασκευή τους προκειμένου να μπορούμε να επεξεργαστούμε μεγάλου μεγέθους εικόνες και να διευκολύνουμε αποτελεσματικά την ταξινόμηση και την ανάκτησή τους.

Προτείνεται μια μέθοδος ανίχνευσης εξεχουσών περιοχών χρησιμοποιώντας αντίθεση εικόνας βασισμένη σε ιστόγραμμα (HC).

Οι τιμές των εξεχουσών περιοχών στους συγκεκριμένους χάρτες ανατίθενται ανά pixel με βάση το χρωματικό διαχωρισμό του pixel από τα υπόλοιπα pixels της εικόνας για την παραγωγή χαρτών εξεχουσών περιοχών πλήρους ανάλυσης.

Χρησιμοποιείται ιστόγραμμα για αποδοτική επεξεργασία ενώ παράλληλα εφαρμόζεται και μια διαδικασία εξομάλυνσης για τον έλεγχο της κβαντοποίησης της εικόνας.

Σημειώνεται ότι ο αλγόριθμος μπορεί να μην παρέχει αξιόπιστα αποτελέσματα για εικόνες με πλούσιο φόντο γύρω από το αντικείμενο ενδιαφέροντος.

Ως βελτίωση των HC χαρτών μπορούμε να ενσωματώσουμε χωρικές σχέσεις για την παραγωγή χαρτών εξεχουσών περιοχών που χρησιμοποιούν αντίθεση βασισμένη σε περιοχές (RC). Έτσι πρώτα διαχωρίζουμε την εικόνα σε περιοχές και μετά εκχωρούμε τιμές για τις εξεχουσες περιοχές ανά περιοχή.

Η τιμή της εξεχουσας περιοχής μιας εικόνας υπολογίζεται χρησιμοποιώντας μια συνολική αντίθεση που υπολογίζεται από την αντίθεση της περιοχής και των χωρικών αποστάσεων των άλλων περιοχών της εικόνας.

Σχετικές Εργασίες

Στην παρούσα εργασία επικεντρωνόμαστε σε βιβλιογραφία σχετική με την ανίχνευση εξεχουσών περιοχών από κάτω προς τα πάνω η οποία μπορεί να είναι πιο κοντά στην ανθρώπινη αντίληψη ή πιο καθαρά υπολογιστική διαδικασία ή και τα δύο μαζί.

Οι μέθοδοι αυτού του τύπου χρησιμοποιούν χαμηλού επιπέδου επεξεργασία για να ορίσουν την αντίθεση των περιοχών της εικόνας χρησιμοποιώντας χαρακτηριστικά όπως η ένταση, το χρώμα και τα ακμές.

Οι μέθοδοι τοπικής αντίθεσης ερευνούν τη σπανιότητα των περιοχών της εικόνας με “σεβασμό” στις (μικρές) γειτονικές περιοχές.

Βασίζονται στο μοντέλο “early representation” το οποίο επηρεάζεται αρκετά από την ανθρώπινη αντίληψη. Αυτό το μοντέλο κατασκευάστηκε από τους Koch και Ullman[17].

Οι Itti και Koch [15] ορίζουν τις εξέχουσες περιοχές χρησιμοποιώντας τις διαφορές του κέντρου της εικόνας με την υπόλοιπη εικόνα και αυτό εφαρμόζεται σε πολλαπλά επίπεδα των εικόνων.

Οι Ma and Zhang [19] προτείνουν μια εναλλακτική μέθοδο τοπικής αντίθεσης για παραγωγή χαρτών εξεχουσών περιοχών οι οποίοι ελεγκτούνται χρησιμοποιώντας ένα ασαφές μοντέλο ανάπτυξης.

Οι Harel and Koch [12] κανονικοποιούν το μοντέλο των Ittie και Koch με σκοπό να τονιστούν περισσότερο τα εμφανή μέρη και να συνδυαστούν με άλλους χάρτες σημαντικών μερών μιας εικόνας.

Οι Liu, Yan, Sun και άλλοι [18] βρίσκουν μια τιμή αντίθεσης πολλών επιπέδων με γραμμικό συνδυασμό αντιθέσεων σε μια Gaussian πυραμίδα μιας εικόνας.

Τέτοιες μέθοδοι που χρησιμοποιούν τοπική αντίθεση έχουν την τάση να παράγουν υψηλότερες τιμές εξεχουσών περιοχών κοντά

στα όρια των αντικειμένων αντί να τα τονίζουν ομοιόμορφα.

Οι μέθοδοι καθολικής αντίθεσης υπολογίζουν τις εξέχουσες περιοχές της εικόνας χρησιμοποιώντας την αντίθεση της περιοχής με “σεβασμό” στην υπόλοιπη εικόνα.

Οι Zhai & Shah [29] ορίζουν την τιμή της εξέχουσας περιοχής σε επίπεδο pixel υπολογίζοντας την τιμή της αντίθεσης σε όλα τα υπόλοιπα pixels.

Ωστόσο για περισσότερη αποδοτικότητα μετατρέπουν την εικόνα από RGB σε Lab και χρησιμοποιούν μόνο τη φωτεινότητα που είναι η πρώτη συνιστώσα από την εικόνα σε μορφή Lab. Έτσι όμως δε λαμβάνονται υπόψη οι τιμές των άλλων συνιστωσών-καναλιών.

Οι Achanta και άλλοι [2] προτείνουν μια μέθοδο με βάση τη συχνότητα η οποία ορίζει απευθείας τις τιμές των εξεχουσών περιοχών ανά pixel

χρησιμοποιώντας τις χρωματικές διαφορές από το χρωματικό μέσο όρο της εικόνας.

Παραγωγή τιμών εξεχουσών περιοχών μέσω αντίθεσης βασισμένης σε ιστόγραμμα

Με βάση την παρατήρηση ότι η ανθρώπινη όραση είναι ευαίσθητη σε οπτικά σήματα με αντίθεση, προτείνεται η μέθοδος αντίθεσης βασισμένης σε ιστόγραμμα η οποία ορίζει τις τιμές των εξεχουσών περιοχών για κάθε pixel μιας εικόνας χρησιμοποιώντας στατιστικά των χρωμάτων της εικόνας.

Ειδικά η τιμή της εξέχουσας περιοχής ενός pixel ορίζεται χρησιμοποιώντας την αντίθεσή του σε σχέση με τα υπόλοιπα pixels της εικόνας.

Ο μαθηματικός τύπος είναι:

$$S(I_k) = \sum_{\forall I_i \in I} D(I_k, I_i) \quad \mathbf{(1)}$$

Όπου $D(I_k, I_i)$ είναι η χρωματική απόσταση μεταξύ των pixels I_k, I_i στο Lab χρωματικό κώδικα. Μπορεί να επεκταθεί και στον εξής τύπο:

$$S(I_k) = D(I_k, I_1) + D(I_k, I_2) + \dots + D(I_k, I_N) \quad \mathbf{(2)}$$

Όπου το N είναι ο συνολικός αριθμός των pixels. Είναι εύκολο να παρατηρηθεί ότι τα pixels με την ίδια τιμή χρώματος έχουν την ίδια τιμή εξέχουσας περιοχής.

Σύμφωνα με την παραδοχή αυτή και δεδομένου ότι η μέτρηση αυτή δεν έχει κάποια σημασία για τις χωρικές σχέσεις μπορούμε να αλλάξουμε τον τύπο υπολογισμού της τιμής της εξέχουσας περιοχής ομαδοποιώντας τους όρους που έχουν το ίδιο χρώμα.

Ο τύπος υπολογισμού γίνεται ως εξής:

$$S(i_k)=S(c_i)=\sum_{j=1}^n f_j D(c_i, c_j) \quad (3)$$

Όπου το c_i είναι η τιμή του χρώματος του pixel I_k , n είναι ο αριθμός των διαφορετικών χρωμάτων των pixels και f_j είναι η συχνότητα του χρώματος του pixel c_j στην εικόνα I . Σημειώνεται ότι για να προστατέψουμε τα στατιστικά των χρωμάτων μιας εξέχουσας περιοχής από πιθανή καταστροφή από παρόμοια χρώματα από άλλες περιοχές μπορούμε να κατασκευάσουμε ένα σύστημα χρησιμοποιώντας διάφορες μάσκες πάνω σε μια εικόνα.

Αναβάθμιση της ταχύτητας του αλγόριθμου:
Υπολογίζοντας τις τιμές των εξεχουσών περιοχών για κάθε pixel της εικόνας σύμφωνα με τον τύπο (1) ο χρόνος που χρειάζεται είναι $O(N^2)$, κάτι που είναι υπολογιστικά πολύ ακριβό ακόμα και για μεσαίου μεγέθους εικόνες. Ο τύπος (3) χρειάζεται χρόνο $O(N)+O(n^2)$ που μπορεί να βελτιωθεί σε $O(N)$ εάν $O(n^2) \leq O(N)$. Γι' αυτό ο τρόπος για να βελτιώσουμε την ταχύτητα είναι να μειώσουμε τον αριθμό των χρωμάτων των pixel της εικόνας.

Οι Zhai και Shah [29] μειώνουν τον “ n ” αριθμό των χρωμάτων χρησιμοποιώντας μόνο τη φωτεινότητα. Σ' αυτή την περίπτωση $n^2 = 256^2 \ll N$. Ωστόσο η μέθοδός τους έχει το μειονέκτημα να αγνοούνται οι τιμές των άλλων καναλιών.

Σ' αυτή την εργασία χρησιμοποιούμε όλες τις συνιστώσες κι όχι μόνο την πρώτη (φωτεινότητα).

Για να μειώσουμε τον αριθμό των χρωμάτων πρώτα κβαντίζουμε κάθε χρωματική συνιστώσα έτσι ώστε να έχει 12 διαφορετικές τιμές κάτι που μειώνει τον αριθμό των χρωμάτων σε $12^3 = 1728$.

Θεωρώντας ότι ένα χρώμα σε μια εικόνα τυπικά καλύπτει ένα μικρό τμήμα του συνόλου των pixels, μειώνουμε ακόμα περισσότερο τον αριθμό των χρωμάτων απορρίπτοντας τα λιγότερο χρησιμοποιούμενα χρώματα.

Επιλέγουμε τα πιο συχνά χρησιμοποιούμενα χρώματα και ελέγχουμε αν καλύπτουν τουλάχιστον το 95% των pixels της εικόνας. Τα χρώματα που απορρίφθηκαν αντικαθίστανται με τα πλησιέστερα χρώματα στο ιστόγραμμα.

Εξομάλυνση εικόνας: Ωστόσο η κβαντοποίηση δημιουργεί κάποιον “θόρυβο”. Μερικά παρόμοια χρώματα μπορεί να κβαντιστούν σε διαφορετικές τιμές. Για να μειωθούν τα παράσιτα που δημιουργήθηκαν προηγουμένως χρησιμοποιούμε μια διαδικασία εξομάλυνσης για να βελτιώσουμε τα αποτελέσματα των τιμών των εξεχουσών περιοχών για κάθε χρώμα.

Αντικαθιστούμε την τιμή της εξέχουσας περιοχής κάθε χρώματος με το μέσο όρο των τιμών των εξεχουσών περιοχών των παρόμοιων χρωμάτων (υπολογισμένο με χρωματική απόσταση στο Lab).

Τυπικά επιλέγουμε $m=n/4$ πλησιέστερα χρώματα για να ορίσουμε την τιμή της εξέχουσας περιοχής του χρώματος c με τον εξής τύπο:

$$S'(c) = \frac{1}{(m-1)T} \sum_{i=1}^m (TD(c, c_i) S(c_i)) \quad (4)$$

Όπου $T = \sum_{i=1}^m D(c, c_i)$ είναι το σύνολο των αποστάσεων μεταξύ του χρώματος c και των m γειτόνων c_i κι ο παράγοντας κανονικοποίησης υπολογίζεται με τον εξής τύπο:

$$\sum_{i=1}^m T - D(c, c_i) = (m-1)T$$

Σημειώνεται ότι χρησιμοποιείται ένας γραμμικά μεταβαλλόμενος παράγοντας εξομάλυνσης ($T - D(c, c_i)$) για να εκχωρηθούν μεγαλύτερα βάρη στα χρώματα που είναι πιο κοντά στο χρώμα c .

Σε πειράματα βρέθηκε ότι τέτοια γραμμικά μεταβαλλόμενα βάρη είναι καλύτερα από Gaussian βάρη τα οποία μειώνονται πολύ απότομα.

Σημειώνεται ότι οι ράβδοι του ιστογράμματος είναι πιο κοντά μεταξύ τους μετά από την εξομάλυνση δείχνοντας τα παρόμοια χρώματα να βρίσκονται πιο κοντά το ένα με το άλλο, κάτι το οποίο μειώνει τον θόρυβο.

Λεπτομέρειες εκτέλεσης αλγόριθμου: Για να κβαντιστεί ο χρωματικός κώδικας σε 12^3 διαφορετικά χρώματα διαιρούμε ομοιόμορφα κάθε χρωματικό κανάλι σε 12 διαφορετικά επίπεδα. Ενώ τα χρώματα κβαντίζονται στον χρωματικό χώρο RGB, υπολογίζουμε τις χρωματικές διαφορές στο Lab λόγω μεγαλύτερης ακρίβειας. Με πειράματα παρατηρήθηκε ότι ο θόρυβος της κβαντοποίησης στον χώρο Lab είναι μεγαλύτερος. Τα καλύτερα αποτελέσματα πετυχαίνονται με κβαντοποίηση στον RGB χρωματικό χώρο ενώ παράλληλα υπολογίζουμε την απόσταση στον Lab.

Παραγωγή τιμών εξεχουσών περιοχών μέσω αντίθεσης βασισμένης σε περιοχές

Οι άνθρωποι δίνουν περισσότερη προσοχή στις περιοχές που έχουν μεγαλύτερη αντίθεση από το τριγύρω περιβάλλον τους [8].

Εκτός από την αντίθεση οι χωρικές σχέσεις παίζουν σημαντικό ρόλο στην ανθρώπινη προσοχή.

Η υψηλή αντίθεση της περιοχής σε σχέση με τον τριγύρω της χώρο συνήθως είναι πιο δυνατό στοιχείο για την τιμή της εξέχουσας περιοχής παρά η υψηλή τιμή αντίθεσης σε σχέση με τις πιο μακρινές περιοχές.

Δεδομένου ότι η απευθείας εισαγωγή χωρικών σχέσεων κατά τη διάρκεια υπολογισμού της αντίθεσης του pixel είναι υπολογιστικά πολύ δαπανηρή, υλοποιούμε μια μέθοδο με την οποία ενσωματώνονται οι χωρικές σχέσεις σε ένα υπολογισμό της αντίθεσης ανά περιοχή.

Σ' αυτή τη μέθοδο πρώτα διαχωρίζουμε την εικόνα σε περιοχές, μετά υπολογίζουμε την αντίθεση της κάθε περιοχής και ορίζουμε την τιμή εξέχουσας περιοχής για κάθε περιοχή ως το σταθμισμένο σύνολο των αντιθέσεων της περιοχής σε σχέση με όλες τις άλλες περιοχές της εικόνας.

Τα βάρη αποδίδονται σύμφωνα με τις χωρικές αποστάσεις, με τις πιο μακρινές περιοχές να έχουν πιο μικρά βάρη.

Αντίθεση περιοχής μέσω αραιού ιστογράμματος:
 Πρώτα διαχωρίζουμε την εικόνα σε περιοχές χρησιμοποιώντας τη μέθοδο των Felzenszwalb και Huttenlocher η οποία δέχεται σαν είσοδο ένα διάγραμμα [9]. Η αρχικοποίηση των στοιχείων της εικόνας (pixels) γίνεται με τυχαίο τρόπο κάθε φορά. Πιο συγκεκριμένα ταξινομούνται τα στοιχεία και για κάθε στοιχείο υπάρχουν και αντίστοιχες τιμές “βάρους” που υπολογίζονται κάθε φορά που εκτελείται ο αλγόριθμος από τις οποίες καθορίζεται η ομοιότητά του με όλα τα γειτονικά του στοιχεία (pixels). Αν η ομοιότητα με κάποια από τα γειτονικά στοιχεία (pixels) είναι χαμηλή (υψηλές τιμές βάρους) σημαίνει ότι δεν ανήκουν στις ίδιες περιοχές και διαχωρίζονται σε ξεχωριστές περιοχές (regions). Επιπλέον χρησιμοποιείται κι ένα κατώφλι για τον καθορισμό της ανομοιότητας μεταξύ των δυο στοιχείων το οποίο δεν είναι σταθερό γιατί προκύπτει από την τιμή βάρους του κάθε στοιχείου (pixel) σε συνδυασμό με κάποια σταθερή τιμή που καθορίζεται από το χρήστη. Μετά υπολογίζουμε το ιστόγραμμα της κάθε περιοχής όπως στο προηγούμενο κεφάλαιο.

Η τιμή της εξέχουσας περιοχής για κάθε περιοχή r_k υπολογίζεται από τη μέτρηση της αντίθεσης με όλες τις άλλες περιοχές της εικόνας.

Ο τύπος είναι:

$$S(r_k) = \sum_{r_i \neq r_k} w(r_i) D_r(r_k, r_i) \quad (5)$$

Όπου $w(r_i)$ είναι το βάρος της περιοχής r_i και $D_r(., .)$ είναι η χρωματική απόσταση μεταξύ των δυο περιοχών.

Στην παρούσα φάση χρησιμοποιούμε τον αριθμό των pixels της περιοχής σαν τη τιμή του βάρους της για να έχουμε μεγαλύτερη τιμή αντίθεσης στις μεγαλύτερες περιοχές. Η χρωματική απόσταση μεταξύ δυο περιοχών ορίζεται ως εξής:

$$D(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i}, c_{2,j}) \quad (6)$$

Όπου $f(c_{k,i})$ είναι η συχνότητα του i -th χρώματος ανάμεσα σε όλα τα n_k χρώματα στην k -th περιοχή r_k όπου $k = \{1, 2\}$.

Η χρωματική συχνότητα υπολογίζεται για κάθε χρώμα της k -th περιοχής και προκύπτει από τον αριθμό των στοιχείων (pixels) που έχουν το ίδιο χρώμα.

Σημειώνεται ότι χρησιμοποιείται η συχνότητα του χρώματος στην περιοχή σαν βάρος του χρώματος προκειμένου να εντοπίσουμε περισσότερες διαφορές μεταξύ των κυρίαρχων χρωμάτων.

Χωρικά σταθμισμένη αντίθεση περιοχής:

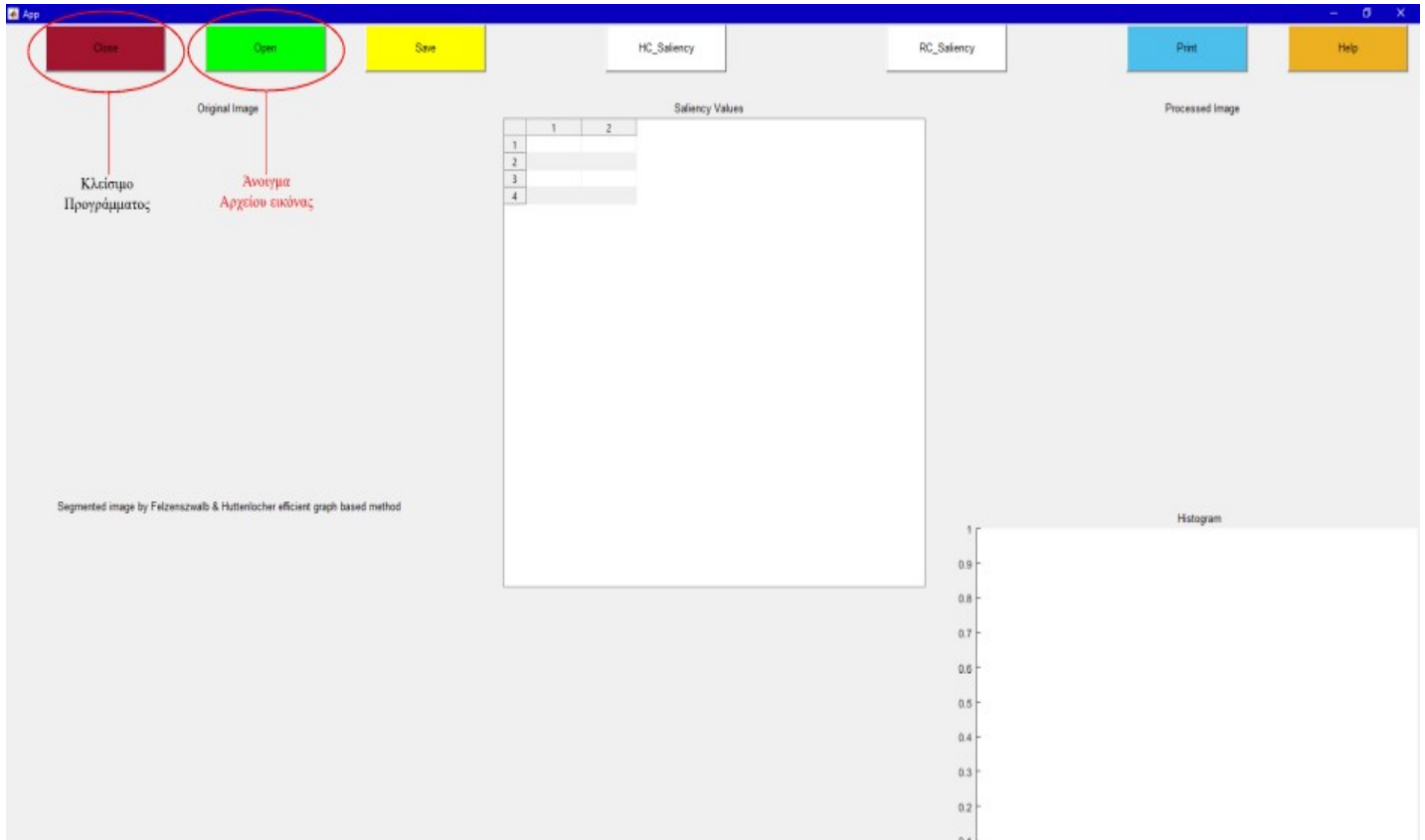
Επιπλέον συμπεριλαμβάνουμε χωρικές πληροφορίες χρησιμοποιώντας μια χωρική στάθμιση στον τύπο (5) προκειμένου να αυξήσουμε τις τιμές για τις κοντινότερες περιοχές και να μειώσουμε τις τιμές για τις μακρινότερες. Ο τύπος είναι:

$$S(r_k) = \sum_{r_i \neq r_k} \exp(-D_s(r_k, r_i)/\sigma_s^2) w(r_i) D_r(r_k, r_i) \quad (7)$$

Όπου $D_s(r_k, r_i)$ είναι η χωρική απόσταση μεταξύ των περιοχών r_k, r_i και το σ_s ελέγχει την ένταση της χωρικής στάθμισης. Μεγαλύτερες τιμές του σ μειώνουν τις τιμές της χωρικής στάθμισης επειδή η αντίθεση των μακρινών περιοχών θα επηρεάζει περισσότερο την τιμή εξέχουσας περιοχής της συγκεκριμένης περιοχής.

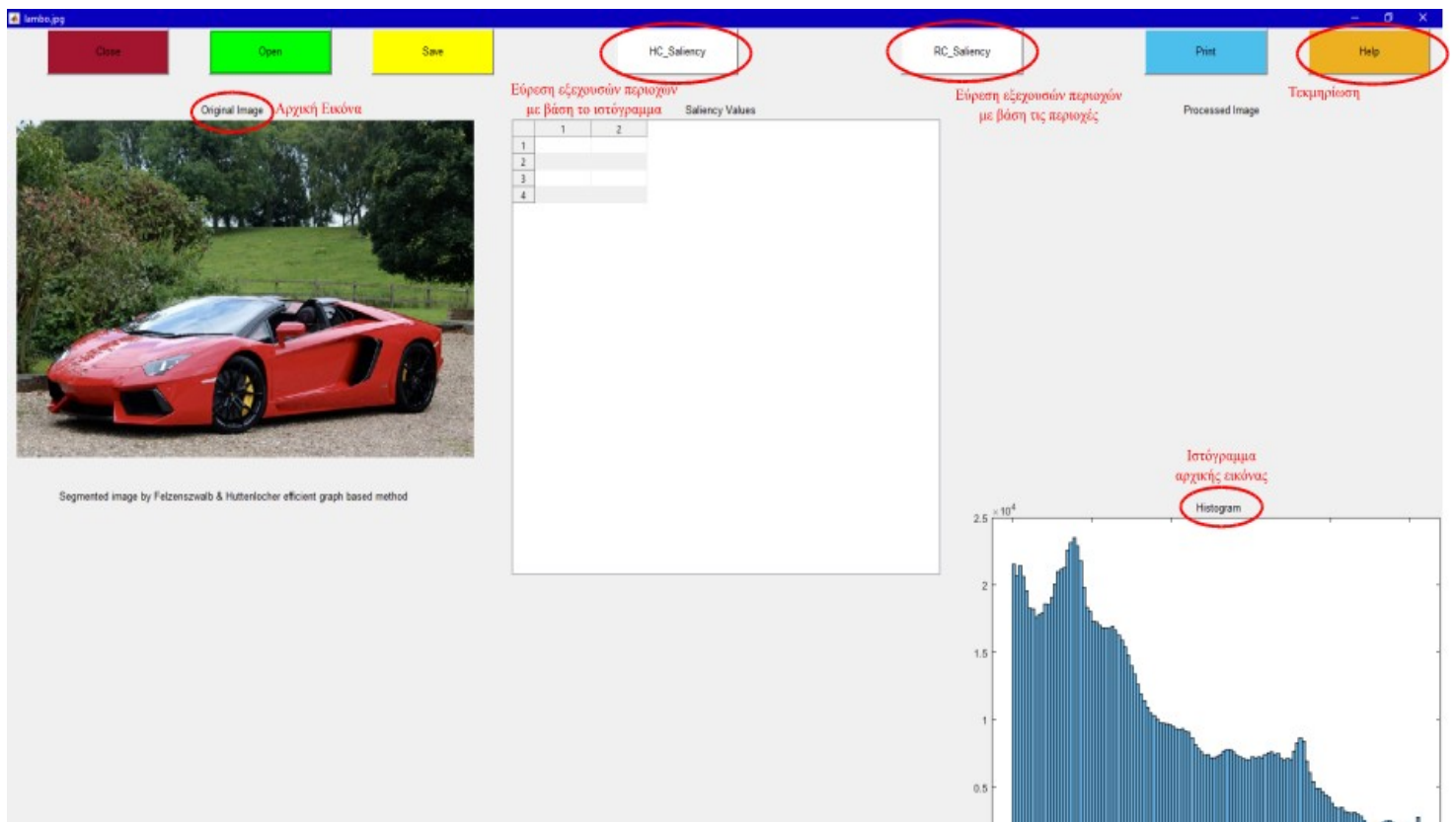
Η χωρική απόσταση μεταξύ δυο περιοχών ορίζεται ως η Ευκλείδεια απόσταση μεταξύ των κέντρων των δυο περιοχών. Εδώ η τιμή του $\sigma^2 = 0.4$ και οι συντεταγμένες των pixels κανονικοποιούνται στο διάστημα $[0,1]$.

Παρουσίαση εφαρμογής



Στην παραπάνω εικόνα φαίνεται η εφαρμογή όταν ανοίγει για πρώτη φορά. Τα κουμπιά στο επάνω μέρος του παράθυρου με σειρά από αριστερά προς τα δεξιά είναι ως εξής:

- 1) Κλείσιμο εφαρμογής
- 2) Άνοιγμα αρχείου εικόνας
- 3) Αποθήκευση επεξεργασμένης εικόνας
- 4) Πρώτη μέθοδος (HC) εύρεσης εξεχουσών περιοχών
- 5) Δεύτερη μέθοδος (RC) εύρεσης εξεχουσών περιοχών
- 6) Εκτύπωση επεξεργασμένης εικόνας
- 7) Οδηγίες χρήσης εφαρμογής



Αρχικά πατάμε με το ποντίκι στο πράσινο κουμπί για να επιλέξουμε την εικόνα που επιθυμούμε για επεξεργασία.

Επάνω αριστερά όπως φαίνεται και στην παραπάνω φωτογραφία φαίνεται η αρχική εικόνα που θα επεξεργαστεί η εφαρμογή.

Κάτω δεξιά φαίνεται το ιστόγραμμα της αρχικής εικόνας.

Έπειτα επιλέγουμε ανάλογα με τη μέθοδο που επιθυμούμε να γίνει η επεξεργασία της αρχικής εικόνας και πατάμε ένα από τα δυο γκριζα κουμπιά που βρίσκονται στη μέση του επάνω μέρους του παράθυρου της εφαρμογής.

Close Open Save HC_Saliency RC_Saliency Εκτύπωση Print Help

Original Image

Αποθήκευση επεξεργασμένης εικόνας

Τιμές εξεχουσών περιοχών

Saliency Values

row	283	284	285	286	287	288	289
211	06	3.3716e+06	3.3716e+06	3.1247e+06	3.4787e+06	3.4787e+06	3.4787e+06
212	06	3.3716e+06	3.1247e+06	3.4787e+06	3.4787e+06	3.4787e+06	3.4787e+06
213	06	3.1247e+06	3.1708e+06	3.4787e+06	3.4787e+06	3.1708e+06	3.1247e+06
214	06	3.1708e+06	3.1708e+06	3.1708e+06	3.4787e+06	3.1708e+06	3.1247e+06
215	06	3.1708e+06	3.1708e+06	3.4787e+06	3.4787e+06	3.4787e+06	3.1708e+06
216	06	3.1708e+06	3.4787e+06	3.4787e+06	3.1708e+06	3.4787e+06	3.1708e+06
217	06	3.1708e+06	3.4787e+06	3.4787e+06	3.1708e+06	3.1708e+06	3.4787e+06
218	06	3.4787e+06	3.1708e+06	3.1708e+06	3.4787e+06	3.4787e+06	3.4787e+06
219	06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.4787e+06	4.5766e+06
220	06	3.4787e+06	3.4787e+06	3.4787e+06	3.4787e+06	3.4787e+06	3.1708e+06
221	06	3.1708e+06	3.4787e+06	3.4787e+06	3.1708e+06	3.1708e+06	3.1708e+06
222	06	3.4787e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1247e+06
223	06	3.1708e+06	3.1247e+06	3.1247e+06	3.1708e+06	3.4787e+06	3.1708e+06
224	06	3.1708e+06	3.1708e+06	3.1247e+06	3.1247e+06	3.1708e+06	3.4787e+06
225	06	3.1708e+06	3.4787e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06
226	06	3.1708e+06	3.4787e+06	3.4787e+06	3.1708e+06	3.1708e+06	3.1708e+06
227	06	3.1708e+06	3.1708e+06	3.4787e+06	3.1708e+06	3.1247e+06	3.1708e+06
228	06	3.1247e+06	3.1247e+06	3.1708e+06	3.1247e+06	3.1247e+06	3.1708e+06
229	06	3.1708e+06	3.1708e+06	3.1247e+06	3.1708e+06	3.1247e+06	3.1708e+06
230	06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1247e+06	3.1708e+06
231	06	3.1247e+06	3.1247e+06	3.1247e+06	3.1247e+06	3.1708e+06	3.1708e+06
232	06	3.1247e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1247e+06
233	06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06	3.1708e+06
234	06	3.1247e+06	3.1708e+06	3.1708e+06	3.1247e+06	3.1247e+06	3.1708e+06
235	06	3.4787e+06	3.4787e+06	3.1708e+06	3.1247e+06	3.1247e+06	3.1708e+06

Processed Image Επεξεργασμένη εικόνα

Segmented image by Felzenszwalb & Huttenlocher efficient graph based method

Ιστόγραμμα επεξεργασμένης εικόνας

Histogram

Αφού ολοκληρωθεί η επεξεργασία της εικόνας που επιλέχθηκε επάνω δεξιά εμφανίζεται το αποτέλεσμα της επεξεργασίας, στη μέση εμφανίζεται ένας πίνακας με τιμές που η κάθε μία τιμη σε κάθε κελί του πίνακα αντιστοιχεί στην τιμή εξεχουσας περιοχής του κάθε pixel.

Στην περίπτωση της δεύτερης μεθόδου (RC) εμφανίζεται και μια επιπλέον εικόνα στο κάτω αριστερό μέρος του παράθυρου που είναι το αποτέλεσμα της κατάτμησης της εικόνας σε περιοχές πριν την επεξεργασία της.

Τέλος στο κάτω δεξιό μέρος εμφανίζεται πλέον το ιστόγραμμα της επεξεργασμένης εικόνας που βρίσκεται ακριβώς από επάνω. Μετά αν επιθυμούμε μπορούμε να αποθηκεύσουμε το αποτέλεσμα της επεργασίας πατώντας το κίτρινο κουμπί όπως επίσης υπάρχει δυνατότητα εκτύπωσης του αποτελέσματος πατώντας το γαλάζιο κουμπί που βρίσκεται ακριβώς πάνω από εκεί που εμφανίζεται η επεξεργασμένη εικόνα.

Πειραματικές συγκρίσεις

Οι αλγόριθμοι HC & RC υλοποιήθηκαν σε Matlab. Για τις άλλες μεθόδους IT, GB, SR, FT, CA χρησιμοποιήθηκαν οι υπάρχουσες υλοποιήσεις. Για απλές εικόνες η HC μέθοδος χρειάζεται $O(N)$ χρόνο υπολογισμού του αποτελέσματος και είναι αρκετά αποτελεσματική για εφαρμογές σε πραγματικό χρόνο. Αντίθετα η RC μέθοδος είναι πιο αργή λόγω της κατάτμησης της εικόνας αλλά παράγει χάρτες εξεχουσών περιοχών πολύ υψηλότερης ποιότητας.

Κατάτμηση σταθερού κατωφλίου. Ο πιο απλός τρόπος για να κατατμηθούν οι εξέχουσες περιοχές είναι να οριστεί στον χάρτη εξεχουσών περιοχών ένα κατώφλι T_f από 0 έως 255. Για να υπάρξουν πιο σίγουρα κι αξιόπιστα αποτελέσματα συγκρίθηκε το πόσο ικανοποιητικά οι διάφορες μέθοδοι ανιχνεύουν τις εξέχουσες περιοχές χρησιμοποιώντας διάφορες τιμές κατωφλίου από 0 έως 255.

Saliency cut. Θεωρούμε ότι οι χάρτες εξεχουσών περιοχών θα βοηθήσουν στη κατάτμηση των εξεχόντων αντικείμενων. Οι χάρτες χρησιμοποιήθηκαν προηγουμένως για αυτόματα και όχι ορισμένη από το χρήστη κατάτμηση αντικειμένων.

Στην παρούσα εργασία χρησιμοποιείται επαναληπτικά η μέθοδος Grab-cut για να τελειοποιηθεί το αποτέλεσμα της κατάτμησης που προέκυψε από την εφαρμογή του κατωφλίου.

Στην περίπτωση αυτή δεν χρησιμοποιείται περιοχή ορισμένη από το χρήστη διαδραστικά όπως στην συνηθισμένη εφαρμογή της μεθόδου Grabcut αλλά ορίζεται αυτόματα χρησιμοποιώντας τη κατάτμηση που λαμβάνεται από τη μετατροπή του χάρτη εξεχουσών περιοχών σε δυαδική μορφή χρησιμοποιώντας ένα προσαρμοσμένο κατώφλι επιλεγμένο εμπειρικά προκειμένου να δώσει ένα αξιόπιστο αποτέλεσμα.

Αφού αρχικοποιείται η διαδικασία μια φορά εφαρμόζεται η μέθοδος του Grabcut επαναληπτικά για τη βελτίωση του αποτελέσματος. Χρειάστηκαν το πολύ 2 επαναλήψεις με βάση τα πειράματα που έγιναν. Μετά από κάθε επανάληψη εφαρμόζονται οι ενσωματωμένες στο πρόγραμμα του Matlab μέθοδοι dilation & erosion για να παραχθεί ο χάρτης που θα χρησιμοποιηθεί στην επόμενη επανάληψη. Όπως φαίνεται η περιοχή που βρίσκεται εκτός της περιοχής όπου εφαρμόστηκαν οι παραπάνω 2 μέθοδοι βρίσκεται πλέον στο φόντο (background) της εικόνας ενώ η άλλη περιοχή όπου εφαρμόστηκαν οι 2 μέθοδοι βρίσκεται στο προσκήνιο (foreground) και οι υπόλοιπες περιοχές του χάρτη δεν ορίζεται αν βρίσκονται στο προσκήνιο ή το φόντο. Η μέθοδος Grabcut η οποία είναι από μόνη της μια επαναληπτική μέθοδος που χρησιμοποιεί συνδυασμό Γκαουσιανών μοντέλων βοηθάει στον ορισμό των περιοχών των εξεχόντων αντικειμένων σε κάθε επανάληψη. Οι περιοχές που βρίσκονται πιο κοντά στο εξέχον αντικείμενο τείνουν περισσότερο να γίνουν μέρος του αντικειμένου σε σχέση με τις πιο μακρινές.

Αλλαγή μεγέθους εικόνας. Οι χάρτες εξεχουσών περιοχών χρησιμοποιούνται για τον προσδιορισμό της σχετικής σημασίας των τμημάτων όλης της εικόνας. Οι RC χάρτες βοηθούν στην παραγωγή καλύτερων αποτελεσμάτων μετά από αλλαγή μεγέθους της εικόνας από τη στιγμή που οι περιοχές του εξέχοντος αντικείμενου δεν έχουν τυχόν ανωμαλίες κάτι που είναι σημαντικό για τις μεθόδους αλλαγής μεγέθους της εικόνας που είναι βασισμένες στην ενέργεια. Οι χάρτες εξεχουσών περιοχών έχουν υψηλότερες τιμές στα άκρα του αντικείμενου κάτι που είναι λιγότερο βολικό για εφαρμογές όπως αλλαγή μεγέθους εικόνας που απαιτούν ολόκληρο το αντικείμενο να είναι το ίδιο τονισμένο.

Επίλογος και μελλοντικές εργασίες

Παρουσιάστηκαν και υλοποιήθηκαν δυο μέθοδοι ανίχνευσης εξεχουσών περιοχών βασισμένες στην ολική αντίθεση, αυτές είναι οι: Histogram based contrast(HC), δηλαδή μέθοδος εύρεσης εξεχουσών περιοχών βασισμένη σε ιστόγραμμα & Region based contrast(RC), δηλαδή μέθοδος εύρεσης εξεχουσών περιοχών βασισμένη σε περιοχές. Όσο η HC μέθοδος είναι γρήγορη και παράγει αποτελέσματα υψηλής ποιότητας με προσεγμένες λεπτομέρειες, τόσο η RC μέθοδος παράγει υψηλής ποιότητας χάρτες με μειωμένο κόστος υπολογιστικών πόρων. Τα πειράματα έδειξαν ότι τα αποτελέσματα ήταν πολύ ικανοποιητικά ενώ οι συγκεκριμένες μέθοδοι είναι απλές στην υλοποίηση αλλά και αποτελεσματικές.

Στο μέλλον προβλέπεται να αναπτυχθούν γρήγοροι κι αποτελεσματικοί αλγόριθμοι που θα χρησιμοποιούν χωρικές συσχετίσεις για εύρεση εξεχουσών περιοχών και θα παράγουν αποτελέσματα πολύ υψηλής ποιότητας. Επίσης θα αναπτυχθούν γρήγοροι και αποτελεσματικοί αλγόριθμοι που θα μπορούν να διαχειριστούν φωτογραφίες με πλούσια φόντα.

Βιβλιογραφία

- [1] R. Achanta, F. Estrada, P. Wils, and S. Susstrunk. Salient region detection and segmentation. In ICVS, pages 66–75. Springer, 2008. 2, 4, 6
- [2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In CVPR, pages 1597–1604, 2009. 1, 2, 4, 5, 6, 7
- [3] R. Achanta and S. Susstrunk. Saliency Detection for Content-aware Image Resizing. In ICIP, 2009. 7
- [4] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. ACM TOG, 28(5):124:1–10, 2009. 1
- [5] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system: an overview. IEEE Trans. on Consumer Electronics, 46(4):1103–1127, 2002. 1
- [6] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. ACM TOG, 21(3):769–776, 2002. 7
- [7] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. Annual review of neuroscience, 18(1):193–222, 1995. 1
- [8] W. Eihhauser and P. Konig. Does luminance-contrast contribute to a saliency map for overt visual attention? European Journal of Neuroscience, 17:1089–1097, 2003. 4
- [9] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. IJCV, 59(2):167–181, 2004. 4, 5
- [10] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In CVPR, pages 2376–2383, 2010. 2, 4, 5, 6, 7
- [11] J. Han, K. Ngan, M. Li, and H. Zhang. Unsupervised extraction of visual attention objects in color images. IEEE TCSV, 16(1):141–145, 2006. 1, 6

- [12] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in neural information processing systems*, 19:545, 2007. 2, 4, 6
- [13] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, pages 1–8, 2007. 2, 4, 5, 6
- [14] H. Huang, L. Zhang, and T.-N. Fu. Video painting via motion layer manipulation. *Comput. Graph. Forum*, 29(7):2055–2064, 2010. 7
- [15] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998. 1, 2, 4, 6
- [16] B. Ko and J. Nam. Object-of-interest image segmentation based on human attention and semantic region clustering. *J Opt Soc Am*, 23(10):2462, 2006. 1, 6
- [17] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985. 1, 2
- [18] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. In *CVPR*, pages 1–8, 2007. 2
- [19] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, pages 374–381, 2003. 2, 4, 6
- [20] S. K. Mannan, C. Kennard, and M. Husain. The role of visual salience in directing eye movements in visual object agnosia. *Current biology*, 19(6):247–248, 2009. 1
- [21] J. Reynolds and R. Desimone. Interacting roles of attention and visual salience in v4. *Neuron*, 37(5):853–863, 2003. 1
- [22] C. Rother, V. Kolmogorov, and A. Blake. “Grabcut”—Interactive foreground extraction using iterated graph cuts. *ACM TOG*, 23(3):309–314, 2004. 5, 6

[23] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In CVPR, pages II: 37–44, 2004. 1

[24] H. Teuber. Physiological psychology. *Annual Review of Psychology*, 6(1):267–296, 1955. 1

[25] A. M. Triesman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. 1

[26] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, pages 5:1–7, 2004. 1

[27] O. S. Yu-Shuen Wang, Chiew-Lan Tai and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5), 2008. 1

[28] S. Zeki. *Inner vision: An exploration of art and the brain*. Oxford University Press, 1999. 7

[29] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *ACM Multimedia*, pages 815–824, 2006. 2, 3, 4, 5, 6, 7

[30] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Comput. Graph. Forum*, 28(7):1897–1906, 2009. 1, 7

[31] Y.-F. Zhang, S.-M. Hu, and R. R. Martin. Shrinkability maps for content-aware video resizing. *Comput. Graph. Forum*, 27(7):1797–1804, 2008. 1

Παράρτημα

Κώδικας Matlab για Παραγωγή τιμών εξεχουσών περιοχών μέσω αντίθεσης βασισμένης σε ιστόγραμμα (HC)

```
function [imhc,Sal] = H (path,filename)

main = cd (path);
[img, map] = imread([path,filename]);
cd (main);

s=size(img,1)*size(img,2);

imgq = KMeansColorReduction (img,map);

lab_img = rgb2lab(imgq);

L = lab_img(:,:,1);
A = lab_img(:,:,2);
B = lab_img(:,:,3);
%%

fpix = Clr_Freq(img);
v=1;
pix=zeros(s,4);

for i = 1:size(lab_img,1)
    for j = 1:size(lab_img,2)
        pix(v,1)=L(i,j);
        pix(v,2)=A(i,j);
        pix(v,3)=B(i,j);
        pix(v,4)=fpix(i,j);
        v=v+1;
    end
end

pix( ~any(pix(:,1),2), : ) = [];
pix = unique(pix(:,:),'rows','stable');

n = size(pix,1);
Sc = zeros(n);
```

```

imhc = zeros(size(lab_img,1),size(lab_img,2));

Dist = zeros(n,n);

for i = 1:n
    for c=1:n
        Dist(i,c) = cmcde(pix(i,1:3),pix(c,1:3));
        Sc(i) = Sc(i) + Dist(i,c)*pix(c,4);
    end

    [row,col] = find((lab_img(:,:,1) ==
pix(i,1)) & (lab_img(:,:,2) == pix(i,2)) & (lab_img(:,:,3) ==
pix(i,3)));

    for j = 1:size(row,1)
        imhc(row(j),col(j)) = Sc(i);
    end

end

MAX = max(imhc(:));
imhc = imhc./MAX;
%%
T = zeros(n);
m = n/4;
m = int8(m);
m = double(m);

[D,I] = sort(Dist);

SS = zeros(n);
S = zeros(n);
Sal = zeros(n,n);

for i = 1:n

    for c=1:m
        T(i) = T(i) + D(c,i);
    end

    for c=1:m
        idx = I(c,i);
        SS(i) = SS(i) + ( (T(i) - D(c,i)) * Sc(idx));
    end

    S(i) = (1 / ((m-1) * T(i)) ) * SS(i);

    [row,col] = find((lab_img(:,:,1) ==
pix(i,1)) & (lab_img(:,:,2) == pix(i,2)) & (lab_img(:,:,3) ==
pix(i,3)));

```

```

        for l = 1:size(row,1)
            imhc(row(l),col(l)) = S(i);
            Sal(row(l),col(l)) = S(i);
        end
    end
end

MAX = max(imhc(:));
imhc = imhc./MAX;
%%

for i=1:4
    avg=mean2(imhc);

    fseed = imhc(:, :) > avg;
    bseed = imhc(:, :) <= avg;

    fmask = zeros(size(imhc));
    bmask = fmask;
    fmask(:, :) = fseed;
    bmask(:, :) = bseed;

    roi = false(size(imhc));
    roi(5:end-5,5:end-5,:) = true;
    %roi(size(fmask,1),size(fmask,2)) = true;

    L = superpixels(imhc,500);

    imhc = grabcut(imhc,L,roi,fmask,bmask);
    imhc = im2uint8(imhc);

    t = adaptthresh(imhc,1);
    imhc = imbinarize(imhc,t);
    imhc = im2uint8(imhc);

    SE = strel('square', 5);
    imhc = imdilate(imhc,SE);
    imhc = imerode(imhc,SE);
end

imhc = im2uint8(imhc);
end

```

Color frequency

```
function [normImage] = Clr_Freq(originalImage)

rows = size(originalImage,1);
columns = size(originalImage,2);

for row = 1 : rows
    for col = 1: columns
        redValue = floor(double(originalImage(row, col,
1)) / reductionFactor) + 1;% Convert from 0-255 to 1-256
        greenValue = floor(double(originalImage(row, col,
2)) / reductionFactor) + 1;% Convert from 0-255 to 1-256
        blueValue = floor(double(originalImage(row, col, 3))
/ reductionFactor) + 1;    % Convert from 0-255 to 1-256
        gamut3D(redValue, greenValue, blueValue) =
gamut3D(redValue, greenValue, blueValue) + 1;
    end
end

% Now construct the color frequency image.
% Make an image where we get the color of the original
image, and have the output value of the color
% frequency image be the number of times that exact color
occurred in the original image.
colorFrequencyImage = zeros(rows, columns);
for row = 1 : rows
    for col = 1: columns
        redValue = floor(double(originalImage(row, col,
1)) / reductionFactor) + 1;% Convert from 0-255 to 1-256
        greenValue = floor(double(originalImage(row, col,
2)) / reductionFactor) + 1;% Convert from 0-255 to 1-256
        blueValue = floor(double(originalImage(row, col, 3))
/ reductionFactor) + 1;    % Convert from 0-255 to 1-256
        freq = gamut3D(redValue, greenValue, blueValue);
        colorFrequencyImage(row, col) = freq;
    end
end
end
numberOfColors = double(rows) * double(columns);
numberOfColors = sum(sum(sum(gamut3D)));
%sumr = sum(sum(colorFrequencyImage));
normImage = colorFrequencyImage * 255 /
max(max(colorFrequencyImage));
end
```

Κώδικας Matlab για Παραγωγή τιμών εξεχουσών περιοχών μέσω αντίθεσης βασισμένης σε περιοχές (RC)

```
function [imrc,imgs,imrcS] = RSal(path,filename)
%% Preprocessing Image

f = waitbar(0,'Preprocessing Your Image Please Wait...');
pause(.05)

main = cd (path);
[img,map] = imread([path,filename]);
cd (main);

%
p = 0.9;
imgdouble = im2double(img);
avg = mean2(imgdouble);
s = std2(imgdouble);

img = imadjust(img,[avg-p*s 1],[avg+p*s 1]);
[img,~,~] = imreducehaze(img,p,'method','approxdcpr');

img = localcontrast(img,0.2,-0.4);
img = localcontrast(img,0.3,-0.8);

img = imsharpen(img,'Radius',0.3,'Amount',0.1);
%'Threshold',0);
img = uint8(img);
f1 = figure;
imshow(img);
pause(2);
close(f1);
%}
%% //Segmenting Image

waitbar(.1,f,'Segmenting Image');
pause(.05)

cd('Felzenszwalb-Segmentation-master');
[imgs,NR] = segmentFelzenszwalb(img, 1, 500, 200, false, 0 ,
false);
imgs=imgs+1;
cd ..;

%% //Reducing Colors
waitbar(.2,f,'Reducing Colors');
```



```

pause(.05)

img = KMeansColorReduction (img,map,NR);
img=uint8(img);
lab_img = rgb2lab(img);
%% //Initializing Regions

waitbar(.3,f,'Initializing Regions');
pause(.05)

reg = regionprops3(imgs,'Volume','Centroid','VoxelList');

weight = table2array(reg(:,1));
centr = table2array(reg(:,2));
pix = table2array(reg(:,3));

weight( ~any(weight(:,1),2), : ) = [];
centr( ~any(centr(:,1),2), : ) = [];

idx=cellfun(@isempty,pix);
pix(idx) = [];
%% //Initializing Computing Tables

waitbar(.4,f,'Initializing Computing Tables');
pause(.05)

Table = Init(lab_img,NR,imgs);
%% //Spatial Distances Computing

waitbar(.5,f,'Spatial Distances Computing');
pause(.05)

Ds = Dist (centr,NR);
%% //Color Distances Computing

waitbar(.6,f,sprintf('%d Color Distances Computing',NR));
pause(.05)

idx=cellfun(@isempty,Table);
Table(idx) = [];

Dr = CDist (NR,Table);
%% //Saliency Values Computing

waitbar(.7,f,'Saliency Values Computing');
pause(.05)
SS = Saliency(Ds,NR,weight,Dr);
%% //Values Assignment

waitbar(.8,f,'Values Assignment');
pause(.05)

```

```

imrc = Assignment(img,SS,pix);
imrcS = imrc;
%% //Thresholding Saliency Map

waitbar(.9,f,'Thresholding Saliency Map');
pause(.05)

for i=1:4

    MAX = mean(imrc(:));

    fseed = imrc(:, :) > MAX;%*0.05;
    bseed = imrc(:, :) <= MAX;%*0.05;

    fmask = zeros(size(imrc));
    bmask = fmask;
    fmask(:, :) = fseed;
    bmask(:, :) = bseed;

    roi = false(size(imrc));
    roi(5:end-5,5:end-5,:) = true;
    %roi(size(fmask,1),size(fmask,2)) = true;

    L = superpixels(imrc,500);

    imrc = grabcut(imrc,L,roi,fmask,bmask);
    imrc = im2uint8(imrc);

    t = adaptthresh(imrc,1);
    imrc = imbinarize(imrc,t);
    imrc = im2uint8(imrc);

    SE = strel('square', 5);
    imrc = imdilate(imrc,SE);
    imrc = imerode(imrc,SE);
end
%% Finishing Process

waitbar(.95,f,'Finishing Process');
pause(.05)
imrc = im2uint8(imrc);
waitbar(1,f,'Finished');
pause(.05)
close(f)
end

```

Init.m

```
function Table = Init(lab_img,NR,imgs)

lab_imgres = reshape(lab_img,[],3);
imgsrres = reshape(imgs,[],1);
%imgres = reshape(img,[],3);

%[lab_imgres,ia,~] =
unique(lab_imgres(:,:,),'rows','stable');

lab_imgress = zeros(size(lab_imgres));
%imgress = zeros(size(imgres));

[imgss,idx] = sort(imgsrres);

for i=1:size(lab_imgres,1)

    lab_imgress(i,:) = lab_imgres((idx(i)),:);
    %imgress(i,:) = imgres((idx(i)),:);
end

v = 1;
vv = 1;
Table = cell(NR,1);

for i=1:NR

    tt = zeros((size(imgss,1)),3);

    while(vv==imgss(v))
        if(v<size(lab_imgres,1))
            tt(v,:) =
[lab_imgress(v,1),lab_imgress(v,2),lab_imgress(v,3)];
%,imgress(v,1),imgress(v,2),imgress(v,3),imgss(v,1)] ;
            v=v+1;
        else
            break
        end
    end
    tt( ~any(tt(:,1),2), : ) = [];
    Table(i) = {tt(:,:)};
    vv = vv+1;
end
end
```

Dist.m

```
function Ds = Dist(centr,n)

Ds = zeros(n,n);
for i=1:n
    for j=1:n
        if (j~=i) && (i<j)
            Ds(i,j) = pdist2(centr(i),centr(j));
        else
            Ds(i,j) = Ds(j,i);
        end
    end
end

MAX = max(Ds(:));
Ds = Ds./MAX;
end
```

Cdist.m

```
function Dr = CDist(nr,Table)
%% Initializing

w = waitbar(0,'Color Distances Computing ...');
pause(.05)

Dr = zeros(nr,nr);
%% Initializing 1st Table

for i=1:nr

    waitbar((1/nr)*i,w,sprintf('\nComputing Distance Between
%d/%d \n',i,nr));
    pause(.05)

    region1 = cell2mat(Table(i));
    p = zeros(size(region1,1),3);

    for r=1:size(region1,1)

        p(r,1) = region1(r,1);%L
        p(r,2) = region1(r,2);%A
        p(r,3) = region1(r,3);%B
    end

    [~,~,ic] = unique(p, 'rows', 'stable');
    h = accumarray(ic, 1);
    Res = h(ic);

    f1 = zeros(size(p,1),1);
    for ii=1:size(p,1)

        f1(ii) = Res(ii);
    end

    [p,~,~] = unique(p(:,:),'rows','stable');
    %% Initializing 2nd Table

    for j=1:nr

        if (j~=i) && (i<j)

            region2 = cell2mat(Table(j));
            pp = zeros(size(region2,1),3);

            for rr=1:size(region2,1)
```

```

        pp(rr,1) = region2(rr,1);
        pp(rr,2) = region2(rr,2);
        pp(rr,3) = region2(rr,3);
    end

    [~,~,ic] = unique(pp, 'rows', 'stable');
    h = accumarray(ic, 1);
    Res = h(ic);

    f2 = zeros(size(pp,1),1);
    for ii=1:size(pp,1)

        f2(ii) = Res(ii);
    end

    [pp,~,~] = unique(pp(:,:),'rows','stable');
    %% Calc Color Dist

    DD = zeros(size(p,1),1);

    for r=1:size(p,1)
        for rr=1:size(pp,1)

            if (r~=rr) && (r<rr)

                DD(r) = DD(r) +
(cmcde(p(r,1:3),pp(rr,1:3))*f1(r)*f2(rr));
            else

                DD(r) = DD(r) + DD(rr);
            end
        end
    end

    Dr(i,j) = sum(DD(:));
else

    Dr(i,j) = Dr(j,i);
end
end
end
end
%% Normalize

MAX = max(Dr(:));
Dr = Dr./MAX;
close(w);
end

```

Saliency.m

```
function SS = Saliency(Ds,n,weight,Dr)

S = zeros(n,n);
for i=1:n
    for j=1:n
        if (j~=i) && (i<j)

            S(i,j) = exp(-Ds(i,j)/0.4)*weight(j)*Dr(i,j);
        else

            S(i,j) = S(j,i);
        end
    end
end

SS = sum(S,2);
end
```

Assignment.m

```
function imrc = Assignment(img,SS,pix)

imrc = zeros(size(img,1),size(img,2));
for i=1:size(SS,1)

    region = cell2mat(pix(i));

    for j=1:size(region,1)

        x1 = region(j,2);
        y1 = region(j,1);
        imrc(x1,y1) = SS(i);
    end
end

MAX = max(imrc(:));
imrc = imrc./MAX;
end
```

CMCDE.m

```
function [de,dl,dc,dh] = cmcde(lab1,lab2,s1,sc)

if (nargin<4)
    %disp('using default values of 1:c')
    s1=1; sc=1;
end

de = zeros(1,size(lab1,2));
dl = zeros(1,size(lab1,2));
dc = zeros(1,size(lab1,2));
dh = zeros(1,size(lab1,2));

% first compute the CIELAB deltas
dl = lab2(:,1)-lab1(:,1);
dc = (lab2(:,2).^2 + lab2(:,3).^2).^0.5-(lab1(:,2).^2 +
lab1(:,3).^2).^0.5;
dh = ((lab2(:,2)-lab1(:,2)).^2 + (lab2(:,3)-lab1(:,3)).^2 -
dc.^2);
dh = (abs(dh)).^0.5;
% get the polarity of the dh term
dh = dh.*dhpolarity(lab1,lab2);

% now compute the CMC weights
Lweight = [lab1(:,1)<16]*0.511+(1-
[lab1(:,1)<16]).*(0.040975*lab1(:,1))./(1 +
0.01765*lab1(:,1));
[h,c] = cart2pol(lab1(:,2), lab1(:,3));
h = h*180/pi;

Cweight = 0.638 + (0.0638*c)./(1 + 0.0131*c);
index = (164<h & h<345);
T = index.*(0.56 + abs(0.2*cos((h+168)*pi/180))) + ...
(1-index).*(0.36 + abs(0.4*cos((h+35)*pi/180)));
F = ((c.^4)./(c.^4 + 1900)).^0.5;
Hweight = Cweight.*(T.*F + 1 - F);

dl = dl./(Lweight*s1);
dc = dc./(Cweight*sc);
dh = dh./Hweight;

de = (dl.^2 + dc.^2 + dh.^2).^0.5;
end

function [p] = dhpolarity(lab1,lab2)
% function [p] = dhpolarity(lab1,lab2)
% computes polarity of hue difference
% p = +1 if the hue of lab2 is anticlockwise
```



```

% from lab1 and p = -1 otherwise
[h1,c1] = cart2pol(lab1(:,2), lab1(:,3));
[h2,c2] = cart2pol(lab2(:,2), lab2(:,3));

h1 = h1*180/pi;
h2 = h2*180/pi;

index = (h1<0);
h1 = (1-index).*h1 + index.*(h1+360);
index = (h2<0);
h2 = (1-index).*h2 + index.*(h2+360);

index = (h1>180);
h1 = (1-index).*h1 + index.*(h1-180);
h2 = (1-index).*h2 + index.*(h2-180);

p = (h2-h1);

index = (p==0);
p = (1-index).*p + index*1;
index = (p>180);
p = (1-index).*p + index.*(p-360);

p = p./abs(p);

end

```

Kmeans Color Reduction.m

```
function outImg = KMeansColorReduction (ImgMat,inMap,NR)

%Read Input Image

s_img = size(ImgMat);
s_map = size(inMap);

if(s_map(1) == 0)
    %sRGB Color Image
    inImg = ImgMat;
else
    %Indexed Color Image
    inImg = ind2rgb(ImgMat, inMap);
    inImg = round(inImg .* 255);
end

%K-Means

r = inImg(:,:,1);
g = inImg(:,:,2);
b = inImg(:,:,3);
inputImg = zeros((s_img(1) * s_img(2)), 3);
inputImg(:,1) = r(:);
inputImg(:,2) = g(:);
inputImg(:,3) = b(:);
inputImg = double(inputImg);

[idx, C] = kmeans(inputImg, NR, 'EmptyAction', 'error',
'MaxIter', 1000);

palette = round(C);

%Color Mapping

idx = uint8(idx);
outImg = zeros(s_img(1),s_img(2),3);
temp = reshape(idx, [s_img(1) s_img(2)]);

for i = 1 : 1 : s_img(1)
    for j = 1 : 1 : s_img(2)
        outImg(i,j,:) = palette(temp(i,j),:);
    end
end
end
end
```