



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Ευφυής Φορητή Πλατφόρμα Ηλεκτρονικού Εμπορίου
με Δυνατότητα Επεξεργασίας Εικόνας και
Ενσωμάτωσης Γεωγραφικών Δεδομένων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΛΙΑΝΟΥ ΑΝΑΣΤΑΣΙΟΥ

Επιβλέπων : Βασίλειος Λούμος
Καθηγητής Ε.Μ.Π.

Αθήνα, 26 Ιανουαρίου 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Ευφυής Φορητή Πλατφόρμα Ηλεκτρονικού Εμπορίου με Δυνατότητα Επεξεργασίας Εικόνας και Ενσωμάτωσης Γεωγραφικών Δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΛΙΑΝΟΥ ΑΝΑΣΤΑΣΙΟΥ

Επιβλέπων : Βασίλειος Λούμος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26^η Ιανουαρίου 2011.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Βασίλειος Λούμος

.....
Ελευθέριος Καγιάφας

.....
Χρήστος-Νικόλαος
Αναγνωστόπουλος

Καθηγητής Ε.Μ.Π.

Καθηγητής Ε.Μ.Π.

Επίκουρος Καθηγητής Π.Α.

Αθήνα, Ιανουάριος 2011

(Υπογραφή)

.....

ΛΙΑΝΟΣ ΑΝΑΣΤΑΣΙΟΣ

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

Copyright © Αναστάσιος Λιανός, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved

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

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

Περίληψη

Σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη μίας ολοκληρωμένης ευφυούς φορητής πλατφόρμας ηλεκτρονικού εμπορίου (m-commerce). Το σύστημα αυτό επιτρέπει την εισαγωγή αγγελιών για ενοικίαση και πώληση οικιών από οποιαδήποτε φορητή πλατφόρμα, που υποστηρίζει Java ME. Οι αγγελίες αυτές, αποθηκεύονται στο υποσύστημα της βάσης δεδομένων. Ο χρήστης έχει τη δυνατότητα να αποθηκεύσει φωτογραφίες της εν λόγω αγγελίας στον εξυπηρετητή (server) και να αναζητήσει αγγελίες που τον ενδιαφέρουν.

Κύριο άξονα των χαρακτηριστικών της πλατφόρμας αποτελούν οι εικόνες που συνοδεύουν κάθε αγγελία. Εκτός της δυνατότητας προσάρτησης εικόνων σε κάθε αγγελία από τους χρήστες, δίνεται η δυνατότητα αναζήτησης αγγελιών, όχι μόνο μέσω των δεδομένων της βάσης, αλλά και μέσω σύγκρισης μιας εικόνας-δείγματος με τις εικόνες των αποθηκευμένων αγγελιών. Επίσης, μέσω επεξεργασίας των φωτογραφιών εξάγονται συμπεράσματα για κάποια χαρακτηριστικά της οικίας που αυτές απεικονίζουν και συμπληρώνονται αυτόματα οι αντίστοιχες φόρμες, απλοποιώντας τη διαδικασία εισαγωγής μιας νέας αγγελίας. Υλοποιήθηκε επίσης και σύστημα συστάσεων, το οποίο προτείνει στους χρήστες και άλλες αγγελίες που πιθανά να τους ενδιαφέρουν. Τις προτάσεις αυτές, τις στηρίζει στα δεδομένα των αναζητήσεων που έχουν πραγματοποιήσει, αλλά και σε συγκρίσεις ομοιότητας των εικόνων των αποτελεσμάτων με τις αποθηκευμένες στη βάση.

Για τη δοκιμή του συστήματος που υλοποιήθηκε εισήχθησαν στη βάση δεδομένων αγγελίες και αντίστοιχες εικόνες οικιών από βάσεις φωτογραφιών στο διαδίκτυο. Η εισαγωγή πραγματοποιήθηκε με χρήση προσομοιωτών κινητών τηλεφώνων που υποστηρίζονται από μία πληθώρα κατασκευαστών. Τα αποτελέσματα των δοκιμών κατέδειξαν τη σταθερότητα, ταχύτητα και αποτελεσματικότητα του συστήματος.

Στα πλαίσια της ανάπτυξης του συστήματος αξιοποιήθηκαν τεχνολογία και αλγόριθμοι από τις περιοχές της επεξεργασίας εικόνας, πρωτοκόλλων επικοινωνίας, βάσεων δεδομένων. Συγκεκριμένα, χρησιμοποιήθηκε γλώσσα Java για την υλοποίηση της φορητής πλατφόρμας (Java Micro Edition – Java ME) και της πλατφόρμας εξυπηρετητή (Java Server Pages – JSP), Java Servlets). Επίσης, για το σχεδιασμό της

ιστοσελίδας του εξυπηρετητή έγινε χρήση της scripting γλώσσας Javascript. Τέλος, η βάση δεδομένων υλοποιήθηκε με τη Δομημένη Γλώσσα Ερωτημάτων SQL σε εξυπηρετητή MySQL.

Για την επεξεργασία των φωτογραφιών χρησιμοποιήθηκαν από το σύστημα ένας τυπικός αλγόριθμος Ανάπτυξης Περιοχών (region growing) για Κατακερμάτιση Εικόνας σε Περιοχές (image segmentation), ο αλγόριθμος Ανίχνευσης Ακμών του Canny (Canny Edge Detection) και η μέθοδος σύγκρισης εικόνων με βάση τα συζευγμένα ιστογράμματα εικόνων.

Λέξεις Κλειδιά: <<ηλεκτρονικό εμπόριο, φορητή πλατφόρμα, κατακερμάτιση εικόνας, συστήματα σύστασης, σύγκριση εικόνων >>

Abstract

This dissertation aims to develop a smart client-server m-commerce platform. It allows the user to upload real estate ads, which include images, which are then inserted into a database subsystem. The insertions took place using a wide range of mobile devices emulators from most enterprise developers.

The majority of the implemented system's features rely on the stored images of the ads. So, not only can the users upload their own real estate ads including images, but they are also able to make a query using either text or photos as search inquiry. The system also extracts features of the aforementioned houses from their uploaded photos and automatically fills the related forms for the user, thus resulting in ease of access. Finally, the platform's recommendation subsystem relies also on image processing techniques. It recommends ads to the users, using both data from their inquiries and image comparison between the images of the results and those of the database.

The system was tested by inserting multiple test ads and their related photos into the database. The test results proved the system to be stable, fast and effective.

The system was implemented using technology and algorithms from the scientific fields of signal processing, communication protocols and databases. Specifically, the Java language was used for implementing the mobile (Java Micro Edition – ME) and server platform (Java Server Pages – JSP, Java Servlets). In addition, the script language Javascript was used in the design process of the server website. Lastly, the database system was implemented using the Structured Query Language MySQL.

The algorithms of the system used for image processing are a typical Region Growing Image Segmentation Algorithm, the Canny Edge Detection Algorithm and the Joint Histograms Image Comparison Method as suggested by Greg Pass and Ramin Zabih.

Keywords: <<e-commerce, m-commerce, image segmentation, mobile platform, image comparison, recommender system, Windows algorithm, Java>>

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ.....	1
1.1. Ορισμοί.....	4
1.1.1. Ηλεκτρονικό Εμπόριο (e-commerce)	4
1.1.2. Ηλεκτρονικό Εμπόριο με χρήση Φορητών Συσκευών (m-commerce)	4
1.1.3. Ψηφιακή Επεξεργασία Εικόνας – Εικονοστοιχείο	5
1.2. Συστήματα Ηλεκτρονικού Εμπορίου.....	6
2. ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	9
2.1. Μέθοδος Σύγκρισης Ομοιότητας Εικόνων	9
2.2. Συστήματα Σύστασης	11
3. ΜΕΘΟΔΟΣ.....	17
3.1. Εύρεση Επιπλωμένου Χώρου από Εικόνα	17
3.1.1. Αλγόριθμος Κατάτμησης Εικόνων μέσω Ανάπτυξης Περιοχών.....	21
3.1.1.1 Στάδιο Προ-Επεξεργασίας Αλγόριθμου Ανάπτυξης Περιοχών	22
3.1.1.2 Στάδιο Κατάτμησης Εικόνας Αλγόριθμου Ανάπτυξης Περιοχών.....	23
3.1.2. Αλγόριθμος Εύρεσης Ακμών Canny	27
3.1.2.1 Στάδιο Μείωσης Θορύβου	27
3.1.2.2 Στάδιο Υπολογισμού Βαθμίδας Κλίσης	29
3.1.2.3 Στάδιο Καταστολής των Μη-μεγίστων.....	32
3.1.2.4 Στάδιο Κατωφλίωσης με Υστέρηση	33
3.1.3. Τελικός Αλγόριθμος Εύρεσης Επιπλωμένου Χώρου από Εικόνα.....	35
3.2. Σύγκριση Ομοιότητας Εικόνων	37
3.2.1. Στάδιο Προ-επεξεργασίας Εικόνων προς Σύγκριση.....	39
3.2.2. Στάδιο Εξαγωγής Συζευγμένων Ιστογραμμάτων	41
3.2.3. Στάδιο Υπολογισμού μετρικής L1 και Εξαγωγής Συμπερασμάτων.....	44
3.2.4. Εφαρμογή Αλγόριθμου Σύγκρισης Εικόνων	47
3.3. Αλγόριθμος Εύρεσης Κυρίαρχου Χρώματος.....	49
3.3.1. Μετατροπή RGB Τιμών σε HSV.....	49
3.3.2. Υπολογισμός Κυρίαρχου Χρώματος	51
3.3.3. Εφαρμογές Υπολογισμού Κυρίαρχου Χρώματος.....	52
4. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ.....	55
4.1. Υποσύστημα Πελάτη	57
4.1.1. Η Αρχιτεκτονική της Java ME - MIDP	58
4.1.2. Η κλάση MIDlet User	61
4.1.3. Η βοηθητική κλάση CameraCanvas	68
4.1.4. Η βοηθητική κλάση SupportMethods.....	71
4.1.5. Η βοηθητική κλάση Base64.....	73
4.2. Υποσύστημα Εξυπηρετητή	75
4.2.1. Τεχνολογία Web Components	76
4.2.2. Τμήμα Διεπαφής – Ιστοσελίδα Εξυπηρετητή.....	77
4.2.3. Τμήμα Servlet Εξυπηρετητή.....	84
4.2.4. Βοηθητική Κλάση ImageMethods.....	88
4.2.5. Βοηθητική Κλάση CannyEdgeDetector	93
4.2.6. Βοηθητική Κλάση SimpleRegionGrowing.....	94
4.2.7. Βοηθητική Κλάση Base64	95
4.3. Υποσύστημα Βάσης Δεδομένων.....	96
5. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	101
6. ΕΠΙΛΟΓΟΣ	107

7. ΠΑΡΑΡΤΗΜΑΤΑ	109
7.1. Χρωματικά μοντέλα εικόνων.....	109
7.2. Διαμορφώσεις εικόνων	112
7.3. Κατάλογος Κλάσεων και Μεθόδων.....	113
7.4. Βιβλιογραφία	120

1. ΕΙΣΑΓΩΓΗ

Η διπλωματική αυτή έχει ως αντικείμενο την υλοποίηση μίας ολοκληρωμένης ευφούς φορητής πλατφόρμας ηλεκτρονικού εμπορίου. Η πλατφόρμα αυτή έχει τις εξής δυνατότητες:

- Εισαγωγή αγγελίας για ενοικίαση ή πώληση οικίας από φορητή συσκευή που υποστηρίζει Java ME ή ηλεκτρονικό υπολογιστή
- Αναζήτηση αγγελιών με βάση είτε αποθηκευμένα δεδομένα, είτε φωτογραφίες
- Προβολή γεωγραφικών πληροφοριών με βάση τη διεύθυνση που αντιστοιχεί στις οικίες των αγγελιών
- Αυτόματη συμπλήρωση κάποιων στοιχείων (επιπλωμένο, τύπος πατώματος, χρώμα εξωτερικών τοίχων) της φόρμας δεδομένων για την εισαγωγή αγγελίας με βάση στοιχεία που λαμβάνονται από εικόνες
- Σύστημα συστάσεων (recommender systems) με βάση ομοιότητα χώρων των οικιών, όπως αυτή προκύπτει από αντίστοιχες εικόνες των αγγελιών

Το κείμενο αυτό επιχειρεί να παρουσιάσει τα μέρη του συστήματος και τις αρχές, τεχνικές, τεχνολογίες και μεθοδολογίες στις οποίες βασίστηκε. Έτσι, τα αρχικά κεφάλαια ασχολούνται με το γενικό πλαίσιο του συστήματος και στο τελευταίο κεφάλαιο αναλύεται το σύστημα με βάση την τελική μορφή του, όπως αυτή διαμορφώθηκε μετά την ολοκλήρωση του ελέγχου του.

Το σύστημα αποτελείται από τρία υποσυστήματα:

- A) του Πελάτη,
- B) του Εξυπηρετητή,
- Γ) της Βάσης Δεδομένων.

Πιο αναλυτικά, στο κεφάλαιο 1. ΕΙΣΑΓΩΓΗ γίνεται μια εισαγωγική αναφορά στα συστήματα ηλεκτρονικού εμπορίου, τα χαρακτηριστικά και τις χρήσεις τους. Επίσης δίνονται οι ορισμοί και οι εφαρμογές που σχετίζονται με το σύστημα.

Στη συνέχεια, στο κεφάλαιο 2. ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ αναλύονται οι σχετικοί με την υλοποίηση της πλατφόρμας αλγόριθμοι, όπως αυτοί προτάθηκαν στις αντίστοιχες δημοσιεύσεις τους. Γίνεται ανάλυση των πλεονεκτημάτων και μειονεκτημάτων τους και τεκμηριώνονται τα κριτήρια που οδήγησαν στην τελική επιλογή αυτών έναντι εναλλακτικών επιλογών.

Στο κεφάλαιο 3 αναφέρονται οι αλγόριθμοι και οι τεχνικές που χρησιμοποιήθηκαν για την επεξεργασία των φωτογραφιών και την εξαγωγή συμπερασμάτων από αυτές. Αναλύεται ακόμη η διαδικασία κατά την οποία επιτυγχάνεται η αποστολή πολυμεσικών αρχείων από την φορητή συσκευή στον εξυπηρετητή.

Στο κεφάλαιο 4 περιγράφεται η λειτουργία και υλοποίηση του κάθε υποσυστήματος. Το κεφάλαιο αυτό χωρίζεται σε τρία υποκεφαλαία, ένα για κάθε υποσύστημα:

4.1 Σύστημα Πελάτη,

4.2 Σύστημα Εξυπηρετητή,

4.3 Βάση Δεδομένων.

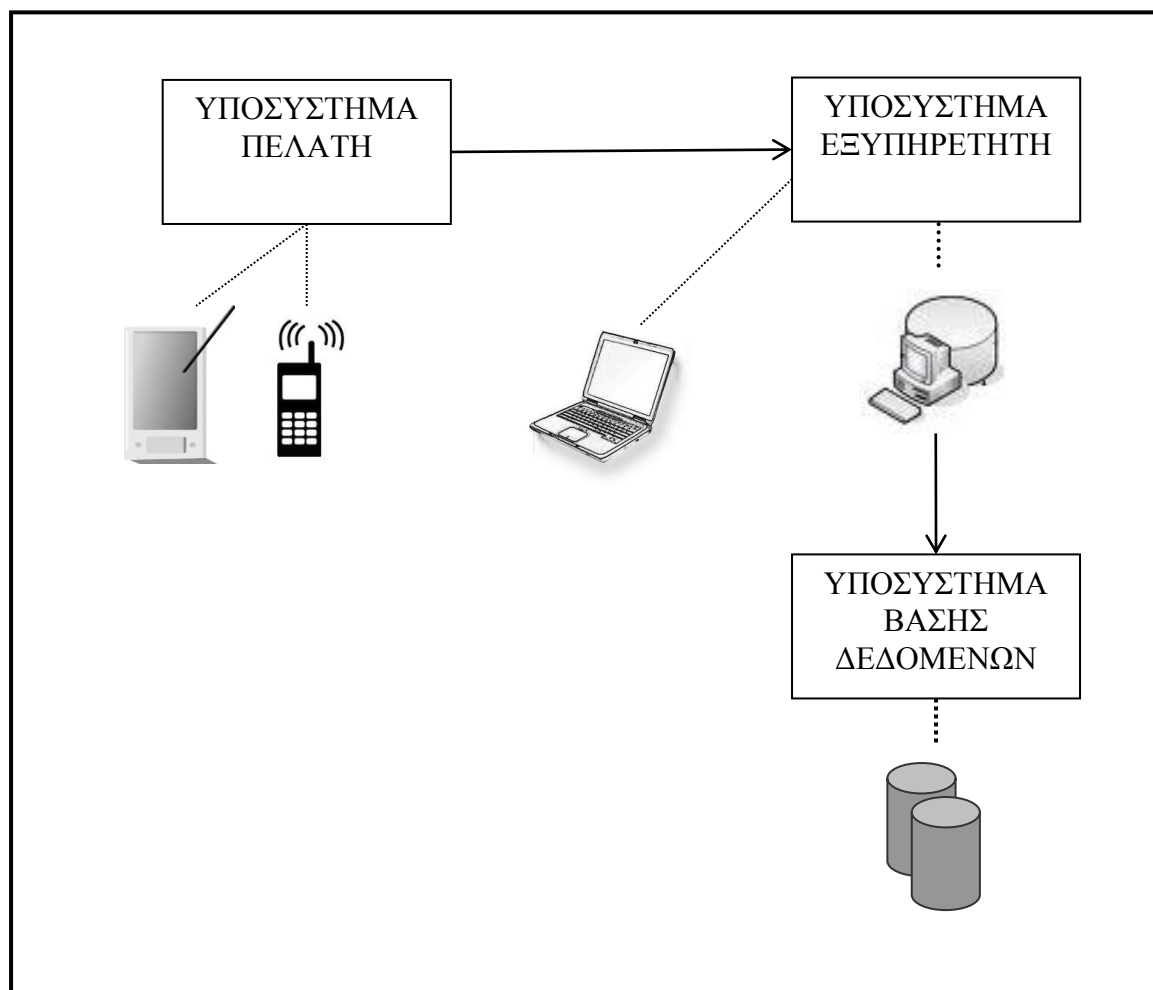
Για καθένα από τα παραπάνω συστήματα γίνεται ανάλυση του σχεδιασμού του, των μεθοδολογιών που διέπουν αυτή, του τρόπου επικοινωνίας του με τα υπόλοιπα υποσυστήματα. Επίσης περιγράφονται οι λεπτομέρειες της υλοποίησής τους και δίνονται στιγμιότυπα από την εκτέλεσή τους.

Στο κεφάλαιο 5, παρουσιάζονται πειραματικά αποτελέσματα του συστήματος. Επίσης παρουσιάζονται μερικά παραδείγματα εκτέλεσης των διαφόρων αλγορίθμων επεξεργασίας εικόνας που χρησιμοποιήθηκαν προς μέτρηση της αποτελεσματικότητας της πλατφόρμας.

Στο έκτο κεφάλαιο, γίνεται μία σύνοψη των συμπερασμάτων που προκύπτουν, ενώ παράλληλα παρουσιάζονται και μελλοντικές επεκτάσεις του συστήματος με σκοπό την περαιτέρω βελτίωσή του.

Το έβδομο και τελευταίο κεφάλαιο, περιέχει τα παραρτήματα με τις βιβλιογραφικές αναφορές, μία γενική περιγραφή των χρωματικών μοντέλων που χρησιμοποιήθηκαν καθώς και μία πλήρη λίστα των κλάσεων και μεθόδων αυτών.

Στην εικόνα 1.1 παρουσιάζεται σχηματικά η διάρθρωση του συστήματος.



Σχήμα 1.1 : Η πλήρης Πλατφόρμα Ηλεκτρονικού Εμπορίου που υλοποιήθηκε κατά τη διάρκεια αυτής της διπλωματικής εργασίας

Ακολουθεί μια γενική περιγραφή των Συστημάτων Ηλεκτρονικού Εμπορίου (e-commerce) και ειδικότερα των Συστημάτων Ηλεκτρονικού Εμπορίου με χρήση Φορητών Συσκευών (m-commerce). Αρχικά δίνονται κάποιοι ορισμοί και γίνεται σύντομη ιστορική αναδρομή στην εξέλιξη των συγκεκριμένων συστημάτων. Στη συνέχεια, αναλύεται η χρησιμότητά τους και παρατίθενται παραδείγματα εφαρμογών

τους. Ακολουθεί περιγραφή του περιβάλλοντος χρήσης των εν λόγω συστημάτων και τέλος αναφέρονται προβλήματα που δημιουργούνται κατά τη λειτουργία τους.

1.1. Ορισμοί

1.1.1. Ηλεκτρονικό Εμπόριο (e-commerce)

Ως ηλεκτρονικό εμπόριο ορίζεται το εμπόριο που πραγματοποιείται με ηλεκτρονικά μέσα, το οποίο βασίζεται δηλαδή στην ηλεκτρονική μετάδοση δεδομένων. Αποτελεί έκφραση των λεγόμενων υπηρεσιών εξ αποστάσεως. Πρόκειται για μια ολοκληρωμένη συναλλαγή που πραγματοποιείται μέσω του διαδικτύου χωρίς να είναι απαραίτητη η φυσική παρουσία των συμβαλλομένων μερών, δηλαδή του πωλητή και του αγοραστή, οι οποίοι μπορούν να βρίσκονται ακόμα και σε διαφορετικές χώρες. Το ηλεκτρονικό εμπόριο διακρίνεται σε

- A) έμμεσο και
- B) άμεσο

Ο πρώτος όρος χρησιμοποιείται όταν πρόκειται για την ηλεκτρονική παραγγελία υλικών αγαθών που μπορούν να παραδοθούν μόνο με παραδοσιακούς φυσικούς τρόπους όπως είναι το ταχυδρομείο.

Άμεσο είναι το ηλεκτρονικό εμπόριο που περιλαμβάνει παραγγελία, πληρωμή και παράδοση άυλων αγαθών και υπηρεσιών. Η πληρωμή των υπηρεσιών αυτών γίνεται είτε με πιστωτικές κάρτες είτε με ηλεκτρονικό χρήμα.

1.1.2. Ηλεκτρονικό Εμπόριο με χρήση Φορητών Συσκευών (m-commerce)

Το ηλεκτρονικό εμπόριο με χρήση φορητών συσκευών, όπως υποδηλώνει και το όνομά του, αποτελεί εξειδικευμένη υποκατηγορία της έννοιας του ηλεκτρονικού εμπορίου. Πρόκειται για ηλεκτρονικές συνδιαλλαγές, οι οποίες λαμβάνουν χώρα με

χρήση κάποιας φορητής συσκευής, όπως κινητού τηλεφώνου, προσωπικού ψηφιακού βοηθού (personal digital assistant – pda) ή υπολογιστή χειρός (palmtop).

1.1.3. Ψηφιακή Επεξεργασία Εικόνας – Εικονοστοιχείο

Ως ψηφιακή εικόνα ορίζεται ως μία συνάρτηση δύο μεταβλητών, $f(x,y)$, όπου x και y είναι οι χωρικές συντεταγμένες, και η τιμή της f για οποιοδήποτε ζεύγος συντεταγμένων (x,y) ονομάζεται ένταση ή επίπεδο γκριζου (grey level) της εικόνας σε εκείνο το σημείο. Όταν τα x , y και η τιμή της f αποτελούν πεπερασμένες, διακριτές ποσότητες, τότε η εικόνα ονομάζεται ψηφιακή.

Ο τομέας της ψηφιακής επεξεργασίας φωτογραφίας (digital image processing) αναφέρεται στην επεξεργασία ψηφιακών εικόνων με χρήση ηλεκτρονικού υπολογιστή. Άξιζει να σημειωθεί ότι η ψηφιακή φωτογραφία αποτελείται από πεπερασμένο αριθμό στοιχείων, καθένα από τα οποία έχει τόσο συγκεκριμένη θέση στο χώρο όσο και τιμή. Τα στοιχεία αυτά ονομάζονται εικονοστοιχεία (pixel elements - pixels).

Η όραση αποτελεί την πλέον αναπτυγμένη αίσθηση του ανθρώπινου οργανισμού, συνεπώς δεν προκαλεί έκπληξη το γεγονός ότι οι εικόνες διαδραματίζουν τον σημαντικότερο ρόλο στην ανθρώπινη αντίληψη. Παρ'όλα αυτά, η επεξεργασία ψηφιακής εικόνας δεν περιορίζεται στο οπτικό εύρος του ηλεκτρομαγνητικού φάσματος που γίνεται αντιληπτό από τον άνθρωπο, αλλά το καλύπτει σχεδόν ολόκληρο. Λειτουργίες ψηφιακής επεξεργασίας εικόνας λαμβάνουν χώρα από το επίπεδο ακτινοβολίας γάμμα μέχρι και αυτό της ραδιο-ακτινοβολίας ενώ μπορούν να εφαρμοστούν σε εικόνες προερχόμενες από πηγές που δεν είναι σύνηθες για τους ανθρώπους να συνδέουν με φωτογραφίες. Τέτοιες εμπεριέχουν υπέρηχους, ηλεκτρονική μικροσκοπία και φωτογραφίες δημιουργημένες από ηλεκτρονικούς υπολογιστές. Τουτέστιν, η ψηφιακή επεξεργασία εικόνας καλύπτει έναν ευρύτατο τομέα εφαρμογών.

1.2. Συστήματα Ηλεκτρονικού Εμπορίου

Αρχικά, ως σύστημα ηλεκτρονικού εμπορίου ορίζονταν οποιοδήποτε σύστημα διαχείρισης εμπορικών συναλλαγών έκανε χρήση τεχνολογιών όπως Ηλεκτρονική Ανταλλαγή Δεδομένων (Electronic Data Interchange - EDI) και Ηλεκτρονική Μεταφορά Πόρων (Electronic Funds Transfer - EFT), οι οποίες πρωτοπαρουσιάστηκαν το 1970. Οι συγκεκριμένες τεχνολογίες επέτρεπαν την ηλεκτρονική μεταφορά εμπορικών εγγράφων όπως τιμολόγια ή αποδείξεις. Με την πάροδο των ετών και κυρίως με την έλευση του διαδικτύου και την αποδοχή των πιστωτικών καρτών και των μηχανημάτων αυτόματων συναλλαγών από το ευρύ κοινό τα συστήματα ηλεκτρονικού εμπορίου γνώρισαν μεγάλη άνθηση και αποτέλεσαν το αντικείμενο υλοποίησης ποικίλων τεχνολογιών. Έτσι, υπήρξε προσθήκη τεχνολογιών Εξόρυξης Δεδομένων (Data Mining) ή Αποθήκευσης Δεδομένων (Data Warehousing) και λίγο μετά την υλοποίηση του πρώτου συστήματος αγορών μέσω διαδικτύου, ήρθαν να προστεθούν Πρωτόκολλα Ασφάλειας και ευρυζωνικές υπηρεσίες.

Σήμερα τα συστήματα ηλεκτρονικού εμπορίου βρίσκουν εφαρμογές σε διάφορους τομείς. Επιγραμματικά αναφέρονται οι παρακάτω:

- Αγορές μέσω διαδικτύου και εντοπισμός παραγγελιών
- Τραπεζικές συναλλαγές μέσω διαδικτύου
- Εγχώρια και διεθνή συστήματα πληρωμών
- Λογισμικό Καλαθιού Αγορών (shopping cart software)
- Ηλεκτρονικά Εισιτήρια
- Ηλεκτρονικές Μικρές Αγγελίες

Από τη δεκαετία του 1990 και μετά οι εφαρμογές ηλεκτρονικού εμπορίου επεκτάθηκαν και σε πλατφόρμες φορητών συσκευών, αξιοποιώντας τεχνολογία που πηγάζει από αυτές, όπως για παράδειγμα προσφορά υπηρεσιών εξαρτώμενων από την τοποθεσία (location based services) .

Κάθε μία από τις παραπάνω εφαρμογές όμως, διακατέχεται από μία σειρά μειονεκτημάτων, ειδικά όταν υλοποιείται με απώτερο σκοπό τη χρήση σε κάποια

φορητή συσκευή. Τέτοια προβλήματα μπορεί να είναι τόσο σχεδιαστικής φύσεως όσο και λειτουργικής. Κύρια πηγή προβλημάτων αποτελούν οι περιορισμένες δυνατότητες των φορητών συσκευών σε επίπεδο λογισμικού και υλικού [Tarasewich et al., 2002]). Επίσης, η έλλειψη σε βάθος παραμετροποίησης της μορφής της γραφικής επιφάνειας χρηστών οδηγεί, είτε σε ελλιπείς ή δύσχρηστες καθολικές εφαρμογές, είτε σε εφαρμογές απευθυνόμενων σε ένα πολύ περιορισμένο κοινό. Τέλος, ας σημειωθεί και το γεγονός ότι γίνεται αναφορά σε πρόσφατες τεχνολογικές εξελίξεις, οι οποίες δεν έχουν ακόμα χρησιμοποιηθεί ευρέως και η ένταξή τους στην καθημερινότητα του κοινού δεν έχει οριστικοποιηθεί. Έτσι, υπάρχει συνεχής μεταβολή κι εξέλιξη αυτών, γεγονός που δημιουργεί συγκεκριμένες σχεδιαστικές αδυναμίες, αφού σύγχρονες μέθοδοι και τεχνολογίες καθίστανται παρωχημένες πριν ακόμα την ολοκλήρωση της υλοποίησής τους.

Τα προαναφερθέντα προβλήματα καλείται να αντιμετωπίσει το σύστημα που σχεδιάζεται στα πλαίσια αυτής της διπλωματικής εργασίας.

2. ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

Στα πλαίσια της ανάπτυξης της ευφυούς φορητής πλατφόρμας ηλεκτρονικού εμπορίου, που υλοποιείται σε αυτή τη διπλωματική εργασία έγινε χρήση συγκεκριμένων τεχνικών, μεθόδων και τεχνολογίας που αφορούν τους τομείς επεξεργασίας εικόνας, όρασης υπολογιστών και κωδικοποίησης δεδομένων. Στο κεφάλαιο αυτό παρουσιάζονται οι κυριότερες αλγοριθμικές λειτουργίες που εκτελούνται από το σχεδιασθέν σύστημα, όπως αυτές παρουσιάστηκαν από την ευρύτερη ακαδημαϊκή κι ερευνητική κοινότητα κι αναλύονται οι λόγοι για τους οποίους τελικά επιλέχτηκαν έναντι εναλλακτικών. Τέλος, αναφέρονται κι αιτιολογούνται οι όποιες τροποποιήσεις πραγματοποιήθηκαν κατά την ένταξη των ανωτέρω μεθόδων στο υλοποιηθέν σύστημα.

2.1. Μέθοδος Σύγκρισης Ομοιότητας Εικόνων

Η εύρεση μιας εύρωστης και αποτελεσματικής μεθόδου σύγκρισης εικόνων αποτέλεσε ένα από τα κύρια αντικείμενα έρευνας της τελευταίας δεκαετίας στον τομέα της επεξεργασίας εικόνας. Η συνήθης πρακτική που ακολουθείται από την πλειονότητα των προτεινόμενων μεθόδων βασίζεται στην εξαγωγή των ιστογραμμάτων χρώματος από τις φωτογραφίες και τη μεταξύ τους σύγκριση χρησιμοποιώντας κάποια μετρική, όπως για παράδειγμα την Ευκλείδεια απόσταση ή την μετρική L1 (L1 metric) της Taxicab γεωμετρίας [R. Brunelli και O. Mich]. Αυτό αιτιολογείται κυρίως από την ευκολία υπολογισμού των ιστογραμμάτων μιας εικόνας από ηλεκτρονικό υπολογιστή. Σε αυτό προστίθεται το γεγονός ότι οι αλλαγές στο ιστόγραμμα όταν έχουμε μετακίνηση της οπτικής γωνίας από την οποία τραβήχτηκε η φωτογραφία είναι αμελητέες και τέλος διότι είναι στατιστικά ορθή η υπόθεση ότι διαφορετικές εικόνες έχουν διαφορετικά ιστογράμματα.

Το βασικό μειονέκτημα της παραπάνω μεθόδου έγκειται στο γεγονός ότι στα ιστογράμματα δεν αποθηκεύονται χωρικές πληροφορίες, παρά μόνο χρωματικές. Έτσι, φωτογραφίες με παρόμοια ιστογράμματα μπορεί να στην πράξη να διαφέρουν

σε μεγάλο βαθμό, λόγω της απόκλισης που παρουσιάζουν στην χωρική διάταξη των χρωμάτων τους, όπως φαίνεται και στην εικόνα 2.1.1.



Εικόνα 2.1.1 : Δύο εικόνες με παρόμοια ιστογράμματα χρώματος

Τα μειονεκτήματα αυτά πρακτικά εξαλείφει η μέθοδος σύγκρισης εικόνων με βάση τα συζευγμένα ιστογράμματα εικόνων [Greg Pass και Ramin Zabih, 1999], η οποία και χρησιμοποιήθηκε στην υλοποίηση του συστήματος που αποτελεί θέμα της συγκεκριμένης διπλωματικής. Σύμφωνα με την παραπάνω μέθοδο, δημιουργούνται πολυδιάστατα ιστογράμματα για κάθε εικόνα, τα οποία εμπεριέχουν τόσο χρωματική όσο και χωρική πληροφορία. Πιο συγκεκριμένα, το ιστόγραμμα περιέχει τα κάτωθεν τοπικά χωρικά στοιχεία για κάθε εικονοστοιχείο προσθετικά στις πληροφορίες χρώματος:

- A) Ένταση Ακμών (edge density)
- B) Ιδιοσυστασία (texturedness)
- Γ) Βαθμίδα Κλίσης (gradient magnitude)
- Δ) Βαθμός Εικονοστοιχείου (pixel rank)

,όπου για τυχαίο εικονοστοιχείο της εικόνας ορίζονται:

- Ένταση Ακμών: Το ποσοστό των γειτονικών εικονοστοιχείων στο εν λόγω εικονοστοιχείο που αποτελούν ακμές.

- **Ιδιοσυστασία:** Το ποσοστό των γειτονικών εικονοστοιχείων στο εν λόγω εικονοστοιχείο που η έντασή τους διαφέρει περισσότερο από μία τιμή κατωφλίου ορισμένη από εμάς.
- **Βαθμίδα Κλίσης:** Το μέτρο του πόσο γρήγορα αλλάζει η ένταση στην κατεύθυνση της μεγαλύτερης αλλαγής.
- **Βαθμός:** Το ποσοστό των γειτονικών εικονοστοιχείων στο εν λόγω εικονοστοιχείο που η έντασή τους είναι μικρότερη αυτού.

Πειραματικά αποτελέσματα των ερωτημάτων που τέθηκαν σε βάση 210,000 φωτογραφιών ανάλυσης 192x128 εικονοστοιχείων κατέδειξαν τη σταθερότητα, ταχύτητα και αποτελεσματικότητα της μεθόδου. Αξιοσημείωτο είναι επίσης το ποσοστό επιτυχών συγκρίσεων που προκύπτει κάνοντας χρήση μόνο κάποιων εκ των χωρικών τοπικών στοιχείων. Πιο συγκεκριμένα, στην περίπτωση χρήσης μόνο της πληροφορίας χρώματος και της βαθμίδας κλίσης παρατηρείται απόκλιση 4% από τη μέγιστη απόδοση της τεχνικής (η οποία είναι 93%), ενώ παράλληλα το κέρδος σε χρόνο εκτέλεσης τριπλασιάζεται.

Τα αποτελέσματα αυτά οδήγησαν στο σχεδιασμό του τμήματος συγκρίσεων των φωτογραφιών του συστήματος με βάση τα ιστογράμματα που περιείχαν πληροφορίες χρώματος και τιμές της βαθμίδας κλίσης. Επιτυγχάνεται με αυτό τον τρόπο ένα σταθερό κι αποτελεσματικό σύστημα, το οποίο παράλληλα παραμένει αρκετά ταχύ ώστε να ανταποκριθεί στις αυξημένες σε ταχύτητα απαιτήσεις τις εφαρμογής, που απορρέουν από τη σύγκριση εικόνων πολύ μεγαλύτερης ανάλυσης και της ανάγκης γρήγορης εμφάνισης αποτελεσμάτων στον χρήστη.

2.2. Συστήματα Σύστασης

Τα συστήματα σύστασης χρησιμοποιούνται κυρίως σε πλατφόρμες ηλεκτρονικού εμπορίου, με σκοπό να δοθούν στους χρήστες τους περισσότερες επιλογές ή και να τους προταθεί ευρύτερη ποικιλία προϊόντων. Απώτερος στόχος είναι η βελτίωση των πωλήσεών τους, αν πρόκειται για ιστοσελίδες έμμεσου ηλεκτρονικού εμπορίου, ή της επιτυχούς παροχής υπηρεσιών, αν πρόκειται για ιστοσελίδες άμεσου ηλεκτρονικού εμπορίου. Σε κάθε περίπτωση, ένα σύστημα προτάσεων αυξάνει τη

χρησιμότητα της εκάστοτε ιστοσελίδας οδηγώντας σε μεγιστοποίηση της απόδοσης του συστήματος.

Με βάση τον τρόπο με τον οποίο παράγουν τις συστάσεις προς τους χρήστες, τα συστήματα σύστασης διαχωρίζονται σε [Gediminas Adomavicius και Alexander Tuzhilin, 2005]:

- A) Συστάσεων Περιεχομένου
- B) Συνεργατικών Συστάσεων
- Γ) Υβριδικής Προσέγγισης

Πιο αναλυτικά, τα συστήματα **συστάσεων περιεχομένου** επιλέγουν τις συστάσεις που παρουσιάζουν στο χρήστη, βασιζόμενα στις προηγούμενες επιλογές, βαθμολογήσεις και αναζητήσεις του ίδιου. Τα κριτήρια επιλογής βασίζονται σε τεχνικές Ανάκτησης Πληροφορίας (Information Retrieval), ενώ χρήση των εν λόγω συστημάτων γίνεται κυρίως, αλλά όχι μόνο σε περιπτώσεις όπου τα δεδομένα είναι μορφής κειμένου. Εκτός των τεχνικών ανάκτησης πληροφορίας, έχουν προταθεί, αναπτύσσονται ή χρησιμοποιούνται και άλλες τεχνικές, όπως δέντρα αποφάσεων, νευρωνικά δίκτυα και συσταδοποίησης (clustering).

Τα συστήματα αυτής της ομάδας όμως, μειονεκτούν έναντι άλλων κατηγοριών διότι παρουσιάζουν υπερεξειδίκευση. Χαρακτηρίζονται δηλαδή από αδυναμία σύστασης προϊόντων ή υπηρεσιών, για τις οποίες οι χρήστες δεν έχουν πραγματοποιήσει αναζήτηση ή βαθμολογήσει. Παράλληλα, σε αρκετές περιπτώσεις είναι αμφισβητήσιμη η χρησιμότητα συστάσεων, οι οποίες όχι μόνο προσεγγίζουν, αλλά σχεδόν ταυτίζονται με την αρχική αναζήτηση. Για την αποφυγή των παραπάνω περιορισμών τα φίλτρα ενός τέτοιου συστήματος ελέγχουν τον πλήρη κατάλογο των διαθέσιμων επιλογών, απορρίπτοντας τόσο τις πολύ αδιάφορες, όσο και τις υπερβολικά όμοιες, σε βαθμό ταύτισης, συστάσεις.

Τα συστήματα **συνεργατικών συστάσεων** εν αντιθέσει, αποσκοπούν στην πρόβλεψη πιθανά ενδιαφέροντων συστάσεων με βάση βαθμολογήσεις, αναζητήσεις και επιλογές άλλων χρηστών. Οι αλγόριθμοι που χρησιμοποιούνται από τα συστήματα αυτά είναι δύο ειδών, μνήμης και μοντέλου. Οι αλγόριθμοι μνήμης είναι ευρετικές μέθοδοι

(heuristics), οι οποίες υπολογίζουν μία τιμή προβλεπόμενης εμφάνισης για κάθε ένα από τα στοιχεία που ήδη έχουν επιλεγεί από άλλους χρήστες στο παρελθόν. Οι τιμές αυτές υπολογίζονται συνήθως με χρήση αθροίσματος τιμών με βάρη. Συγκρίνοντας μέσω κάποιας μετρικής απόστασης, τις παραπάνω τιμές με την αντίστοιχη τιμή των αποτελεσμάτων της αναζήτησης του χρήστη μία δεδομένη στιγμή, παρουσιάζουν τα N πιο όμοια αποτελέσματα ως συστάσεις.

Αντίθετα, οι αλγόριθμοι μοντέλου χρησιμοποιούν τις τιμές προβλεπόμενης εμφάνισης με σκοπό τη δημιουργία ενός πιθανοτικού μοντέλου, με βάση το οποίο υπολογίζονται οι τιμές προβλεπόμενης εμφάνισης για όσα στοιχεία δεν έχουν αναζητηθεί ή βαθμολογηθεί ακόμη από τους χρήστες.

Σε κάθε περίπτωση, ανεξάρτητα από τον αλγόριθμο που χρησιμοποιούν, τα συστήματα συνεργατικών συστάσεων χαρακτηρίζονται από σποραδικότητα (sparsity). Αυτό συμβαίνει, διότι σε όλα σχεδόν τα συστήματα ο αριθμός των προβλέψεων που πρέπει να γίνουν είναι δυσανάλογα μεγάλος, σε σχέση με τις βαθμολογήσεις ή αναζητήσεις σε προϊόντα που έχουν ήδη γίνει. Απαιτείται επομένως, πολύ καλό μοντέλο για την απόδοση τιμών προβλεπόμενης εμφάνισης. Επίσης, χρήστες με ιδιαίτερες προτιμήσεις ή αναζητήσεις δεν είναι εύκολο να ικανοποιηθούν από τις συστάσεις ενός τέτοιου συστήματος, λόγω παρουσίας συστάσεων που απευθύνονται στην πλειοψηφία των χρηστών.

Για την αντιμετώπιση των παραπάνω ζητημάτων γίνεται χρήση **υβριδικών μεθόδων**. Κατά καιρούς έχουν προταθεί διάφορες τέτοιες μέθοδοι, οι οποίες διαφέρουν στον τρόπο που συνδυάζουν μεθόδους. Έτσι, μπορεί να υλοποιούν ξεχωριστά και παράλληλα το συνεργατικό μοντέλο και το μοντέλου περιεχομένου, συνδυάζοντας στη συνέχεια τα αποτελέσματά τους. Επίσης, είναι δυνατή η ενσωμάτωση κάποιων χαρακτηριστικών του συνεργατικού μοντέλου σε ένα σύστημα συστάσεων με βάση το περιεχόμενο. Το ίδιο είναι εφικτό και αντίστροφα, δηλαδή η ενσωμάτωση χαρακτηριστικών σύστασης με βάση το περιεχόμενο σε ένα συνεργατικό σύστημα. Τέλος, συχνά κατασκευάζεται ένα συνδυαστικό, ενοποιητικό μοντέλο, το οποίο υλοποιεί ενοποιημένα τις δύο παραπάνω τεχνικές.

Με βάση τις παραπάνω τρεις τεχνικές, πραγματοποιούνται όλες οι υλοποιήσεις συστημάτων σύστασης, οι οποίες μπορούν να κατηγοριοποιηθούν [Schafer et al.,1999] ως εξής:

- A) Ομοίων Αντικειμένων
- B) Πλοήγησης-Αναζήτησης
- Γ) Ηλεκτρονικού Ταχυδρομείου
- Δ) Γραπτών Σχόλιων
- E) N-Καλύτερα
- ΣΤ) Ταξινομημένης Σειράς Αναζήτησης

,όπου με το σύστημα:

- **Ομοίων Αντικειμένων:** Προτείνονται στο χρήστη αντικείμενα ή επιλογές αναζήτησης όμοιες με τις μέχρι τώρα επιλογές του. Σκοπός είναι να προταθούν στο χρήστη προϊόντα/υπηρεσίες, σχετικά με τις μέχρι τώρα αναζητήσεις του, τα οποία είτε δεν είχε προσέξει, είτε δε γνώριζε. Οι υλοποιήσεις των εν λόγω συστημάτων από διάφορες γνωστές ιστοσελίδες επιτρέπει την περαιτέρω εξατομίκευση, ανάλογα με τις ανάγκες του κάθε χρήστη.
- **Πλοήγησης-Αναζήτησης:** Δίνεται στο χρήστη η δυνατότητα λεπτομερειακής επολογής των παραμέτρων αναζήτησης των αντικειμένων που τον ενδιαφέρουν. Επιπλέον, όλες οι συστάσεις του εν λόγω συστήματος ταξινομούνται, κατηγοριοποιούνται και παρουσιάζονται άμεσα και δομημένα στο χρήστη με τη μορφή άμεσων συνδέσμων (links). Υποβοηθάται με αυτό τον τρόπο κάθε μορφή ηλεκτρονικού εμπορίου, διότι οι χρήστες αποκτούν τη δυνατότητα μεγαλύτερου φιλτραρίσματος των επιλογών τους.
- **Ηλεκτρονικού Ταχυδρομείου:** Προτείνονται μέσω ηλεκτρονικού ταχυδρομείου επιλογές στους χρήστες, όμοιες σε όλους, με βάση τα μέχρι τώρα στατιστικά στοιχεία της πλατφόρμας. Με αυτό τον τρόπο, γίνονται άμεσα γνωστές στην πλειοψηφία των χρηστών οι νέες προσθήκες στην πλατφόρμα και παρατηρείται αύξηση της αποδοτικότητας των προσφερόμενων υπηρεσιών. Αυτό συμβαίνει διότι ο χρήστης αποκτά τη

δυνατότητα να πληροφορείται πρώτος νέες επιλογές, ενώ παράλληλα μπορεί να παρακολουθεί πιο στενά προϊόντα ή υπηρεσίες που τον ενδιαφέρουν.

- **Γραπτών Σχολίων:** Δίνεται η δυνατότητα σχολιασμού των επιλογών και προϊόντων της πλατφόρμας από τους χρήστες της καθώς βαθμολόγησης. Έτσι, καθίσταται ευκολότερη η επιτυχία στις αναζητήσεις των χρηστών, διότι μπορούν διαβάζοντας απλά τα σχόλια που έχουν γίνει ή με μια απλή ματιά στη βαθμολογία να καταλήξουν αν το προϊόν ή οι υπηρεσίες που αναζητούν τους ενδιαφέρουν. Τέλος, αυξάνεται η πιθανότητα ανατροφοδότησης του συστήματος με συστάσεις, μέσα από συστάσεις που τυχόν γίνονται στα σχόλια των ίδιων των χρηστών.
- **N-Καλύτερα:** Εμφανίζονται μαζί με τα αποτελέσματα που ζητήθηκαν από το χρήστη και τα καλύτερα προϊόντα στην κατηγορία όπου έγινε η αναζήτηση. Η χρήση ενός τέτοιου συστήματος προϋποθέτει τη συλλογή μεγάλου μέρους στατιστικών, που περιγράφουν τις συνήθειες αναζητήσεις κι επιλογές των χρηστών της πλατφόρμας. Επιτυγχάνεται όμως σημαντική αύξηση της αποδοτικότητας. Αυτό συμβαίνει διότι συγκεντρώνονται οι δημοφιλέστερες επιλογές μαζί, καθιστώντας ευκολότερες τις αναζητήσεις των χρηστών. Ταυτόχρονα η απόδοση αυξάνεται επειδή γίνονται συστάσεις στους χρήστες για επιλογές που στατιστικά υπάρχει μεγάλη πιθανότητα να τους ενδιαφέρουν.
- **Ταξινομημένης Σειράς Αναζήτησης:** Τα αποτελέσματα της αναζήτησης του χρήστη εμφανίζονται με σειρά προτεραιότητας, ανάλογη στατιστικών στοιχείων της πλατφόρμας. Ενώ στην περίπτωση των N-Καλύτερων συστημάτων οι συστάσεις περιορίζονται σε ένα μέγιστο αριθμό, στην περίπτωση της Ταξινομημένης Σειράς Αναζήτησης δίνεται η επιλογή στο χρήστη να συνεχίσει την αναζήτησή του σε αποτελέσματα που είναι στατιστικώς πολύ πιθανό να τον ενδιαφέρουν.

Για την ευφυή φορητή πλατφόρμα αυτής της εργασίας, επιλέχτηκε σύστημα προτάσεων όμοιων αντικειμένων. Με την ενσωμάτωσή του επετεύχθηκε μεγιστοποίηση των δυνατοτήτων εξατομίκευσης των προτάσεων ανάλογα με τις επιλογές αναζήτησης του χρήστη. Αυξήθηκε έτσι η έκθεση των χρηστών σε ενδιαφέροντα αποτελέσματα, συνδράμοντας θετικά στη βελτίωση της αποδοτικότητας της πλατφόρμας.

3. ΜΕΘΟΔΟΣ

Στο κεφάλαιο αυτό αναλύονται οι βασικότεροι αλγόριθμοι που χρησιμοποιήθηκαν κατά την υλοποίηση του συστήματος. Κάθε βήμα του αλγορίθμου αναλύεται και παράλληλα δίνεται όπου χρειάζεται το αντίστοιχο θεωρητικό υπόβαθρο.

Πιο συγκεκριμένα, αναφέρονται οι αλγόριθμοι και τεχνικές που χρησιμοποιήθηκαν για την εύρεση των επιπλωμένων χώρων, του εκάστοτε τύπου πατώματος καθώς και του χρώματος των εξωτερικών τοίχων κάθε οικίας, της οποίας φωτογραφίες αποθηκεύονται από το χρήστη στο δίκτυο. Αναλύεται επίσης η μέθοδος σύγκρισης εικόνων που ενσωματώθηκε στο σύστημα. Τέλος, λεπτομερής αναφορά γίνεται και στη μέθοδο που χρησιμοποιήθηκε για αποστολή πολυμεσικών αρχείων από το σύστημα πελάτη στο σύστημα του εξυπηρετητή.

3.1. Εύρεση Επιπλωμένου Χώρου από Εικόνα

Ένα πρώτο χαρακτηριστικό της πλατφόρμας που υλοποιήθηκε αποτελεί η δυνατότητα εξακρίβωσης αν ο χώρος, που απεικονίζει μία εικόνα εσωτερικού χώρου μιας οικίας, είναι επιπλωμένος. Το σύνολο των εν λόγω φωτογραφιών βρίσκεται αποθηκευμένο στο σύστημα της βάσης δεδομένων.

Πρόκειται για εφαρμογή δύο κλασικών μεθόδων επεξεργασίας εικόνας. Οι δύο αυτές μέθοδοι είναι η κατάτμηση εικόνας σε περιοχές και η ανίχνευση ακμών, για την πραγμάτωση των οποίων υλοποιήθηκαν ένας τυπικός αλγόριθμος ανάπτυξης περιοχών (region growing algorithm) κι ο αλγόριθμος ανίχνευσης ακμών του Canny, αντίστοιχα.

Η **κατάτμηση** μιας εικόνας αναφέρεται στην τμηματοποίησή της σε πολλαπλές ομάδες εικονοστοιχείων, τα οποία αποκαλούνται υπερ-εικονοστοιχεία (super pixels). Σκοπός της κατάτμησης είναι να απλοποιηθεί ή/και να μετατραπεί η φωτογραφία σε τέτοια μορφή, ούτως ώστε να είναι πιο ουσιαδές και εύκολο να αναλυθεί. Η τμηματοποίηση εικόνας χρησιμοποιείται συνήθως για τον εντοπισμό αντικειμένων και ορίων, όπως γραμμές ή καμπύλες, σε μία φωτογραφία. Πιο συγκεκριμένα,

αποτελεί τη διαδικασία κατά την οποία κάθε εικονοστοιχείο μιας εικόνας λαμβάνει μία ετικέτα (label), με σκοπό τελικά όλα τα εικονοστοιχεία της ίδιας ετικέτας να μοιράζονται κοινά οπτικά χαρακτηριστικά.

Το αποτέλεσμα της κατάτμησης εικόνων είναι είτε ένα σύνολο από περιοχές, οι οποίες καλύπτουν στο σύνολό της την εικόνα, είτε μία ομάδα εξωτερικών καμπύλων (contours), οι οποίες έχουν εξαχθεί από τη φωτογραφία. Κάθε ένα από τα εικονοστοιχεία που ανήκουν σε μία περιοχή είναι όμοια με τα υπόλοιπα του ίδιου τμήματος με βάση κάποιο χαρακτηριστικό ή κάποια υπολογίσιμη ιδιότητα, όπως χρώμα, ένταση ή υφή. Γειτονικές περιοχές παρουσιάζουν σημαντικές διαφορετικές με βάση το ίδιο χαρακτηριστικό ή χαρακτηριστικά.

Πλήθος αλγορίθμων γενικού σκοπού και τεχνικών έχουν αναπτυχθεί για την επίλυση του προβλήματος κατάτμησης εικόνων. Γεγονός αποτελεί ότι δεν υπάρχει γενική λύση στο πρόβλημα αυτό. Συνεπώς, πολλές τεχνικές συχνά πρέπει να συνδυαστούν με γνώση σχετική με τον τομέα χρήσης τους ώστε να επιλυθεί αποδοτικά το πρόβλημα για τον τομέα αυτό. Ενδεικτικά αναφέρονται παρακάτω κάποιες από αυτές τις κατηγορίες μεθόδων:

- Μέθοδοι Συστάδας (clustering methods)
- Τεχνικές Συμπίεσης
- Μέθοδοι Ιστογράμματος
- Ανίχνευση Ακμών
- Μέθοδοι Ανάπτυξης Περιοχών
- Μέθοδοι Μερικών Διαφορικών Εξισώσεων

Όπως ήδη έχει αναφερθεί, στην υλοποίηση για την εν λόγω διπλωματική εργασία χρησιμοποιήθηκε μία τυπική μέθοδος ανάπτυξης περιοχών, η οποία αναλύεται παρακάτω.

Η **ανίχνευση ακμών** αποτελεί ένα θεμελιώδες εργαλείο στην ψηφιακή επεξεργασία εικόνας και στην όραση υπολογιστών (computer vision), ειδικά στους τομείς της ανίχνευσης γνωρισμάτων (feature detection) ή της εξαγωγής αυτών (feature

extraction). Σκοπός είναι η αναγνώριση σημείων σε μία ψηφιακή φωτογραφία, στα οποία η φωτεινότητα ή κάποιο άλλο χαρακτηριστικό της παρουσιάζει ασυνέχεια.

Κίνητρο για την εύρεση των ακμών μίας φωτογραφίας αποτέλεσε το γεγονός ότι απότομες μεταβολές ή ασυνέχειες σε χαρακτηριστικά της είναι πολύ πιθανό να αντιστοιχούν σε:

- Ασυνέχειες βάθους
- Ασυνέχειες στον προσανατολισμό επιφανειών
- Αλλαγές στις ιδιότητες υλικών
- Διαφοροποιήσεις στο φωτισμό

Ιδανικά, το αποτέλεσμα της εφαρμογής μιας μεθόδου εύρεσης ακμών μπορεί να οδηγήσει σε ένα σύνολο ενωμένων καμπυλών, οι οποίες καταδεικνύουν τα σύνορα αντικειμένων, τα όρια σημαδιών στις επιφάνειες καθώς και τις καμπύλες που αντιστοιχούν σε ασυνέχειες του προσανατολισμού επιφανειών. Τουτέστιν, εφαρμογή ενός αλγόριθμου ανίχνευσης ακμών σε μία εικόνα μειώνει σημαντικά τον όγκο δεδομένων που πρέπει να επεξεργαστούν και απορρίπτει μέρος της πληροφορίας ως αδιάφορη, ενώ ταυτόχρονα διατηρεί τις σημαντικές δομικές ιδιότητες της φωτογραφίας. Αν η ανίχνευση ακμών είναι επιτυχής, τα επόμενα στάδια στην επεξεργασία της εικόνας πολύ πιθανόν να είναι σημαντικά απλούστερα.

Δυστυχώς, κάτι τέτοιο δεν είναι εφικτό, μιας και καθίσταται συχνά αδύνατο να ληφθούν ιδανικά οι ακμές από πραγματικές φωτογραφίες μεγάλης πολυπλοκότητας. Ακμές που εξάγονται από τέτοιες φωτογραφίες συχνά συνοδεύονται από κατακερματισμό, δηλαδή από το γεγονός ότι οι καμπύλες των ακμών δεν είναι ενωμένες λόγω απουσίας τμημάτων ακμών. Πιθανή είναι και η ύπαρξη λανθασμένων ακμών σε παρεμφερείς περιπτώσεις, οι οποίες δεν αντιστοιχούν σε ενδιαφέροντα φαινόμενα, όπως αυτά προκύπτουν από την εικόνα. Το σύνολο των προαναφερθέντων δυσκολιών δυσχαιρένει σημαντικά την μετέπειτα ερμηνεία των εικόνων.

Οι ακμές μπορούν να κατηγοριοποιηθούν είτε ως εξαρτώμενες από την οπτική γωνία, είτε ως ανεξάρτητες της οπτικής γωνίας. Οι ανεξάρτητες από την οπτική γωνία ακμές

αντιστοιχούν σε εγγενείς ιδιότητες τρισδιάστατων αντικειμένων, όπως σημάδια ή σχήμα της επιφάνειάς τους, ενώ οι εξαρτώμενες αλλάζουν ενόσω αλλάζει και η οπτική γωνία. Όπως γίνεται αντιληπτό, αυτές αντιστοιχούν στη γεωμετρία της απεικονιζόμενης σκηνής, όπως για παράδειγμα αντίκειμενα που παρεμβάλλονται. Παραδείγματος χάριν, μία τυπική ακμή να είναι το σύνορο μεταξύ ενός συνόλου κόκκινου κι ενός συνόλου κίτρινου χρώματος.

Υπάρχει πλήθος μεθόδων ανίχνευσης ακμών, αλλά η πλειοψηφία αυτών μπορεί να κατηγοριοποιηθεί σε δύο ομάδες:

- Βασιζόμενες σε αναζήτηση
- Βασιζόμενες σε περάσματα από το μηδέν (zero crossings)

Οι μέθοδοι της πρώτης κατηγορίας ανιχνεύουν ακμές υπολογίζοντας πρώτα κάποιο μέτρο της ισχύος της ακμής. Αυτό είναι συνήθως μιας πρώτης τάξης διαφορική έκφραση, όπως η βαθμίδα κλίσης. Στη συνέχεια, αναζητούν ένα τοπικό κατευθυντήριο μέγιστο της βαθμίδας κλίσης χρησιμοποιώντας μία υπολογισμένη εκτίμηση του τοπικού προσανατολισμού, που συνήθως πρόκειται για την κατεύθυνση της κλίσης.

Οι αλγόριθμοι της δεύτερης κατηγορίας αναζητούν περάσματα από το μηδέν σε μία δεύτερης τάξης διαφορική έκφραση υπολογισμένη από την εικόνα. Η έκφραση αυτή είναι η Λαπλασιανή ή μία μη-γραμμική διαφορική έκφραση.

Σχεδόν πάντα προηγείται της ανίχνευσης ακμών ένα στάδιο προ-επεξεργασίας ή αλλιώς εξομάλυνσης, συνήθως Γκαουσιανή εξομάλυνση, με σκοπό τη μείωση του θορύβου.

Οι μέθοδοι που έχουν δημοσιευτεί διαφέρουν κυρίως στους τύπους φίλτρων εξομάλυνσης που εφαρμόζουν και στον τρόπο που το μέτρο σύγκρισης της ισχύος των ακμών υλοποιείται από αυτές. Επίσης, μιας και οι περισσότερες μέθοδοι ανίχνευσης στηρίζονται στον υπολογισμό της κλίσης της εικόνας, συχνή διαφοροποίηση έγκειται και στον τύπο φίλτρων που χρησιμοποιούν για τον υπολογισμό της κλίσης στην κατεύθυνση των αξόνων.

Όπως ήδη έχει αναφερθεί, στην υλοποίηση για αυτή τη διπλωματική εργασία χρησιμοποιήθηκε η μέθοδος ανίχνευσης ακμών του Canny, η οποία ανήκει στην κατηγορία μεθόδων αναζήτησης. Ο τρόπος λειτουργίας της αναλύεται ενδελεχώς παρακάτω.

3.1.1. Αλγόριθμος Κατάτμησης Εικόνων μέσω Ανάπτυξης Περιοχών

Η ανάπτυξη περιοχών αποτελεί, όπως προαναφέρθηκε, μία μέθοδο κατάτμησης εικόνων βασισμένη στα εικονοστοιχεία της. Ενώ αρκετές τεχνικές για την τμηματοποίηση φωτογραφιών επικεντρώνονται στην εύρεση των ορίων κάθε περιοχής, η συγκεκριμένη τεχνική υπολογίζει την περιοχή απευθείας.

Πιο συγκεκριμένα, η μέθοδος διατυπώνεται μαθηματικά ως εξής:

$$(a) \cup_{i=1}^n R_i = R$$

$$(b) H R_i \text{ είναι μία συνεχής περιοχή} \quad , \text{ για } i = 1, 2, \dots, n$$

$$(c) R_i \cap R_j = \emptyset \quad , \text{ για όλα τα } i = 1, 2, \dots, n$$

$$(d) P(R_i) = ΑΛΗΘΕΣ \quad , \text{ για } i = 1, 2, \dots, n$$

$$(e) P(R_i \cup R_j) = ΨΕΥΔΕΣ \quad , \text{ για κάθε γειτονική περιοχή } R_i \text{ και } R_j$$

,όπου το $P(R_i)$ αποτελεί ένα λογικό κατηγορήμα ορισμένο στα k στοιχεία του συνόλου $P(R_k)$ και \emptyset είναι το κενό σύνολο.

- Η πρόταση (α) ορίζει ότι η κατάτμηση πρέπει να είναι πλήρης. Δηλαδή, κάθε εικονοστοιχείο της εικόνας να ανήκει σε μία περιοχή.
- Η πρόταση (β) απαιτεί τα στοιχεία της περιοχής να ενώνονται με κάποιο προκαθορισμένο τρόπο.
- Η πρόταση (γ) υποδηλώνει ότι οι περιοχές πρέπει να είναι ανεξάρτητες μεταξύ τους.
- Η πρόταση (δ) έχει να κάνει με τα κριτήρια που πρέπει να ικανοποιούνται από τα εικονοστοιχεία σε μία περιοχή. Για παράδειγμα, $P(R_i) = ΑΛΗΘΕΣ$ αν όλα τα εικονοστοιχεία του R_i έχουν την ίδια τιμή γκρίζου επιπέδου.

- Η πρόταση (e) υποδεικνύει ότι οποιεσδήποτε δύο περιοχές R_i και R_j είναι διαφορετικές μεταξύ τους με βάση το κατηγορημα P.

Ο αλγόριθμος του συστήματος που υλοποιεί την κατάτμηση εικόνας με τη μέθοδο της ανάπτυξης απαρτίζεται από δύο διακριτά στάδια, την **προ-επεξεργασία** της εικόνας εισόδου και την **εξαγωγή των κατατμημένων περιοχών** της εικόνας-δείγμα.

Ως είσοδό του δέχεται μία οποιασδήποτε εικόνα προς κατάτμηση των περιοχών της. Αρχικά, το στάδιο προ-επεξεργασίας μετατρέπει την εικόνα στην αντίστοιχη τόνων του γκρι (grayscale). Με αυτό τον τρόπο καθίσταται δυνατή η χρήση της μεθόδου ανάπτυξης περιοχών στο επόμενο στάδιο.

Το επόμενο και τελευταίο στάδιο, είναι η κατάτμηση της εικόνας κλίμακας των γκρι που προέρχεται από την προ-επεξεργασία της αρχικής φωτογραφίας με βάση τον αλγόριθμο ανάπτυξης περιοχών. Για την υλοποίηση της μεθόδου ανάπτυξης περιοχών χρησιμοποιήθηκε η προαναφερθείσα μέθοδος, λαμβάνοντας ως κριτήριο ένταξης σε μια περιοχή μια τιμή καταφλίου τόνου των γκρι. Τούτο το κριτήριο αποτέλεσε και την ανάγκη ύπαρξης του σταδίου προ-επεξεργασίας.

3.1.1.1 Στάδιο Προ-Επεξεργασίας Αλγόριθμου Ανάπτυξης Περιοχών

Το στάδιο αυτό δέχεται στην είσοδό του μία φωτογραφία. Μία εικόνα-δείγμα, η οποία πρόκειται να υποστεί επεξεργασία στις τιμές των εικονοστοιχείων της. Στο κεφάλαιο 5 (ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ) αναλύεται ο τρόπος επιλογής της. Γενικά, μπορούν να είναι κωδικοποιημένες με διάφορους τρόπους (format), με τις κωδικοποιήσεις jpeg, bitmap, gif και png να αποτελούν τις πιο συχνά χρησιμοποιούμενες μορφές αποθήκευσης εικόνας στις φορητές συσκευές. Μπορούν όμως να χρησιμοποιούν μόνο το χρωματικό μοντέλο RGB. Το μοντέλο αυτό επιλέχτηκε διότι αποτελεί το δημοφιλέστερο χρωματικό μοντέλο για τις αποθηκευμένες φωτογραφίες σε όλες τις φορητές συσκευές. Οι διαμορφώσεις της εικόνας και τα χρωματικά μοντέλα που χρησιμοποιούνται και αποτελούν δεκτές εισόδους του υλοποιηθέντος συστήματος δίνονται στα παραρτήματα Α και Β αντίστοιχα.

Ο λόγος ύπαρξης του σταδίου προεπεξεργασίας είναι η μετατροπή των χρωματικών χαρακτηριστικών της εικόνας που δέχεται το υποσύστημα στην είσοδό του σε μορφή, η οποία θα είναι αντιμετωπίσιμη από το επόμενο στάδιο του αλγορίθμου. Αυτό επιτυγχάνεται με μετατροπή της εικόνας σε εικόνα σε τόνους των γκρι.

Οι εικόνες σε τόνους του γκρι διαφέρουν από τις ασπρόμαυρες του ενός bit, οι οποίες στα πλαίσια της ψηφιακής επεξεργασίας φωτογραφίας συνίστανται από δύο μόνο χρώματα, το άσπρο και το μαύρο και καλούνται επίσης δυαδικές εικόνες. Εν αντιθέσει, οι γκριζας κλίμακας φωτογραφίες εμπεριέχουν όλους τους τόνους του γκριζου και καλούνται επίσης μονοχρωματικές, τονίζοντας έτσι την πλήρη απουσία οποιασδήποτε χρωματικής ποικιλίας.

Μέσω της μετατροπής αυτής καθίσταται εφικτή η δυνατότητα χρήσης στο επόμενο στάδιο μιας τιμής κατωφλίου τόνου των γκριζων, ως κριτήριο ένταξης ενός εικονοστοιχείου.

Ο αλγόριθμος μετατροπής της εικόνας σε μία νέα, τόνου των γκρι λειτουργεί ως ακολούθως:

Μετασχηματίζει την εικόνα εισόδου σε νέα, με τιμές στα εικονοστοιχεία της να αντιστοιχούν σε αποχρώσεις του γκριζου, αν αυτό είναι απαραίτητο. Το σύστημα στα επόμενα στάδια επεξεργάζεται μόνο τις μετασχηματισμένες από την προεπεξεργασία εικόνες. Παρ' όλα αυτά, η αρχική ανεπεξεργαστη εικόνα δε διαγράφεται και παραμένει στη βάση. Κι αυτό διότι η αρχική φωτογραφία χρησιμοποιείται σε άλλα υποσυστήματα για εξαγωγή διαφορετικών πληροφοριών.

3.1.1.2 Στάδιο Κατάτμησης Εικόνας Αλγόριθμου Ανάπτυξης Περιοχών

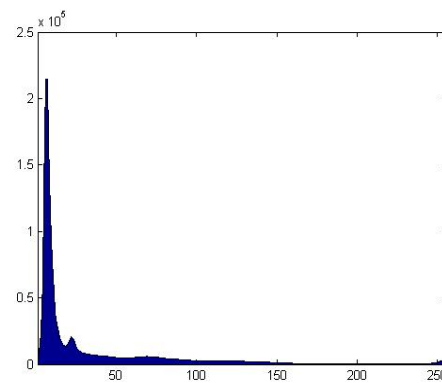
Σκοπός του σταδίου αυτού είναι η κατάτμηση της εικόνας, η οποία βρίσκεται σε μορφή κλίμακας του γκριζου πλέον, σε περιοχές όμοιων χαρακτηριστικών. Κριτήριο ένταξης του εκάστοτε εικονοστοιχείου σε μία περιοχή αποτελεί η τιμή της έντασης του γκριζου που έχει λάβει από το στάδιο προ-επεξεργασίας. Γενικά τα βήματα που ακολουθούν όλες οι υλοποιήσεις της συγκεκριμένης μεθόδου είναι δύο.

Το πρώτο βήμα είναι η επιλογή ενός συνόλου σημείων σποράς (seed points). Η επιλογή των σημείων αυτών βασίζεται σε κριτήρια ανάλογα του περιβάλλοντος που προβλέπεται να γίνει χρήση του αλγόριθμου. Παραδείγματα εικονοστοιχείων που επιλέγονται ως σημεία-σποράς είναι εικονοστοιχεία που ισαπέχουν με βάση κάποιο νοητό πλέγμα ή που έχουν τιμές έντασης μέσα σε ορισμένο εύρος. Η αρχικές περιοχές εκκινούν ως η ακριβής τοποθεσία αυτών των σημείων σποράς.

Στη συνέχεια, οι αρχικές περιοχές αυξάνουν σε μέγεθος κινούμενες από τα αρχικά σημεία προς τα γειτονικά εικονοστοιχεία, ανάλογα πάντα με τα κριτήρια ένταξης στην περιοχή που ο αλγόριθμος ακολουθεί. Τέτοια κριτήρια είναι η ένταση του εκάστοτε εικονοστοιχείου, η υφή ή το χρώμα. Σημαντικό ρόλο στην ανάπτυξη των περιοχών διαδραματίζουν και οι περαιτέρω πληροφορίες που εξάγονται από την εικόνα. Για παράδειγμα, αν ληφθεί ως κριτήριο ένταξης μία τιμή καταφλίου της έντασης των εικονοστοιχείων, γνώση του ιστογράμματος της φωτογραφίας θα μπορούσε να χρησιμοποιηθεί για την εύρεση της ιδανικής τιμής της.



(α)



(β)

Εικόνα 3.1.1.2.1 : (α) Εικόνα δείγμα πριν υποστεί κατάτμηση
(β) Το ιστόγραμμα της εικόνας (α)



(α)

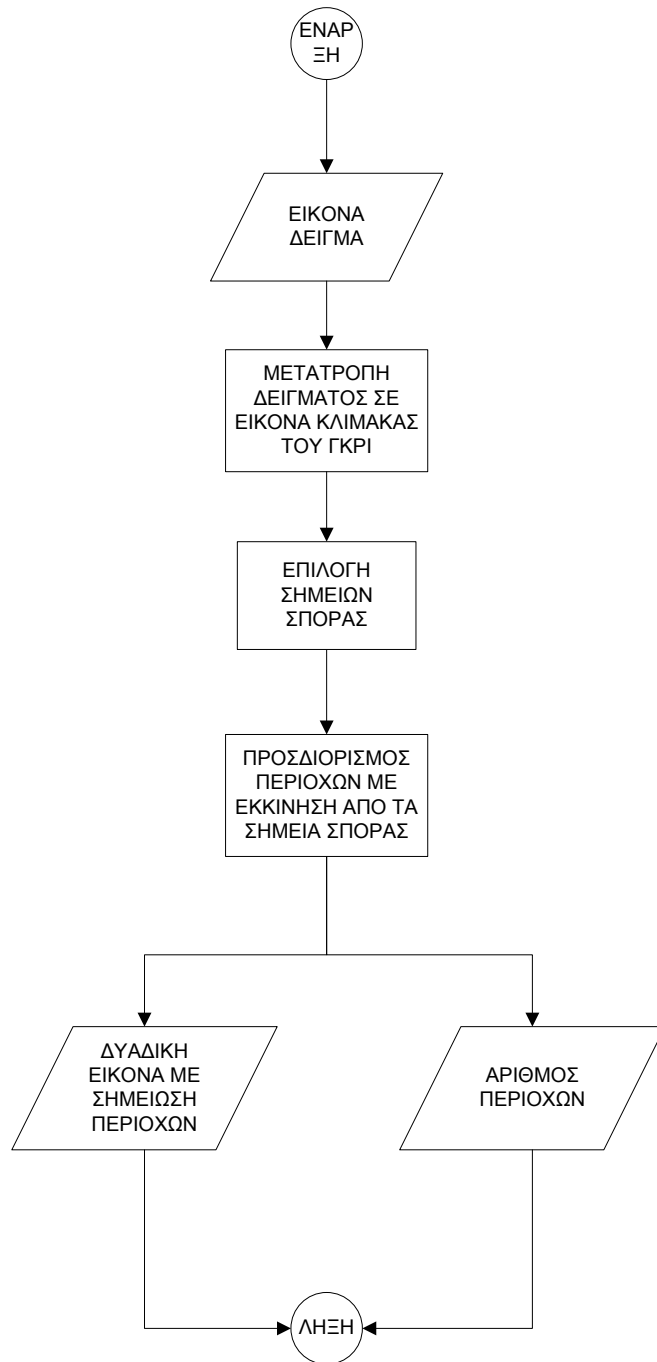


(β)

Εικόνα 3.1.1.2.2 : (α) Εικόνα που έχει υποστεί κατάτμηση με εύρος τιμών κατωφλίου έντασης 225-255

(β) Εικόνα που έχει υποστεί κατάτμηση με εύρος τιμών κατωφλίου έντασης 155-255

Στην υλοποίηση που ακολουθήθηκε στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας, το στάδιο κατάτμησης της εικόνας δέχεται ως είσοδο την έξοδο του σταδίου προ-επεξεργασίας, δηλαδή μία εικόνα με τιμές εικονοστοιχείων στην κλίμακα του γκρι. Κατόπιν, υπολογίζει το ιστόγραμμα της εν λόγω εικόνας, το οποίο χρησιμοποιείται για την εύρεση της ιδανικής τιμής κατωφλίου για την κλίμακα του γκρι, που θα αποτελέσει και το κριτήριο ένταξης ενός εικονοστοιχείου σε περιοχή. Στη συνέχεια, με βάση το δημιουργηθέν κριτήριο εντάσσει αναδρομικά όλα τα εικονοστοιχεία στις περιοχές που αντιστοιχούν. Τέλος, επιστρέφει ως έξοδο τόσο την τελική εικόνα με σημειωμένα τα σύνορα των περιοχών, όσο και τον αριθμό των περιοχών που προέκυψαν.



Σχήμα 3.1.1.1: Διάγραμμα ροής αλγόριθμου κατάτμησης εικόνας με τη μέθοδο Ανάπτυξης Περιοχών.

3.1.2. Αλγόριθμος Εύρεσης Ακμών Canny

Ο τελεστής ανίχνευσης ακμών του Canny αναπτύχθηκε από τον John F.Canny το 1986 και χρησιμοποιεί έναν αλγόριθμο πολλών σταδίων για να εντοπίσει ακμές σε εικόνες. Η μέθοδος αυτή επιτυγχάνει την ικανοποίηση των παρακάτω απαιτήσεων:

- Αποδοτική ανίχνευση: Ο αλγόριθμος εντοπίζει το μεγαλύτερο δυνατό αριθμό πραγματικών ακμών της εικόνας.
- Αποδοτικό εντοπισμό: Οι εντοπισμένες ακμές βρίσκονται όσο το δυνατό πιο κοντά στις πραγματικές ακμές της εικόνας.
- Ελάχιστη αντίδραση: Κάθε ακμή εντοπίζεται και σημειώνεται μόνο μία φορά και πιθανός θόρυβος της εικόνας στατιστικώς σπάνια δημιουργεί ψευδείς ακμές.

Η ικανοποίηση επετεύχθη με χρήση λογισμού διακυμάνσεων (calculus of variations). Η βέλτιστη συνάρτηση, την οποία υπολόγισε ο Canny με σκοπό τη βελτιστοποίηση των απαιτήσεων, μπορεί να περιγραφεί ως το άθροισμα τεσσάρων εκθετικών όρων, αλλά παριστάνεται ικανοποιητικά από την πρώτη παράγωγο μιας Γκαουσιανής.

Τα στάδια από τα οποία αποτελείται ο αλγόριθμος που υλοποιεί τη μέθοδο ανίχνευσης ακμών του Canny στη συγκεκριμένη εργασία είναι τέσσερα, πλήρως διακριτά μεταξύ τους, ενώ ως είσοδο δέχεται εικόνες κωδικοποιημένες σε οποιαδήποτε μορφή και χρωματικό μοντέλο. Το πρώτο στάδιο είναι η μείωση του θορύβου που μπορεί να περιέχει η εικόνα. Στη συνέχεια, υπολογίζεται η βαθμίδα κλίσης που αντιστοιχεί σε κάθε εικονοστοιχείο της. Το τρίτο στάδιο είναι αυτό της καταστολής των μη-μεγίστων (non maximum suppression). Το τέταρτο και τελευταίο στάδιο είναι αυτό της κατωφλίωσης με υστέρηση (thresholding with hysteresis). Τα στάδια αυτά αναλύονται με ακρίβεια στις υποενότητες που ακολουθούν.

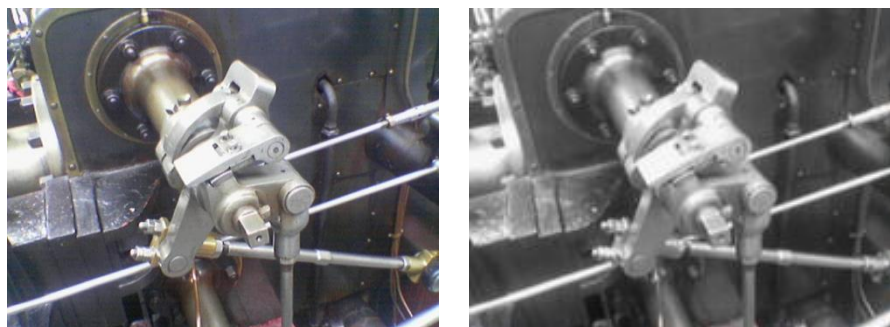
3.1.2.1 Στάδιο Μείωσης Θορύβου

Για τη μείωση του θορύβου της εικόνας που δέχεται ως είσοδο η υλοποίηση του αλγόριθμου ανίχνευσης ακμών του Canny χρησιμοποιήθηκε φίλτρο βασισμένο στην πρώτη παράγωγο μιας Γκαουσιανής. Το στάδιο αυτό είναι απαραίτητο, μιας και η

μέθοδος Canny είναι ιδιαίτερα ευαίσθητη στον θόρυβο που βρίσκεται στα ανεπεξέργαστα δεδομένα της εικόνας που δέχεται στην είσοδο. Επομένως, η ανεπεξέργαστη εικόνα αρχικά συνελίσσεται με ένα Γκαουσιανό φίλτρο. Το αποτέλεσμα μετά τη συνέλιξη είναι μία θολωμένη εκδοχή της αρχικής, η οποία όμως δεν επηρεάζεται από μεμονωμένα εικονοστοιχεία θορύβου σε κανένα επίπεδο. Ακολουθεί ένα παράδειγμα Γκαουσιανού φίλτρου μεγέθους 5x5.

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Σε περίπτωση συνέλιξης του εν λόγω φίλτρου με μία εικόνα προκύπτει αποτέλεσμα, όπως στην εικόνα 3.1.2.1.1:



(α)

(β)

Εικόνα 3.1.2.1.1 : (α) Η αρχική εικόνα
(β) Παράδειγμα εικόνας κλίμακας του γκρι μετά από συνέλιξη με το Γκαουσιανό φίλτρο του παραδείγματος.

Λόγω της ευαισθησίας της μεθόδου του Canny υλοποιήθηκε μία περαιτέρω μέθοδος, εκτός της χρήσης του Γκαουσιανού φίλτρου. Η μέθοδος αυτή στηρίχτηκε στην τεχνική κανονικοποίησης της αντίθεσης με χρήση της φωτεινότητας του κάθε εικονοστοιχείου. Έτσι γίνεται με ευχέρεια η επεξεργασία της εικόνας και η εύρεση των ακμών της.

Η φωτεινότητα είναι το φωτομετρικό μέτρο της έντασης του φωτός ανά μονάδα εμβαδού φωτός που ταξιδεύει σε μία συγκεκριμένη διεύθυνση. Περιγράφει την ποσότητα φωτός που περνά ή εκπέμπεται από μία συγκεκριμένη περιοχή και πέφτει σε μία δοσμένη γωνία. Η μονάδα της φωτεινότητας στο διεθνές σύστημα μονάδων (SI) είναι καντέλα ανά τετραγωνικό μέτρο (cd/m^2).

Στην ψηφιακή επεξεργασία εικόνας, κανονικοποίηση είναι η γραμμική διαδικασία κατά την οποία το εύρος τιμών της έντασης εικονοστοιχείων αλλάζει. Η κανονικοποίηση καλείται αλλιώς και επέκταση αντίθεσης (contrast stretching) ή δυναμική επέκταση εύρους (dynamic range expansion). Σκοπός της είναι η μετατροπή της εικόνας σε μορφή πιο οικίας (κανονικής) στις ανθρώπινες αισθήσεις. Η μετατροπή επιτυγχάνεται με ανακατανομή των εικονοστοιχείων σε όλο το εύρος δυνατών τιμών, ανάλογα πάντα με την αρχική τιμή τους. Αυτό πρακτικά σημαίνει ότι στην περίπτωση κανονικοποίησης της αντίθεσης δημιουργείται καλύτερη διακριτοποίηση των τιμών κάθε εικονοστοιχείου, που οδηγεί σε αύξηση της αντίθεσης της εικόνας συνολικά.



(α)



(β)

Εικόνα 3.1.2.1.2 : (α) Η αρχική εικόνα
(β) Παράδειγμα εικόνας μετά από κανονικοποίηση αντίθεσης.

3.1.2.2 Στάδιο Υπολογισμού Βαθμίδας Κλίσης

Στο στάδιο αυτό γίνεται υπολογισμός της τιμής της βαθμίδας κλίσης κάθε εικονοστοιχείου της εικόνας.

Μαθηματικώς, το διάνυσμα κλίσης (gradient vector) μιας συνάρτησης $f \in \mathbb{R}^n$ ορίζεται ως:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right).$$

Πρόκειται επομένως για έναν τελεστή, ο οποίος εφαρμόζεται πάνω στη συνάρτηση f .

Κάθε μία από τις n συνιστώσες του διανύσματος αυτού εκφράζει την ταχύτητα με την οποία μεταβάλλεται η συνάρτηση f κατά τη διεύθυνση n . Το διάνυσμα κλίσης χαρακτηρίζεται από το μέτρο και την κατεύθυνσή του.

- Η κατεύθυνση καταδειχνει τη διεύθυνση κατά την οποία η συνάρτηση f εμφανίζει την πιο απότομη μεταβολή της. Στην περίπτωση συνάρτησης δύο μεταβλητών ο τύπος υπολογισμού του είναι:

$$\theta = \arctan \left(\frac{G_x}{G_y} \right)$$

, όπου G_x και G_y οι μερικές παράγωγοι πρώτης τάξης ως προς τις δύο μεταβλητές x και y αντίστοιχα.

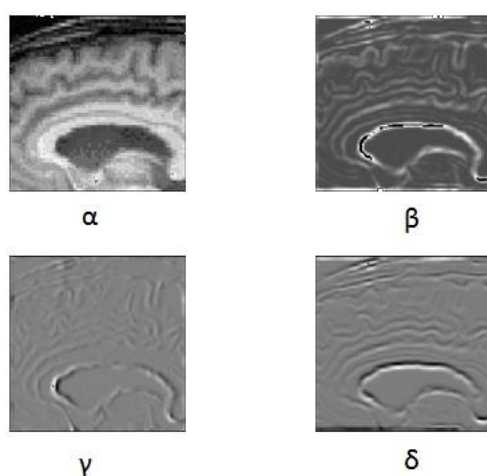
- Το μέτρο της, γνωστό και ως βαθμίδα κλίσης (gradient magnitude), υπολογίζεται από τον τύπο:

$$|\nabla f| = \sqrt{\sum_{i=0}^n \left(\frac{\partial f}{\partial x_i} \right)^2}$$

και υποδεικνύει το πόσο απότομη είναι η μεταβολή στην κατεύθυνση του διανύσματος κλίσης. Για εφαρμογή των παραπάνω στον τομέα επεξεργασίας εικόνας αρκεί η αντιστοιχία μιας εικόνας σε συνάρτηση. Κάτι τέτοιο επιτυγχάνεται εύκολα μέσω της θεώρησης ότι μια φωτογραφία αντιστοιχεί σε συνάρτηση δύο μεταβλητών, όπως έχει ήδη αναφερθεί. Οι δύο αυτές μεταβλητές αποτελούν τις συντεταγμένες x

και y του κάθε εικονοστοιχείου, ενώ οι τιμές της συνάρτησης ταυτίζονται με την τιμή του αντίστοιχου εικονοστοιχείου.

Έπειτα από υπολογισμό της βαθμίδας κλίσης που αντιστοιχεί σε κάθε σημείο της εικόνας, δύναται ο υπολογισμός αυτών των εικονοστοιχείων στα οποία παρατηρείται απότομη αλλαγή έντασης σε σχέση με την γύρω περιοχή τους. Τούτο επιτυγχάνεται αναγνωρίζοντας τα σημεία όπου η βαθμίδα κλίσης αποκτά πολύ μεγάλες τιμές.



Εικόνα 3.1.2.2.1 : Παράδειγμα χρήσης τελεστή κλίσης σε μία φωτογραφία:

(α) Η φωτογραφία-δείγμα

(β) Η βαθμίδα κλίσης του δείγματος

(γ) Το μέτρο της οριζόντιας συνιστώσας της κλίσης του δείγματος

(δ) Το μέτρο της κάθετης συνιστώσας της κλίσης του δείγματος

Ο τελεστής Sobel χρησιμοποιείται για την εύρεση των μερικών παραγώγων στις δύο διευθύνσεις του καρτεσιανού επιπέδου, όπου αναπαρίσταται κάθε εικόνα. Για το σκοπό αυτό γίνεται χρήση των παρακάτω δύο μητρών, μεγέθους 3×3 , οι οποίες συνελίσσονται με την εικόνα. Συνεπώς, οι δύο μερικές παράγωγοι G_x και G_y προκύπτουν από την αρχική φωτογραφία A ως εξής:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A \text{ και } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A$$

Τελικά, στο στάδιο αυτό γίνεται χρήση του τελεστή Sobel με σκοπό την εύρεση της βαθμίδας και της κατεύθυνσης της κλίσης της έντασης κάθε εικονοστοιχείου, με βάση τους μαθηματικούς τύπους που αναφέρθηκαν σε αυτή την ενότητα. Η γωνία της κατεύθυνσης κατόπιν στρογγυλοποιείται σε μία από τέσσερις γωνίες, που αναπαράστουν την οριζόντια, κάθετη και δύο διαγώνιες διευθύνσεις (0, 44, 90 και 135 μοίρες).

3.1.2.3 Στάδιο Καταστολής των Μη-μεγίστων

Στο στάδιο αυτό διενεργείται αναζήτηση με σκοπό την εύρεση τοπικού μεγίστου της βαθμίδας κλίσης στην κατεύθυνση της κλίσης, αν αυτό υπάρχει. Τοπικό μέγιστο της βαθμίδα κλίσης σε κάποιο σημείο, σημαίνει σύμφωνα με τα προαναφερθέντα θεωρητικά στοιχεία ότι το εικονοστοιχείο στο σημείο αυτό είναι σημείο ακμής. Η αναζήτηση επιτυγχάνεται μέσω των παρακάτω υποθέσεων:

- Αν η στρογγυλοποιημένη γωνία είναι 0 μοίρες, τότε το σημείο θα θεωρηθεί ακμή αν η τιμή της έντασής του είναι μεγαλύτερη από τις αντίστοιχες τιμές των εικονοστοιχείων που βρίσκονται στις δυτικές και ανατολικές κατευθύνσεις.
- Αν η στρογγυλοποιημένη γωνία είναι 90 μοίρες, τότε το σημείο θα θεωρηθεί ακμή αν η τιμή της έντασής του είναι μεγαλύτερη από τις αντίστοιχες τιμές των εικονοστοιχείων που βρίσκονται στις βόρειες και νότιες κατευθύνσεις.
- Αν η στρογγυλοποιημένη γωνία είναι 45 μοίρες, τότε το σημείο θα θεωρηθεί ακμή αν η τιμή της έντασής του είναι μεγαλύτερη από τις αντίστοιχες τιμές των εικονοστοιχείων που βρίσκονται στις βορειοανατολικές και νοτιοδυτικές κατευθύνσεις.
- Αν η στρογγυλοποιημένη γωνία είναι 135 μοίρες, τότε το σημείο θα θεωρηθεί ακμή αν η τιμή της έντασής του είναι μεγαλύτερη από τις αντίστοιχες τιμές των εικονοστοιχείων που βρίσκονται στις βορειοδυτικές και νοτιοανατολικές κατευθύνσεις.

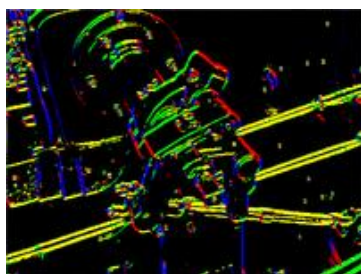
Από το στάδιο αυτό, προκύπτει ένα σύνολο ακμών σε μορφή δυαδικής εικόνας. Τα σύνολα αυτά καλούνται συχνά και λεπτές ακμές (thin edges).

3.1.2.4 Στάδιο Κατωφλίωσης με Υστέρηση

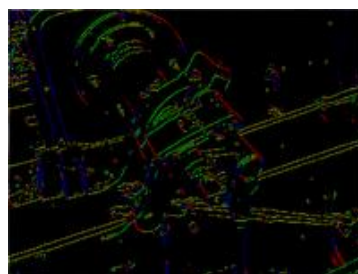
Το τελευταίο στάδιο του αλγόριθμου ανίχνευσης ακμών του Canny, έχει ενισχυτικό ρόλο στο αποτέλεσμα του τρίτου σταδίου. Παράλληλα επιχειρείται αποβολή από το σύνολο των ακμών όσων από αυτές αποδεικνύονται ψευδείς. Ως κριτήριο αποβολής επιλέγεται η κλίση της έντασης κάθε εικονοστοιχείου που αντιστοιχεί σε ακμή. Όσο μεγαλύτερη τιμή έχει η κλίση, τόσο πιο πιθανό είναι να αποτελεί πραγματική ακμή. Αυτός είναι και ο λόγος που στο τελευταίο στάδιο επιχειρείται κατωφλίωση με υστέρηση.

Η συγκεκριμένη μέθοδος απαιτεί δύο τιμές κατωφλίου, μία υψηλή και μία χαμηλή. Μέσω της υπόθεσης ότι οι πραγματικές ακμές πρέπει να είναι πάνω σε συνεχείς καμπύλες της εικόνας, ακολουθείται μία δοσμένη γραμμή και απορρίπτονται όσα εικονοστοιχεία έχουν υψηλές τιμές βαθμίδας κλίσης αλλά δεν αποτελούν μέρος της γραμμής αυτής. Αρχικά, εφαρμόζεται ένα κατώφλι υψηλής τιμής. Με τον τρόπο αυτό επιλέγονται και ξεχωρίζονται τα σημεία που είναι στατιστικώς βέβαιο πως αποτελούν ακμές. Ξεκινώντας στη συνέχεια από τις ακμές αυτές, εφαρμόζεται ένα χαμηλό κατώφλι, το οποίο επιτρέπει τον εντοπισμό ισχνών και μικρών σε πλάτος τμημάτων ακμών, με την προϋπόθεση να έχει βρεθεί ένα σημείο εκκίνησης.

Με την ολοκλήρωση της συγκεκριμένης διαδικασίας, προκύπτει μία δυαδική εικόνα όπου κάθε εικονοστοιχείο της είναι επιλεγμένο είτε ως ακμή, είτε ως μη ακμή. Η εικόνα αυτή είναι και η έξοδος του αλγόριθμου ανίχνευσης ακμών του Canny.



(α)

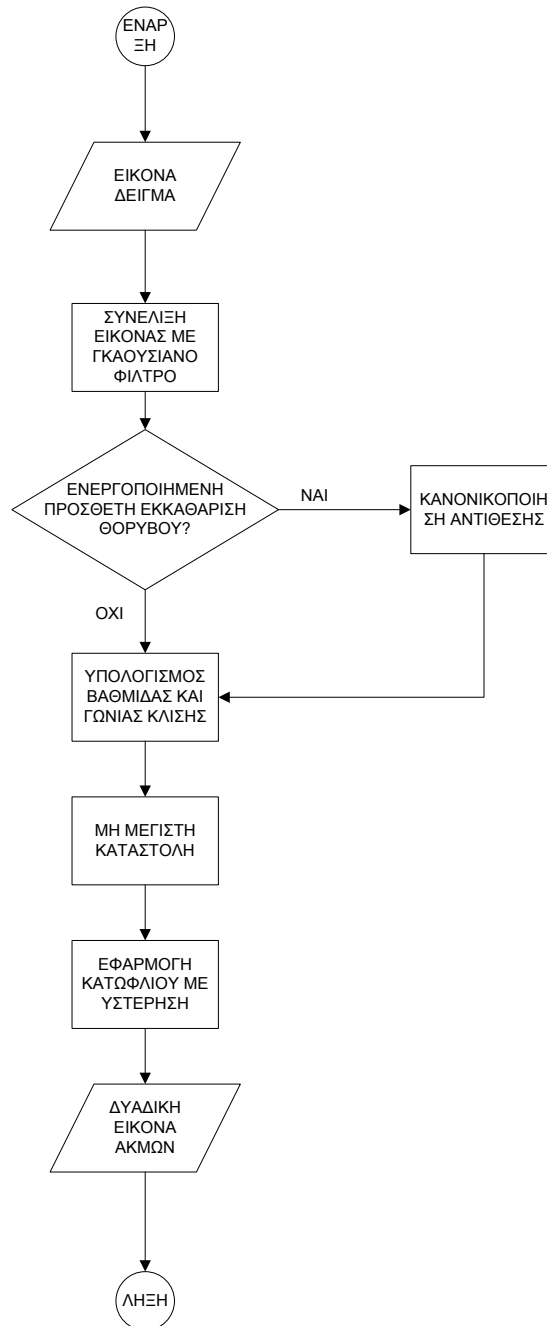


(β)

Εικόνα 3.1.2.4.1 : (α) Παράδειγμα χρήσης τελεστή Sobel στην εικόνα 3.1.2.1.1. Οι

ακμές έχουν χρώματα ανάλογα με την κατεύθυνσή τους. Κίτρινο για 0 μοίρες, πράσινο για 45, μπλε για 90 και κόκκινο για 135 μοίρες.

(β) Η τελική έξοδος του αλγόριθμου ανίχνευσης ακμών Canny.



Σχήμα 3.1.2.1: Διάγραμμα ροής αλγόριθμου εύρεσης ακμών εικόνας του Canny.

3.1.3. Τελικός Αλγόριθμος Εύρεσης Επιπλωμένου Χώρου από Εικόνα

Στις δύο προηγούμενες ενότητες αναλύθηκαν ενδελεχώς ο αλγόριθμος κατάτμησης εικόνας με χρήση της μεθόδου ανάπτυξης περιοχών και ο αλγόριθμος ανίχνευσης ακμών του Canny. Σκόπος της ανάλυσης αυτής ήταν η πληρέστερη κατανόηση της μορφής και των σταδίων του αλγόριθμου εύρεσης επιπλωμένων χώρων από εικόνα.

Κύριος μοχλός της υλοποίησης του εν λόγω αλγόριθμου που πραγματοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας ήταν η ακόλουθη παρατήρηση:

Ο αριθμός αντικειμένων σε μία εικόνα, και δει επίπλων, είναι συνήθως ανάλογος του ποσοστού των εικονοστοιχείων της που αποτελούν ακμές και του αριθμού των περιοχών στις οποίες κατακερματίζεται.

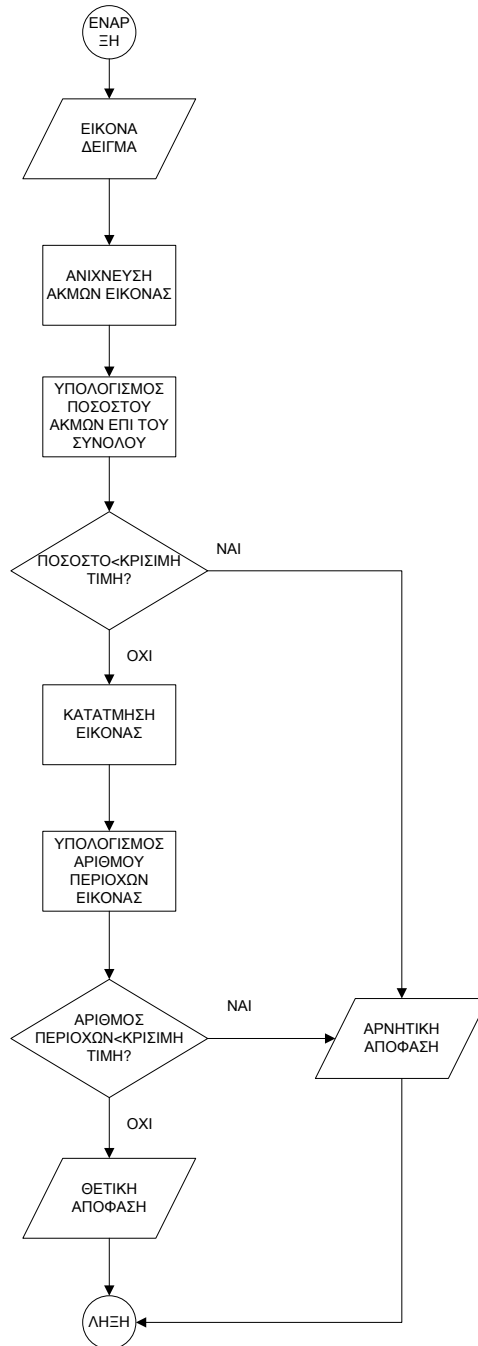
Η παραπάνω πρόταση αποδεικνύεται εύκολα με βάση τους ορισμούς της ακμής και της περιοχής, όπως αυτοί διατυπώθηκαν προηγουμένως.

Συνεπώς, η αλγόριθμος για την εύρεση επιπλωμένων χώρων λαμβάνει ως είσοδο μία εικόνα οποιασδήποτε μορφής κωδικοποίησης και χρωματικού μοντέλου RGB και την επεξεργάζεται στα εξής διακριτά στάδια. Στο πρώτο στάδιο εφαρμόζεται ο αλγόριθμος ανίχνευσης ακμών του Canny στην είσοδο και υπολογίζεται το ποσοστό των εικονοστοιχείων που αποτελούν ακμές επί του συνολικού αριθμού εικονοστοιχείων της εικόνας. Αν το ποσοστό αυτό δεν είναι μικρότερο μιας κρίσιμης τιμής, ο αλγόριθμος συνεχίζει. Διαφορετικά επιστρέφει αρνητική απάντηση.

Σε περίπτωση που η πορεία εκτέλεσης της μεθόδου δε διακοπεί, εφαρμόζεται ο αλγόριθμος κατάτμησης με ανάπτυξη περιοχών στην αρχική εικόνα που δέχτηκε ως είσοδο, όπου και λαμβάνεται ο αριθμός των συνολικών περιοχών στις οποίες μπορεί να διαχωριστεί αυτή. Ακολουθεί σύγκριση του αριθμού αυτού με μία δεύτερη κρίσιμη τιμή, από το αποτέλεσμα της οποίας ο αλγόριθμος λαμβάνει την τελική απόφαση, που αποτελεί και την έξοδό του.

Άξιο αναφοράς είναι το γεγονός της δυνατότητας μη εκτέλεσης του δεύτερου σκέλους του αλγόριθμου αν τα αποτελέσματα από την εύρεση ακμών δεν είναι

αρεστά. Εδώ η υλοποίηση έγινε με τέτοιο τρόπο, ούτως ώστε να εξασφαλίζεται η εγκυρότητα κι αποτελεσματικότητα του αλγόριθμου, ενώ ταυτόχρονα διασφαλίζεται το μέγιστο δυνατό κέρδος σε ταχύτητα εκτέλεσης.

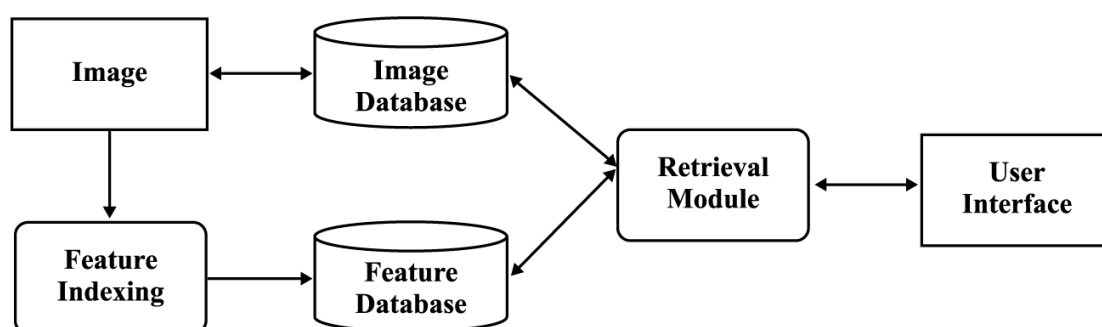


Σχήμα 3.1.3.1: Διάγραμμα ροής αλγόριθμου εύρεσης επιπλωμένων χώρων.

3.2. Σύγκριση Ομοιότητας Εικόνων

Ένα από τα βασικότερα χαρακτηριστικά της πλατφόρμας που υλοποιήθηκε αποτελεί η δυνατότητα σύγκρισης εικόνων που βρίσκονται αποθηκευμένες στη βάση δεδομένων με κάποια εικόνα επιλογής του χρήστη. Πρόκειται για ένα υβριδικό σύστημα ανάκτησης φωτογραφιών με βάση το περιεχόμενό τους (Content Based Image Retrieval system – CBIR), το οποίο υποβοηθάται σε θέματα απόδοσης κάνοντας ταυτόχρονη χρήση κάποιων μεταδεδομένων για μείωση του αριθμού ελέγχων.

Τα συστήματα CBIR αναλύουν το πραγματικό περιεχόμενο της εικόνας και όχι μεταδεδομένα όπως λέξεις-κλειδιά, ετικέτες, ή περιγραφές που συνδέονται με την εικόνα. Στο πλαίσιο «περιεχόμενο» ανήκουν χρώματα, σχήματα, υφές, ή οποιαδήποτε άλλη πληροφορία προερχόμενη από την ίδια την εικόνα. Η δυνατότητα CBIR αποτελεί επιθυμητή προσθήκη σε όλα τα συστήματα βάσεων φωτογραφιών, διότι οι αναζητήσεις εικόνων, η οποία βασίζεται αποκλειστικά σε μεταδεδομένα οδηγεί στην παραγωγή πολλών μη-έγκυρων αποτελεσμάτων. Τούτο είναι καθ'όλα λογικό, αν ληφθεί υπ'όψιν η δυσκολία περιγραφής μιας οποιασδήποτε εικόνας με βάση λέξεις-κλειδιά, ιδιαίτερα σε μια μεγάλη βάση δεδομένων. Τούτέστιν, σε ένα σύστημα που φιλτράρει τις εικόνες με βάση το περιεχόμενό τους παρατηρείται αύξηση απόδοσης έναντι ενός βασιζόμενου αυστηρά σε μεταδεδομένα.



Σχήμα 3.2.1 : Δομή ενός τυπικού CBIR συστήματος

Ιδανικά ένα σύστημα ανάκτησης εικόνας με βάση το περιεχόμενό τους θα χρησιμοποιούσε σημασιολογική ανάλυση (semantic analysis-retrieval). Δηλαδή ο

χρήστης θα καθόριζε λεκτικά το χώρο αναζήτησης και το σύστημα θα αποφαίνονταν για τις ιδιότητες που θα έπρεπε μία φωτογραφία να κατέχει ώστε να γίνει αποδεκτή ως αποτέλεσμα της αναζήτησης. Κάτι τέτοιο όμως δεν είναι δυνατό με βάση τα σημερινά δεδομένα και υπολογιστικές τεχνικές.

Έτσι, κυριότερος τρόπος υποβολής ερωτήματος (query) σε ένα CBIR σύστημα είναι με χρήση παραδειγμάτων (query by example). Κατά αυτό τον τρόπο, ο χρήστης εφοδιάζει το σύστημα με μία εικόνα-παραδειγμα. Στη συνέχεια το σύστημα με συνεχείς συγκρίσεις με κάθε φωτογραφία στη βάση επιλέγει και παρουσιάζει αυτές που πληρούν κάποια αυστηρά ορισμένα κριτήρια ομοιότητας.

Η πιο συνηθισμένη μέθοδος σύγκρισης δύο φωτογραφιών σε ένα CBIR σύστημα είναι με χρήση κάποιας μετρικής απόστασης, όπως η μετρική L1. Προσμετράται συνεπώς η ομοιότητα δύο εικόνων σε ποικίλες διαστάσεις, όπως του χρώματος, της υφής και του σχήματος. Για το σκοπό αυτό έχουν αναπτυχθεί πλήθος μεθόδων και για την υλοποίηση αυτής της εργασίας επιλέχτηκε η μέθοδος των συζευγμένων ιστογραμμάτων εικόνων [Greg Pass, Ramin Zabih, 1999], όπως αυτή παρουσιάστηκε στο προηγούμενο κεφάλαιο.

Ο αλγόριθμος που υλοποιεί την παραπάνω μέθοδο απαρτίζεται από τρία διακριτά στάδια, την προ-επεξεργασία της εικόνας εισόδου, την εξαγωγή των συζευγμένων ιστογραμμάτων της εικόνας-παραδειγμα καθώς και των φωτογραφιών της βάσης, τον υπολογισμό της μετρικής L1 μεταξύ των συνιστωσών του ιστογράμματος του παραδείγματος και των αντίστοιχων συνιστωσών κάθε εικόνας. Το τελευταίο αυτό στάδιο είναι υπεύθυνο για την εξαγωγή των συμπερασμάτων με βάση την τιμή της υπολογισμένης μετρικής L1.

Ως είσοδό του δέχεται την εικόνα-παραδειγμα καθώς και κάθε μία από τις εικόνες προς σύγκριση της βάσης. Αρχικά, το στάδιο προ - επεξεργασίας εφαρμόζει τεχνική κβαντοποίησης των χρωμάτων στις εικόνες αυτές. Η κβαντοποίηση μειώνει κι ομαδοποιεί τα χρώματα κάθε φωτογραφίας σε 16 ομάδες-χρώματα. Με αυτό τον τρόπο ελαχιστοποιούνται όσο το δυνατόν οι διαφοροποιήσεις των εικόνων λόγω των αποκλίσεων που προκύπτουν από διαφορετικούς τόνους των ίδιων χρωμάτων. Αυξάνεται επομένως η πιθανότητα επιτυχημένης σύγκρισης.

Το επόμενο στάδιο είναι, με βάση τον αλγόριθμο σύγκρισης όμοιων εικόνων, η εξαγωγή των συζευγμένων ιστογραμμάτων όλων των εικόνων προς δειγματοληψία.

Το στάδιο αυτό διαχωρίζεται σε δύο επιμέρους υπο-στάδια. Στο πρώτο γίνεται λήψη των χρωματικών χαρακτηριστικών του ιστογράμματος ενώ στο δεύτερο υπολογίζεται η συνιστώσα της βαθμίδα κλίσης.

Τέλος, το στάδιο υπολογισμού της μετρικής L1 κι επεξεργασίας του προκύπτοντος αποτελέσματος έχει ως σκοπό την εξαγωγή ενός, όσο πιο ασφαλούς γίνεται, συμπεράσματος για το αν υφίσταται τελικά ομοιότητα μεταξύ των συγκρινόμενων εικόνων ή όχι.

Οι υποενότητες του κεφαλαίου που ακολουθούν αναφέρονται αναλυτικότερα στα επιμέρους στάδια του αλγορίθμου, όπως αυτά απαριθμήθηκαν παραπάνω.

3.2.1. Στάδιο Προ-επεξεργασίας Εικόνων προς Σύγκριση

Το στάδιο αυτό δέχεται στην είσοδό του δύο φωτογραφίες. Μία εικόνα-παράδειγμα και μία εικόνα προς σύγκριση με την πρώτη από τη βάση, στις οποίες απεικονίζεται είτε εσωτερικός χώρος οικίας, είτε εξωτερικός. Στο κεφάλαιο 4 (ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ) αναλύεται ο τρόπος επιλογής τους. Οι είσοδοι αυτές οφείλουν να πληρούν τις ίδιες προϋποθέσεις με την είσοδο του αλγορίθμου που περιγράφηκε στην ενότητα 3.1. Συνεπώς, μπορούν να είναι κωδικοποιημένες με διάφορους τρόπους (format), όπως jpeg, bitmap, gif ή png αλλά απαιτείται η χρήση του χρωματικού μοντέλου RGB.

Ο λόγος ύπαρξης του σταδίου προεπεξεργασίας είναι η μετατροπή των χρωματικών χαρακτηριστικών των εικόνων που δέχεται το υποσύστημα στην είσοδό του σε μορφή, η οποία θα ελαχιστοποιεί το ενδεχόμενο εξαγωγής λανθασμένου συμπεράσματος από τα επόμενα στάδια. Αυτό καθίσταται δυνατό με χρήση αλγορίθμου κβαντοποίησης χρωμάτων. Έτσι, επιτυγχάνεται πλήρης ομαλοποίηση των χρωματικών διαφοροποιήσεων που υφίστανται σε μία εικόνα, δημιουργώντας ουσιαστικά περιοχές όμοιου χρωματικού τόνου στην εκάστοτε φωτογραφία. Οι τιμές που μπορεί να λάβει κάθε μία εξ'αυτών των περιοχών είναι μία από τις 16 προκαθορισμένες από τον αλγόριθμο.

Ο αλγόριθμος κβαντοποίησης χρώματος του σταδίου προεπεξεργασίας εκτελεί την παρακάτω λειτουργία:

Μετασηματίζει την εικόνα εισόδου σε νέα, με τιμές στα εικονοστοιχεία της να αντιστοιχούν σε αποχρώσεις μία εκ των 16 προκαθορισμένων από τον αλγόριθμο, αν δεν είναι ήδη τέτοιες. Το σύστημα στα επόμενα στάδια επεξεργάζεται μόνο τις μετασηματισμένες από τον αλγόριθμο εικόνες. Παρ' όλα αυτά, η αρχική ανεπεξέργαστη εικόνα δε διαγράφεται. Κι αυτό διότι η αρχική φωτογραφία χρησιμοποιείται σε άλλα υποσυστήματα για εξαγωγή διαφορετικών πληροφοριών, καθώς και από το ίδιο τον αλγόριθμο σύγκρισης εικόνων. Αυτό συμβαίνει όταν η εικόνα επιστρέφεται ως αποτέλεσμα σε περίπτωση που στα μετέπειτα στάδια επεξεργασίας διαπιστωθεί πως πληροί τα κριτήρια ομοιότητας.

Άξιο αναφοράς είναι το γεγονός ότι ο αλγόριθμος σύγκρισης εικόνων παρουσιάζει αδυναμία λειτουργίας όταν οι φωτογραφίες προς σύγκριση είναι διαφορετικής ανάλυσης εικονοστοιχείων. Παρ' όλα αυτά, το στάδιο προ-επεξεργασίας δεν ασχολείται με τη μετατροπή της ανάλυσης της κβαντισμένης χρωματικά εικόνας σε μορφή αντιμετωπίσιμη από τις παραμέτρους του συστήματος. Κι αυτό διότι την ομοιομορφία στα χαρακτηριστικά των εικόνων που βρίσκονται αποθηκευμένες στη βάση αναλαμβάνουν να ελέγχουν και να εξασφαλίζουν τμήματα των υποσυστημάτων εξυπηρετητή και πελάτη, τα οποία αναλύονται ενδελεχώς στο κεφάλαιο 4.



Εικόνα 3.2.1.1 : Είσοδος κι έξοδος του αλγόριθμου κβαντισμού χρωμάτων

3.2.2. Στάδιο Εξαγωγής Συζευγμένων Ιστογραμμάτων

Σκοπός του συγκεκριμένου σταδίου αποτελεί η δημιουργία κι εξαγωγή των συζευγμένων ιστογραμμάτων των εικόνων.

Ένα ιστόγραμμα χρώματος ορίζεται γενικά ως η αναπαράσταση της κατανομής των χρωμάτων μιας εικόνας. Στο σύστημα σύγκρισης παρέχονται ως είσοδος ψηφιακές φωτογραφίες. Το ιστόγραμμα χρώματος ψηφιοποιημένης εικόνας αντιπροσωπεύει τον αριθμό των εικονοστοιχείων της, τα οποία έχουν τιμές είτε σε καθένα από τα πιθανά φάσματα χρώματος, σε περίπτωση ομαδοποίησης, είτε κάθε χρώματος του χρωματικού χώρου (color space) της εικόνας.

Ένα ιστόγραμμα εικόνας παράγεται διακριτοποιώντας αρχικά τα χρώματα της εικόνας σε έναν αριθμό από ομάδες, και μετρώντας τον αριθμό των εικονοστοιχείων, των οποίων οι τιμές ανήκουν σε κάθε μία εκ των ομάδων. Ο βαθμός διακριτοποίησης είναι κάθε φορά ανάλογος της εφαρμογής. Για παράδειγμα, στην περίπτωση του RGB μοντέλου ο μέγιστος αριθμών ομάδων είναι 256, όσες και οι πιθανές τιμές έντασης καθενός από τα bytes του εκάστοτε εικονοστοιχείου.

Ακολουθεί ένα παράδειγμα τρισδιάστατου ιστογράμματος RGB για τη φωτογραφία της εικόνας 3.2.2.1, όπου οι τιμές έντασης κάθε εικονοστοιχείου έχουν ομαδοποιηθεί σε 4 ομάδες:

- Ομάδα 0 : 0-63
- Ομάδα 1 : 64-127
- Ομάδα 2 : 128-191
- Ομάδα 3 : 192-255



Εικόνα 3.2.2.1 : Φωτογραφία RGB μοντέλου μιας γάτας

Στον πίνακα 3.2.2.1 παρουσιάζεται το ιστόγραμμα RGB χρωμάτων για την εικόνα 3.2.2.1 σε μορφή πίνακα.

Πίνακας 3.2.2.1 : Ιστόγραμμα RGB μοντέλου της εικόνας 3.2.2.1

Κόκκινο	Πράσινο	Μπλε	Αριθμός Εικονοστοιχείων
0	0	0	7414
0	0	1	230
0	0	2	0
0	0	3	0
0	1	0	8
0	1	1	372
0	1	2	88
0	1	3	0
0	2	0	0
0	2	1	0
0	2	2	10
0	2	3	1
0	3	0	0
0	3	1	0
0	3	2	0
0	3	3	0
1	0	0	891
1	0	1	13
1	0	2	0
1	0	3	0
1	1	0	592
1	1	1	3462
1	1	2	355
1	1	3	0
1	2	0	0
1	2	1	101
1	2	2	882
1	2	3	16
1	3	0	0
1	3	1	0
1	3	2	0
1	3	3	0
2	0	0	1146

2	0	1	0
2	0	2	0
2	0	3	0
2	1	0	2552
2	1	1	9040
2	1	2	47
2	1	3	0
2	2	0	0
2	2	1	8808
2	2	2	53110
2	2	3	11053
2	3	0	0
2	3	1	0
2	3	2	170
2	3	3	17533
3	0	0	11
3	0	1	0
3	0	2	0
3	0	3	0
3	1	0	856
3	1	1	1376
3	1	2	0
3	1	3	0
3	2	0	0
3	2	1	3650
3	2	2	6260
3	2	3	109
3	3	0	0
3	3	1	0
3	3	2	3415
3	3	3	53929

Ο ορισμός και το θεωρητικό υπόβαθρο σχετικά με τη βαθμίδα κλίσης παρουσιάστηκαν αναλυτικά στην ενότητα 3.1.2.1.

Η βασική ιδέα του αλγόριθμου είναι η δημιουργία ενός τετραδιάστατου ιστογράμματος. Κάθε μία διάσταση περιλαμβάνει τις ομάδες για τα τρία αντιπροσωπευτικά χρώματα, κόκκινο, πράσινο και μπλε του RGB μοντέλου. Η τέταρτη και τελευταία διάσταση περιλαμβάνει τις ομάδες για τη βαθμίδα κλίσης, η

οποία αποτελεί και την χωρική πληροφορία, απαραίτητη για την αποδοτικότητα του αλγορίθμου σύγκρισης.

Αρχικά, ο αλγόριθμος δέχεται ως είσοδο τις κβαντισμένες χρωματικά εικόνες που αποτέλεσαν την έξοδο του σταδίου προεπεξεργασίας. Στη συνέχεια, υπολογίζει τις τιμές για κάθε ένα εκ των συνιστάμενων βασικών χρωμάτων του RGB μοντέλου για όλα τα εικονοστοιχεία της εικόνας. Ο υπολογισμός ολοκληρώνεται με την απαρίθμηση των εικονοστοιχείων που αντιστοιχούν σε κάθε δυνατό συνδιασμό των RGB τιμών, όπως στο παραπάνω παράδειγμα. Αξίζει να σημειωθεί ότι για την επίτευξη της μέγιστης δυνατής διαφοροποίησης, με σκοπό την αποφυγή λανθασμένων αποτελεσμάτων στην μετέπειτα μετρική σύγκριση των εικόνων δεν υπήρξε ομαδοποίηση των τιμών. Στη συνέχεια υπολογίζεται η βαθμίδα κλίσης κάθε εικονοστοιχείου με τη μέθοδο που περιγράφηκε προηγουμένως.

3.2.3. Στάδιο Υπολογισμού μετρικής L1 και Εξαγωγής Συμπερασμάτων

Το τρίτο και τελευταίο στάδιο του αλγορίθμου αποσκοπεί στον υπολογισμό της μετρικής L1, ούτως ώστε να αποφανθεί τελικά για το αποτέλεσμα της σύγκρισης.

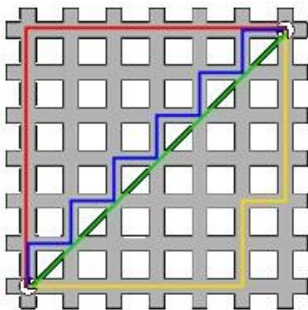
Στην Taxicab γεωμετρία, η οποία παρουσιάστηκε από τον Hermann Minkowski τον 19^ο αιώνα, τη συνήθη Ευκλείδεια συνάρτηση απόστασης αντικαθιστά μία νέα μετρική. Ως εκ τούτου, η απόσταση ανάμεσα σε δύο σημεία ορίζεται ως το άθροισμα της απόλυτης διαφοράς των συντεταγμένων τους. Το αποτέλεσμα αυτό αναφέρεται ως μετρική L1 ή απόσταση Manhattan.

Πιο συγκεκριμένα, η απόσταση d_1 δύο διανυσμάτων \mathbf{p} , \mathbf{q} σε ένα χώρο n διαστάσεων με αυστηρά καθορισμένο καρτεσιανό σύστημα συντεταγμένων είναι το άθροισμα του μήκους των προβολών, πάνω στους άξονες των συντεταγμένων, του ευθύγραμμου τμήματος ανάμεσα στις αντίστοιχες συνιστώσες των \mathbf{p} και \mathbf{q} .

Δηλαδή:

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|,$$

, όπου $\mathbf{P} = (p_1, p_2, \dots, p_n)$ και $\mathbf{Q} = (q_1, q_2, \dots, q_n)$.



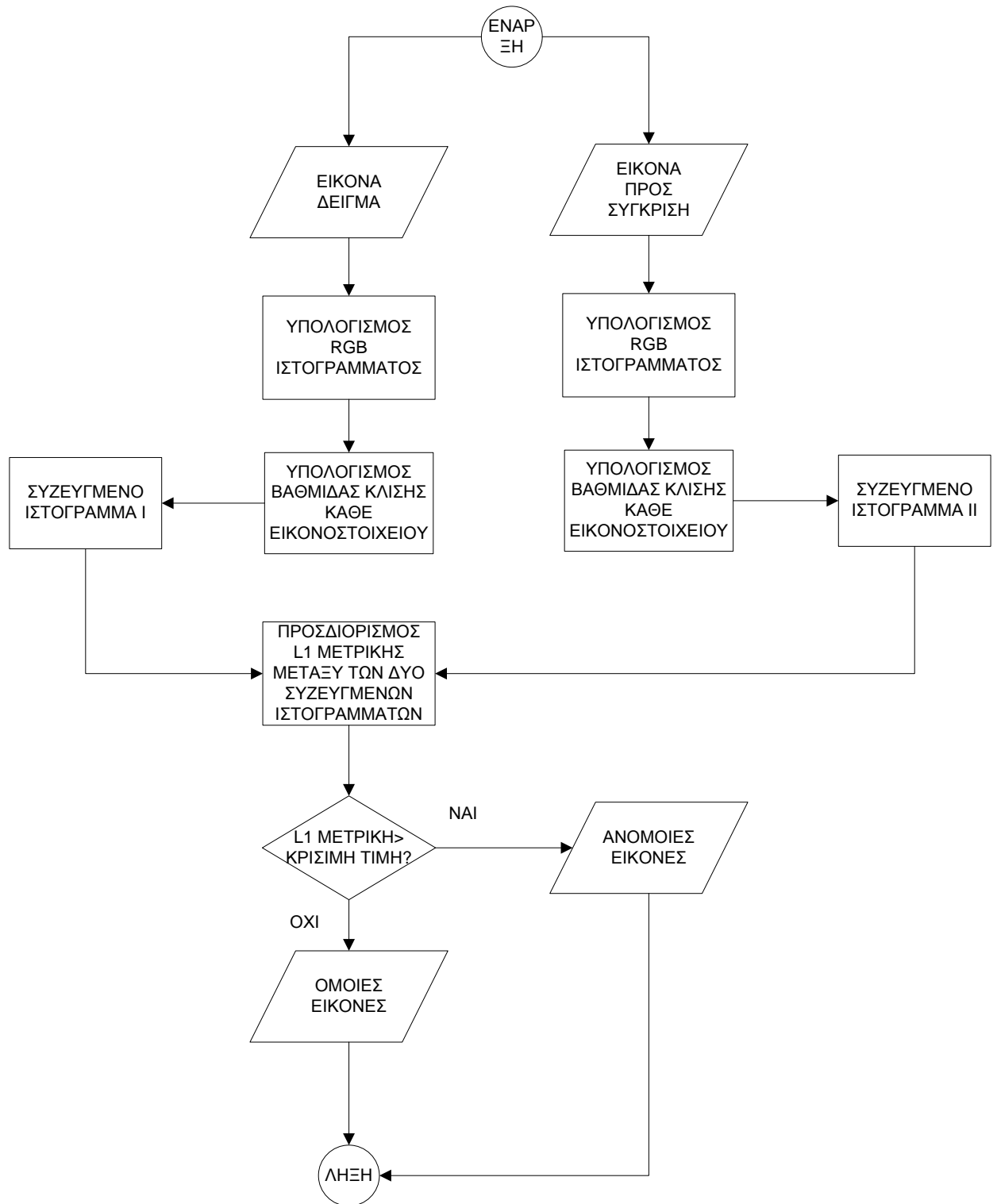
Εικόνα 3.2.3.1 : Στην εικόνα αυτή τα μήκη όλων των γραμμών με βάση την Taxicab γεωμετρία έχουν ίδια τιμή, σε αντίθεση με την Ευκλείδεια, όπου το μικρότερο μήκος έχει η πράσινη.

Το στάδιο αυτό δέχεται ως είσοδο τα συζευγμένα ιστογράμματα τόσο της εικόνας-δείγμα, όσο και των υπόλοιπων προς σύγκριση-εικόνων της βάσης. Κατόπιν, για καθένα από τα εικονοστοιχεία της εκάστοτε φωτογραφίας υπολογίζεται η μετρική L1 της σε σχέση με το αντίστοιχο εικονοστοιχείο του δείγματος.

Κατά αυτό τον τρόπο, αποθηκεύονται τέσσερις τιμές για κάθε εικονοστοιχείο, τρεις από τις οποίες αντιστοιχούν στις μετρικές L1 των βασικών χρωμάτων του RGB μοντέλου, ενώ η τέταρτη στην τιμή L1 της βαθμίδας κλίσης.

Στη συνέχεια, οι τιμές αυτές υφίστανται επεξεργασία ούτως ώστε να αναχθούν σε κλίμακα από 0 έως 100. Σκοπός της συγκεκριμένης επεξεργασίας είναι να υπάρξει μεγαλύτερο εύρος διαφορών, σε μεγέθη τα οποία τις καθιστούν εύκολα αντιληπτές.

Ακολουθεί το διάγραμμα ροής του προαναφερθέντος αλγόριθμου.



Σχήμα 3.2.1: Διάγραμμα ροής αλγόριθμου σύγκρισης εικόνων με τη μέθοδο συζευγμένων ιστογραμμάτων.

Έχοντας πλέον αναχθεί στην κλίμακα αυτή, εξάγονται για κάθε φωτογραφία που συγκρίνεται με την εικόνα-δείγμα τέσσερις μέσες τιμές, σε πλήρη αντιστοιχία των τιμών του κάθε εικονοστοιχείου. Όσο μεγαλύτερες τιμές ανατίθενται σε αυτές, τόσο μεγαλύτερη είναι και η υφιστάμενη διαφορά της εκάστοτε εικόνας με το αρχικό δείγμα.

Η τελική διαδικασία απόφασης του αποτελέσματος σύγκρισης της κάθε εικόνας είναι τυπική. Πρόκειται για μία σύγκριση του γενικού μέσου όρου των τεσσάρων τιμών κάθε φωτογραφίας με μία τιμή κατωφλίου. Η τιμή αυτή μπορεί να μεταβάλλεται ανάλογα με το επιθυμητό ποσοστό ομοιότητας που έχει τεθεί ως στόχος. Η έξοδος του τελευταίου σταδίου, και εν τέλει του αλγόριθμου είναι το αποτέλεσμα της προαναφερθείσας σύγκρισης.

3.2.4. Εφαρμογή Αλγόριθμου Σύγκρισης Εικόνων

Ο αλγόριθμος, του οποίου τα στάδια αναλύθηκαν στις ενότητες 3.2.1, 3.2.2 και 3.2.3, βρίσκει εφαρμογή στο **υποσύστημα συστάσεων** της πλατφόρμας που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας.

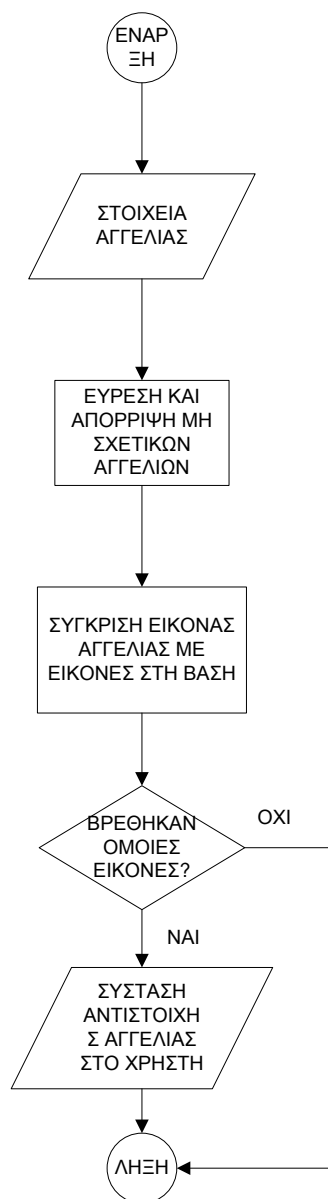
Το υποσύστημα αυτό, αναζητά κι εμφανίζει αγγελίες στο χρήστη αυτόματα. Οι αγγελίες αυτές θα πρέπει να είναι σχετικές με τις αναζητήσεις του εν λόγω χρήστη, ούτως ώστε να διευκολύνεται η αναζήτησή του κι επομένως να αυξάνεται η αποδοτικότητα της πλατφόρμας που υλοποιήθηκε.

Για την υλοποίηση του υποσυστήματος αυτού χρησιμοποιήθηκε αλγόριθμος συστάσεων. Ο αλγόριθμος αυτός, διακρίνεται σε 2 στάδια. Αρχικά, λαμβάνει ως είσοδό του τα στοιχεία της αγγελίας που ο χρήστης κοιτά εκείνη τη στιγμή. Τα δεδομένα αυτά θα του επιτρέψουν να απορρίψει ένα ποσοστό των αποθηκευμένων στη βάση αγγελιών ως μη σχετικές. Τα στοιχεία που επιλέγονται για αυτό το σκοπό αναλύονται εκτενώς στο κεφάλαιο 4.

Στο επόμενο στάδιο διαλέγει μία αντιπροσωπευτική εικόνα της επιλεγμένης από το χρήστη αγγελίας και τη συγκρίνει με όμοιες φωτογραφίες, όσων αγγελιών της βάσης

δεν έχουν απορριφτεί προηγουμένως. Για τη σύγκριση χρησιμοποιείται ο αλγόριθμος σύγκρισης εικόνων με βάση τα συζευγμένα ιστογράμματα. Με τον όρο όμοιες αναφέρονται οι εικόνες που τα μεταδεδομένα τους υποδεικνύουν πως απεικονίζουν ίδιους χώρους. Δηλαδή, μία εικόνα ενός καθιστικού θα συγκριθεί μόνο με φωτογραφίες καθιστικών και ου το καθ'εξής. Σε όσες από τις παραπάνω συγκρίσεις διαπιστωθεί ομοιότητα των εικόνων, η αγγελία στην οποία αντιστοιχούν συστήνεται στο χρήστη.

Ακολουθεί το διάγραμμα ροής του προαναφερθέντος αλγόριθμου.



Σχήμα 3.2.2: Διάγραμμα ροής αλγόριθμου υποσυστήματος συστάσεων.

3.3. Αλγόριθμος Εύρεσης Κυρίαρχου Χρώματος

Δύο ακόμη χαρακτηριστικά της πλατφόρμας που υλοποιήθηκε στα πλαίσια της διπλωματικής αυτής εργασίας είναι η δυνατότητα αναγνώρισης του τύπου πατώματος που απεικονίζει μία εικόνα καθώς και του χρώματος των τοίχων μιας οικίας, αν αυτή απεικονίζεται σε φωτογραφία. Η υλοποίηση αυτών βασίζεται στον αλγόριθμο εύρεσης κυρίαρχου χρώματος, που αναλύεται στην ενότητα αυτή.

Ο αλγόριθμος αυτός δέχεται ως είσοδο μία εικόνα οποιασδήποτε μορφής κωδικοποίησης, η οποία κάνει χρήση του χρωματικού μοντέλου RGB. Στη συνέχεια, διακρίνονται δύο στάδια του αλγορίθμου. Το πρώτο στάδιο **μετατρέπει** τις RGB τιμές του κάθε εικονοστοιχείου σε τιμές HSV του αντίστοιχου μοντέλου, ενώ το δεύτερο υπολογίζει το χρώμα των εικονοστοιχείων με **το μεγαλύτερο ποσοστό** εμφάνισης στην εν λόγω εικόνα, αφού έχει κατατάξει κάθε ένα από τα εμφανιζόμενα χρώματα σε ένα από τα προκαθορισμένα σύνολα χρωμάτων.

3.3.1. Μετατροπή RGB Τιμών σε HSV

Στο πρώτο στάδιο του αλγορίθμου εύρεσης κυρίαρχου χρώματος γίνεται επεξεργασία των τιμών RGB της εικόνας εισόδου με σκοπό τη μετατροπή τους σε τιμές του HSV. Ενώ τα χρωματικά μοντέλα HSV και RGB παρουσιάζονται αναλυτικά στο παράρτημα 6.1, για ευκολία κατανόησης τονίζονται κάποια ιδιαίτερα χαρακτηριστικά κι εφαρμογές τους και σε αυτή την ενότητα.

Πιο συγκεκριμένα, το RGB μοντέλο είναι ένα προσθετικό πρότυπο στο οποίο τα χρώματα κόκκινο, πράσινο και μπλε συνδυάζονται με διάφορους τρόπους για να αναπαραχθούν άλλα χρώματα. Το όνομα του προτύπου και η σύντηξη RGB προέρχονται από τα τρία βασικά χρώματα, το κόκκινο (**R**ed), πράσινο (**G**reen), και το μπλε (**B**lue). Αυτά τα τρία χρώματα δεν πρέπει να συγχέονται με τα τρία ανακλαστικά χρώματα κόκκινο, μπλε, και κίτρινο, τα οποία αναφέρονται στον χώρο των τεχνών ως βασικά χρώματα.

Το HSV, που αλλιώς αναφέρεται και ως HSB (απόχρωση - Hue, κορεσμός - Saturation, φωτεινότητα - Brightness), μετασχηματίζει μη γραμμικά το RGB μοντέλο με σκοπό να προσεγγιστεί καλύτερα η αίσθηση του χρώματος, όπως αυτή γίνεται αντιληπτή από την ανθρώπινη όραση. Οι τύποι μετασχηματισμού του μοντέλου RGB σε HSV είναι οι εξής:

$$H = 60^\circ \times H'$$

$$S = \begin{cases} 0 & , \text{αν } V = 0 \\ \frac{C}{V} & , \text{αλλιώς} \end{cases}$$

$$V = M$$

,όπου

$$H' = \begin{cases} \text{undefined,} & \text{if } C = 0 \\ \frac{G-B}{C} \bmod 6, & \text{if } M = R \\ \frac{B-R}{C} + 2, & \text{if } M = G \\ \frac{R-G}{C} + 4, & \text{if } M = B \end{cases}$$

Και

$$M = \max(R, G, B)$$

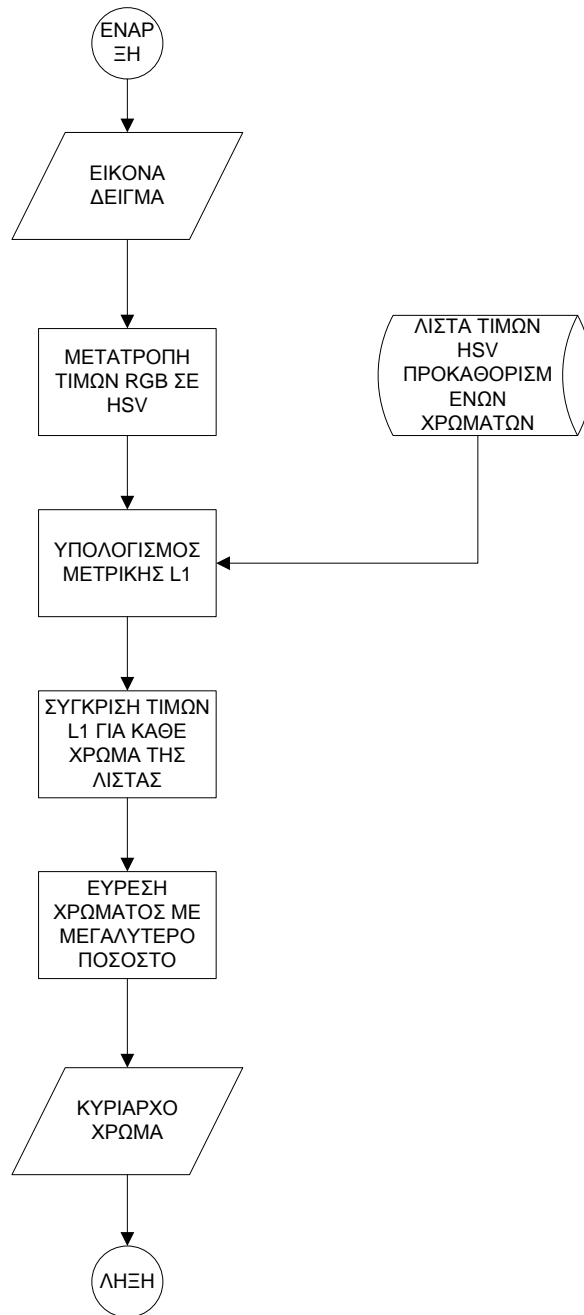
$$m = \min(R, G, B)$$

$$C = M - m$$

Η επιλογή της αναπαράστασης του μοντέλου RGB με τη μορφή HSV έγινε με γνώμονα το επόμενο στάδιο του αλγόριθμου, αυτό της σύγκρισης.

Κι αυτό διότι κλειδί της επιτυχημένης σύγκρισης χρωμάτων είναι η εύρεση ενός μαθηματικού μοντέλου που προσεγγίζει αποδοτικά το ανθρώπινο αισθητήριο. Την προσέγγιση αυτή επιτυγχάνει πολύ καλύτερα το μοντέλο HSV σε σύγκριση με το RGB.

Ακολουθεί το διάγραμμα ροής του προαναφερθέντος αλγόριθμου.



Σχήμα 3.3.1: Διάγραμμα ροής αλγόριθμου εύρεσης κυρίαρχου χρώματος.

3.3.2. Υπολογισμός Κυρίαρχου Χρώματος

Το δεύτερο και τελευταίο στάδιο του αλγόριθμου, εντοπίζει το χρώμα από δοθείσα λίστα, που προσεγγίζεται καλύτερα από το χρώμα κάθε εικονοστοιχείου της εικόνας.

Για την επίτευξη αυτού, εκτελούνται διαδοχικοί υπολογισμοί της μετρικής L1, μεταξύ κάθε μίας εκ των τιμών HSV του εκάστοτε εικονοστοιχείου και των αντίστοιχων τιμών HSV κάθε χρώματος της λίστας. Κατόπιν, υπολογίζεται ο μέσος όρος των L1 αποστάσεων που προέκυψαν για κάθε στοιχείο-χρώμα της λίστας. Τελικά, το στοιχείο της χρωματικής λίστας που προσεγγίζεται καλύτερα από τις τιμές HSV του εικονοστοιχείου επιλέγεται εκείνο του οποίου ο μέσος όρος των L1 αποστάσεων του από τις HSV τιμές του εικονοστοιχείου είναι ο μικρότερος.

Έχοντας υπολογιστεί το χρώμα προσέγγισης κάθε εικονοστοιχείου, μετράται κι αποθηκεύεται το ποσοστό που αυτό καταλαμβάνει στο σύνολο των σημείων της εικόνας. Η τελική έξοδος του αλγόριθμου είναι το χρώμα της λίστας με το μεγαλύτερο ποσοστό κάλυψης.

3.3.3. Εφαρμογές Υπολογισμού Κυρίαρχου Χρώματος

Ο αλγόριθμος, του οποίου τα στάδια αναλύθηκαν στις ενότητες 3.3.1 και 3.3.2, βρίσκει εφαρμογή σε δύο χαρακτηριστικά της πλατφόρμας που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας. Τα χαρακτηριστικά αυτά είναι η **αναγνώριση του τύπου πατώματος και του χρώματος των εξωτερικών τοίχων** της οικίας, όπως αυτή απεικονίζεται σε ανάλογες φωτογραφίες-εισόδους.

i) Το πρώτο χαρακτηριστικό απαιτεί την εισαγωγή μιας εικόνας εσωτερικού χρόνου. Την απαίτηση αυτή φροντίζει να ικανοποιεί το σύστημα του εξυπηρετητή, το οποίο αναλύεται στο κεφάλαιο 4. Έχοντας επομένως ως είσοδο μία τέτοια φωτογραφία, επιλέγονται από το σύνολο της τα σημεία εκείνα που αντιστοιχούν στο κάτω 1/3 της φωτογραφίας. Η επιλογή των συγκεκριμένων εικονοστοιχείων έγινε γνωρίζοντας ότι η συγκεκριμένη περιοχή εμπεριέχει στατιστικώς την πλειοψηφία των σημείων που απεικονίζουν το πάτωμα. Αποφεύγεται με αυτό τον τρόπο η πιθανότητα ελέγχου λανθασμένης περιοχής, αυξάνοντας έτσι την αποδοτικότητα του αλγόριθμου.

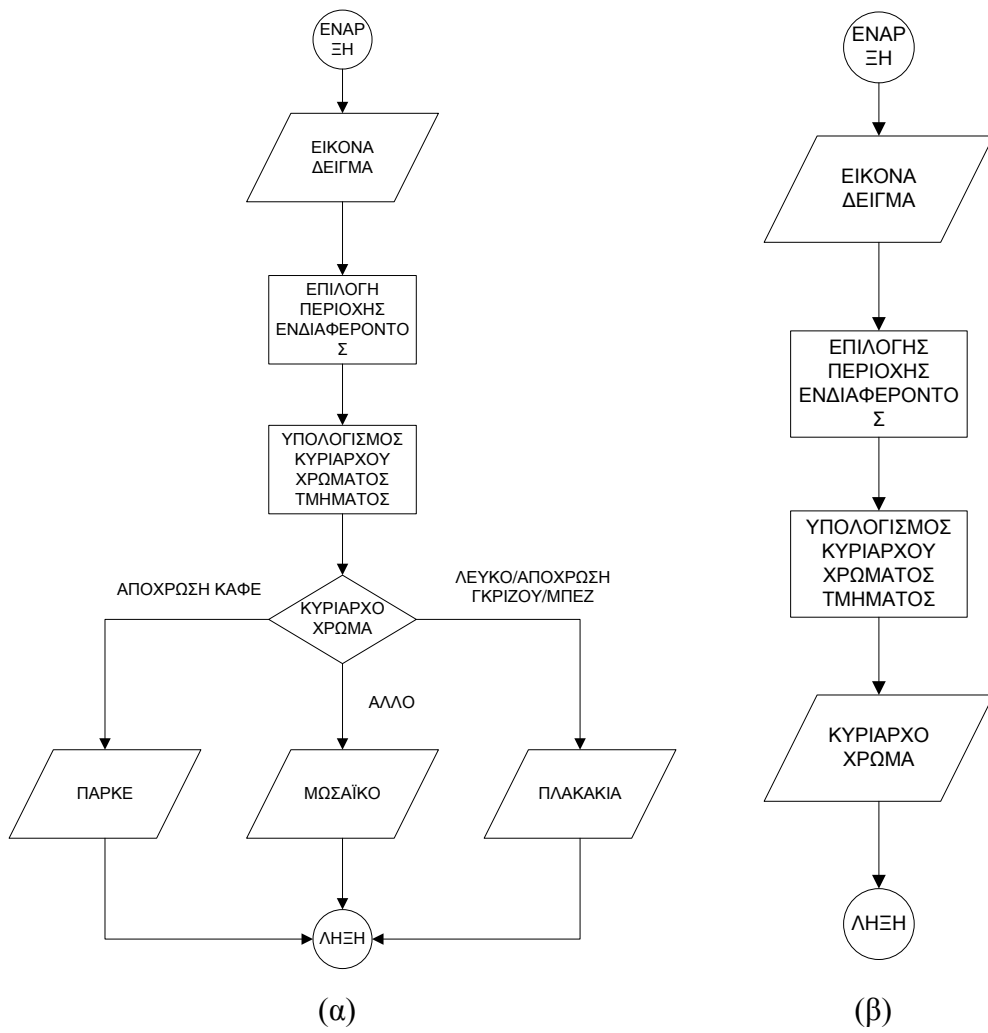
Στο τμήμα της φωτογραφίας που επιλέγεται εφαρμόζεται ο αλγόριθμος εύρεσης κυρίαρχου χρώματος. Ανάλογα με το κυρίαρχο χρώμα που προκύπτει στην έξοδο του αλγόριθμου, είναι δυνατή η επιλογή του κατάλληλου τύπου πατώματος. Αν για

παράδειγμα κυρίαρχο χρώμα επιλεγεί από τον αλγόριθμο το καφέ, τότε το σύστημα αποφαίνεται ότι ο τύπος πατώματος είναι παρκέ.

ii) Παρόμοια λειτουργία εμφανίζει και το δεύτερο χαρακτηριστικό, το οποίο απαιτεί την εισαγωγή μιας εικόνας εξωτερικού χρόνου. Και για αυτή την απαίτηση φροντίζει το σύστημα του εξυπηρετητή, όπως αναλύεται στο κεφάλαιο 4. Η διαφορά με το προαναφερθέν χαρακτηριστικό έγκεται στην επιλογή της περιοχής προς ανάλυση. Σε αυτή την περίπτωση, επιλέγονται από το σύνολο της εικόνας τα σημεία εκείνα που αντιστοιχούν σε ένα τμήμα που καλύπτει το μέσο της φωτογραφίας, επιφάνειας $4/5$ της συνολικής. Η επιλογή των συγκεκριμένων εικονοστοιχείων έγινε γνωρίζοντας ότι η συγκεκριμένη περιοχή εμπεριέχει στατιστικώς την πλειοψηφία των σημείων που απεικονίζουν ένα οίκημα, άρα και τους εξωτερικούς τοίχους αυτού. Αποφεύγεται έτσι και πάλι η πιθανότητα ελέγχου λανθασμένης περιοχής, αυξάνοντας την αποδοτικότητα του αλγόριθμου.

Η διαδικασία από το σημείο αυτό κι έπειτα είναι ίδια με της αναγνώρισης του τύπου πατώματος. Στο τμήμα της φωτογραφίας που έχει επιλεγεί εφαρμόζεται ο αλγόριθμος εύρεσης κυρίαρχου χρώματος, με διαφορετική όμως λίστα χρωμάτων σύγκρισης σε σχέση με την πρώτη περίπτωση. Το κυρίαρχο χρώμα που υπολογίζεται από τον αλγόριθμο, αποτελεί και την τελική επιλογή του συστήματος ως εξωτερικό χρώμα.

Ακολουθεί το διάγραμμα ροής του προαναφερθέντος αλγόριθμου.



Σχήμα 3.3.3.1: (α) Διάγραμμα ροής αλγόριθμου εύρεσης τύπου πατώματος.
 (β) Διάγραμμα ροής αλγόριθμου εύρεσης χρώματος τοίχου.

4. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό αναφέρονται με λεπτομέρεια τα χαρακτηριστικά της πλατφόρμας που υλοποιήθηκε στα πλαίσια της εκπόνησης αυτής της διπλωματικής εργασίας.

Το σύστημα αποτελείται από τα υποσυστήματα του πελάτη, του εξυπηρετητή και της βάσης δεδομένων. Ακολουθεί αναλυτική επεξήγηση του κάθε υποσυστήματος από το οποίο αποτελείται, ενώ τονίζονται και οι τρόποι επικοινωνίας μεταξύ των υποσυστημάτων. Τέλος, αναφέρονται και τα σχεδιαστικά εργαλεία και πλατφόρμες που χρησιμοποιήθηκαν, καθώς και οι λόγοι επιλογής τους.

Όπως έχει προαναφερθεί, σκοπός της εργασίας αυτής είναι η υλοποίηση μίας φορητής πλατφόρμας ηλεκτρονικού εμπορίου, η δυνατότητα χρήσης της οποίας θα παρέχεται από οποιαδήποτε φορητή συσκευή. Η υποστήριξη όλων των υπάρχοντων συσκευών όμως είναι πρακτικά μη υλοποιήσιμη, λόγω της ύπαρξης μεγάλου πλήθους διαφορετικών πλατφόρμων ανάπτυξης εφαρμογών για τις φορητές συσκευές, μη συμβατών μεταξύ τους. Ο αριθμός αυτός οφείλεται στο γεγονός της απουσίας τυποποίησης τεχνολογιών των φορητών συσκευών και στην αδυναμία συνεργασίας μεταξύ των κατασκευαστών. Επομένως, για τη μεγιστοποίηση των συμβατών με την πλατφόρμα της εργασίας αυτής συσκευών, κρίθηκε σκόπιμη η χρήση της Java Micro Edition – Java ME για το υποσύστημα του πελάτη, Java για το υποσύστημα του εξυπηρετητή και MySQL για τη βάση δεδομένων.

Σε σύγκριση με τις εγγενείς πλατφόρμες ανάπτυξης των φορητών συσκευών, το κυριότερο πλεονέκτημα της Java ME είναι η δυνατότητα ανάπτυξης φορητών εφαρμογών. Η φορητότητα της Java ME, και γενικότερα της γλώσσας Java και των επεκτάσεών της συνολικά, οφείλεται στο μοντέλο εκτέλεσής της. Πιο συγκεκριμένα, κατά την εκτέλεση μιας εφαρμογής Java, χρησιμοποιείται η Εικονική Μηχανή της Java (Java Virtual Machine – JVM) για τη μετατροπή του κώδικά της (bytecode) σε γλώσσα μηχανής, δημιουργώντας έτσι ένα επίπεδο συμβατότητας πάνω από το υλικό. Επίσης, θετικό στοιχείο της γλώσσας Java είναι η αντικειμενοστρέφεια της, που διευκολύνει τη σχεδίαση συστημάτων. Ακόμη, η γλώσσα Java μπορεί να χρησιμοποιηθεί ως εργαλείο ανάπτυξης, τόσο διαδικτυακών εφαρμογών με χρήση

τεχνολογίας Σελίδων Διακομιστή Java (Java Server Pages – JSP) και Servlets, όσο και εφαρμογών για φορητές συσκευές μέσω της τεχνολογίας των Midlet, οι οποίες συνεργάζονται άψογα μεταξύ τους. Παράλληλα παρέχει λύσεις για διασύνδεση με βάσεις δεδομένων που λειτουργούν με MySQL. Η χρήση της ίδιας γλώσσας για την ανάπτυξη δύο εκ των τριών υποσυστημάτων της πλατφόρμας, έχοντας παράλληλα λειτουργικό πλαίσιο ανάπτυξης επικοινωνίας αυτών με το τρίτο και τελευταίο υποσύστημα, αυτό της βάσης δεδομένων, είναι ένα ακόμα στοιχείο που λειτούργησε προσθετικά στην επιλογή της Java. Τέλος, αξίζει να σημειωθεί ότι η Java ME δεν υποστηρίζεται από τα τελευταία γενιάς κινητά τηλέφωνα, τα επονομαζόμενα έξυπνα τηλέφωνα (smartphones) δηλαδή. Εν τούτοις, αυτά αποτελούν πολύ μικρό μερίδιο της αγοράς, μικρότερο του 20% παγκοσμίως, το οποίο δεν ενδέχεται να αυξηθεί σημαντικά άμεσα [marketingcharts, 2010]. Συνεπώς, το γεγονός αυτό επηρεάζει ελάχιστα τη φορητότητα της πλατφόρμας.

Το σύστημα που υλοποιείται στα πλαίσια της διπλωματικής αυτής υποστηρίζει 3 επίπεδα ιεραρχίας χρηστών. Το πρώτο επίπεδο είναι αυτό των απλών χρηστών (users). Σε αυτούς παρέχεται η δυνατότητα απλής αναζήτησης αγγελιών, που είναι αποθηκευμένες στη βάση δεδομένων, κάνοντας χρήση μόνο δεδομένων και όχι φωτογραφιών. Το δεύτερο επίπεδο αφορά τους προχωρημένους χρήστες (advanced users). Σε αυτούς δίνεται η δυνατότητα απλής και προχωρημένης αναζήτησης, τόσο με χρήση δεδομένων, όσο και φωτογραφιών. Επίσης, επιτρέπεται να εισάγουν νέες αγγελίες με εικόνες χρησιμοποιώντας τα χαρακτηριστικά αυτόματης συμπλήρωσης φόρμας του συστήματος. Δεν τους δίνεται η δυνατότητα προσθήκης περαιτέρω εικόνων σε μία αγγελία όμως, εφ'όσον ολοκληρώσουν την καταχώρηση της αγγελίας. Το τρίτο και τελευταίο επίπεδο είναι αυτό των διαχειριστών (administrators). Σε αυτούς παρέχονται όλα τα παραπάνω χαρακτηριστικά, με επιπλέον δυνατότητα την προσθήκη φωτογραφιών σε μία αγγελία του συστήματος, οι οποίες λαμβάνονται εκείνη τη στιγμή με χρήση φορητής συσκευής.

Η υλοποίηση όλων των παραπάνω χαρακτηριστικών, και γενικότερα κάθε πτυχής των υποσυστημάτων του πελάτη και του εξυπηρετητή, έγινε κάνοντας χρήση του περιβάλλοντος ανάπτυξης εφαρμογών Netbeans IDE 6.9.

Πιο συγκεκριμένα, η υλοποίηση του υποσυστήματος πελάτη βασίστηκε στην τεχνολογία J2ME με configuration CLDC 1.1 και προφίλ MIDP 2.1. Για τον έλεγχο της ορθής λειτουργίας του, έγινε χρήση της πλατφόρμας προσομοίωσης της Sony Ericsson, SDK 2.5.0.6 για Java, με συσκευή προσομοίωσης την JP8_240x320, η οποία υποστήριζε και το πακέτο (package) Advanced Multimedia Supplements API.

Η υλοποίηση του υποσυστήματος εξυπηρετητή βασίστηκε στις τεχνολογίες JSP και Servlets, όπως αυτές υλοποιούνται στο Java Development Kit (JDK) 5. Επίσης, η ιστοσελίδα, που αποτελεί και το περιβάλλον διεπαφής μεταξύ χρήστη κι εφαρμογής εγκαταστάθηκε και διεκπεραίωσε αιτήσεις τοπικά (localhost), στον Apache-Tomcat Server 6.0.26, όπως αυτός ενσωματώνεται στο περιβάλλον Netbeans. Για τις μεθόδους ψηφιακής επεξεργασίας εικόνας που χρησιμοποιήθηκαν και εκτελέστηκαν στο υποσύστημα εξυπηρετητή, απαραίτητη για την αποδοτική υλοποίησή τους κρίθηκε η χρήση του πακέτου Java Advanced Imaging (jai) 1.1.3.

Τέλος, το υποσύστημα της βάσης δεδομένων στηρίχτηκε στον MySQL Server 5.1. Η δημιουργία και οι όποιες αλλαγές στο σχήμα της βάσης επετεύχθηκαν μέσω του MySQL Workbench 5.2 CE, ενώ για τη σύνδεση των υπόλοιπων υποσυστημάτων με τη βάση δεδομένων χρησιμοποιήθηκε ο MySQL Connector/ODBC 5.1.

4.1. Υποσύστημα Πελάτη

Το υποσύστημα πελάτη υλοποιήθηκε με χρήση τεχνολογίας MIDlet της Java ME. Παρέχει δυνατότητες εισόδου-εγγραφής στην πλατφόρμα για όλα τα επίπεδα χρηστών, εμφανίζοντας κατάλληλα πληροφοριακά μηνύματα σε περίπτωση επιτυχούς ή ανεπιτυχούς σύνδεσης. Ανάλογα το ιεραρχικό επίπεδο του κάθε χρήστη μετά από μία επιτυχημένη είσοδο στο σύστημα, εμφανίζονται όλες οι δυνατές επιλογές του σε μορφή στοιχείων μενού, ενώ παρέχονται σύντομες οδηγίες σε κάθε φόρμα, προς διευκόλυνση των χρηστών. Η επικοινωνία του υποσυστήματος πελάτη με το υποσύστημα του εξυπηρετητή επιτυγχάνεται με χρήση του πρωτοκόλλου Hypertext Transfer Protocol (HTTP).

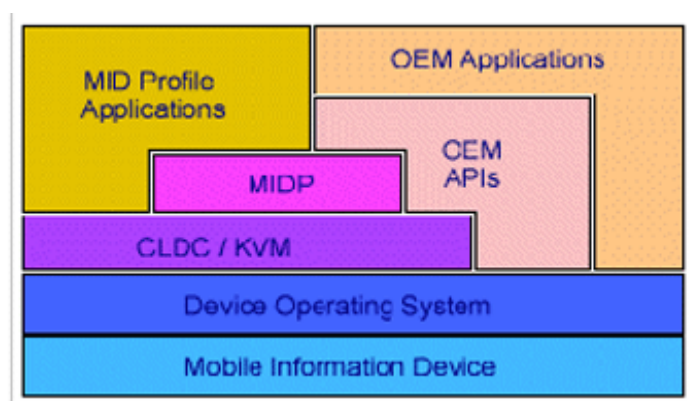
Αποτελείται από 4 κλάσεις. Η κύρια κλάση του υποσυστήματος είναι η User, η οποία υλοποιεί το γραφικό περιβάλλον διεπαφής με το χρήστη (Graphic User Interface – GUI) και παρέχει τις δυνατότητες εισόδου και εξόδου στο σύστημα. Πλαισιώνεται από 3 βοηθητικές κλάσεις. Η κλάση CameraCanvas υλοποιεί το GUI που εμφανίζεται στο χρήστη κατά τη διάρκεια προσπάθειας λήψης μίας φωτογραφίας μέσω της φορητής συσκευής. Η δεύτερη κλάση Base64 υλοποιεί μεθόδους κωδικοποίησης και αποκωδικοποίησης μιας ακολουθίας χαρακτήρων μέσω της κωδικοποίησης Base64. Τέλος, η κλάση SupportMethods, όπως υποδεικνύει και το όνομά της, παρέχει μεθόδους που σκοπό έχουν να υποστηρίξουν την επικοινωνία του πελάτη με το υποσύστημα του εξυπηρετητή.

4.1.1. Η Αρχιτεκτονική της Java ME - MIDP

Η πλατφόρμα Java 2 Micro Edition (J2ME) απευθύνεται στην ανάπτυξη εφαρμογών για φορητές συσκευές, όπως τα κινητά τηλέφωνα και τα PDAs. Οι συσκευές αυτές είναι περιορισμένης μνήμης και υπολογιστικής ισχύος, τροφοδοτούνται από μπαταρίες, έχουν μικρές οθόνες και περιορισμένους τρόπους εισαγωγής και εξαγωγής δεδομένων. Επίσης, είναι σύνηθες να υπάρχει η δυνατότητα σύνδεσης σε κάποιο δίκτυο. Ωστόσο, η πλατφόρμα Java ME δεν απευθύνεται μόνο σε απόλυτα μικρών δυνατοτήτων φορητές συσκευές. Ανάλογα με τους περιορισμούς που εισάγει το μέγεθος της μνήμης της συσκευής, μπορεί να χρησιμοποιηθεί είτε η αρχιτεκτονική CLDC (Connected Limited Device Configuration) είτε η αρχιτεκτονική CDC (Connected Device Configuration), με την πρώτη να παρέχει λιγότερες δυνατότητες, μιας και υποστηρίζει κινητά τηλέφωνα και άλλες περεμφερούς υπολογιστικής ισχύος συσκευές.

Η πλατφόρμα Java ME βασίζεται σε διαστρωματωμένη αρχιτεκτονική. Κάθε στρώμα προσθέτει επιπλέον λειτουργικότητες. Στο κατώτατο στρώμα βρίσκεται η Java Virtual Machine ενώ στο ανώτερο στρώμα κάποιο προφίλ. Το προφίλ είναι ανάλογο των αναγκών της εφαρμογής και των δυνατοτήτων της συσκευής στην οποία απευθύνεται η εφαρμογή. Επίσης, είναι δυνατός ο συνδυασμός προφίλ μεταξύ τους, με αυστηρά καθορισμένο τρόπο. Συνεπώς, δε μπορούν να συνεργαστούν όλα τα

προφίλ μεταξύ τους. Παράδειγμα ενός τέτοιου προφίλ είναι το Mobile Information Device Profile (MIDP), που αποτελεί την πιο δημοφιλή και συνήθη επιλογή.



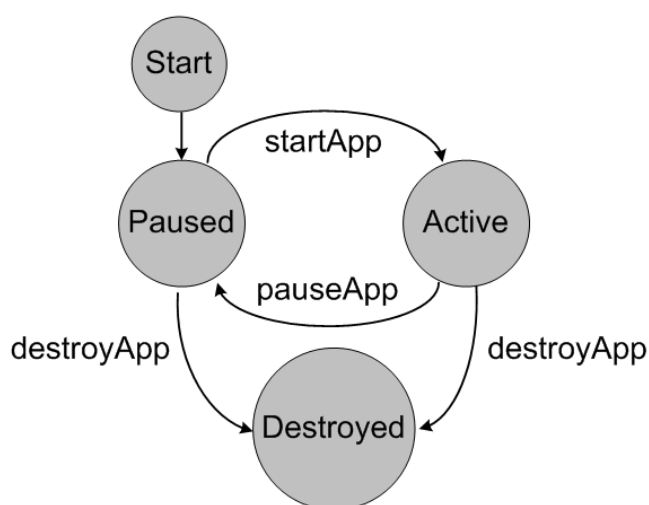
Σχήμα 4.1.1.1 Διαστρωματωμένη αρχιτεκτονική της J2ME

Το πρότυπο CLDC σχεδιάστηκε με απώτερο σκοπό την υποστήριξη εφαρμογών για συσκευές με περιορισμένες δυνατότητες όπως τα κινητά τηλέφωνα και τα PDAs. Η Java Virtual Machine που χρησιμοποιείται είναι πολύ μικρή σε μέγεθος. Αυτό έχει ως αποτέλεσμα αρκετά από τα χαρακτηριστικά της γλώσσας Java να μην υποστηρίζονται, να απουσιάζουν πάρα πολλές βιβλιοθήκες και όσες υπάρχουν, να έχουν συρρικνωθεί σε πολύ μεγάλο βαθμό. Για παράδειγμα, δεν υποστηρίζονται οι τύποι δεδομένων float και long ενώ η βιβλιοθήκη java.util περιέχει μόνο τις κλάσεις Vector, Random, Date και Calendar.

Αντιθέτως, το πρότυπο CDC σχεδιάστηκε με γνώμονα την υποστήριξη συσκευών με περισσότερες δυνατότητες από τις συσκευές τις οποίες εξυπηρετεί το πρότυπο CLDC. Το πρότυπο CDC απευθύνεται σε συσκευές που διαθέτουν επεξεργαστή των 32-bit και μέγεθος μνήμης μεγαλύτερο των 2MB. Όλα τα χαρακτηριστικά καθώς και οι βιβλιοθήκες της γλώσσας Java που απουσιάζουν από το CLDC πρότυπο, υποστηρίζονται από το CDC. Αυτό έχει ως συνέπεια, οι εφαρμογές που υλοποιούνται πάνω στο πρότυπο CDC να παρέχουν στους χρήστες μεγαλύτερο πλήθος δυνατοτήτων από αυτές που κτίζονται πάνω στο πρότυπο CLDC. Επίσης, το πρότυπο CDC προσφέρει καλύτερη δικτυακή υποστήριξη και πιο ευλύγιστους μηχανισμούς ασφάλειας.

Παρ'όλα αυτά, τόσο το πρότυπο CLDC όσο και το πρότυπο CDC έχουν πολύ περιορισμένη λειτουργικότητα. Για παράδειγμα, δεν προσφέρουν την δυνατότητα δημιουργίας γραφικού περιβάλλοντος για τον χρήστη. Επομένως, για τη δημιουργία μιας ολοκληρωμένης εφαρμογής, είναι απαραίτητη η χρησιμοποίηση ενός προφίλ, το οποίο βρίσκεται ιεραρχικά πάνω από αυτά τα πρότυπα στην αρχιτεκτονική της Java ME. Το δημοφιλέστερο τέτοιο προφίλ είναι το Mobile Information Device Profile ή αλλιώς MIDP.

Το MIDP συνδυάζεται με το πρότυπο CLDC και μεγιστοποιεί την αξιοποίηση των περιορισμένων δυνατοτήτων του τελευταίου, παρέχοντας ταυτόχρονα κι εργαλεία για τη δημιουργία γραφικού περιβάλλοντος για το χρήστη. Οι εφαρμογές σε MIDP αποκαλούνται MIDlets και μοιράζονται πλήθος κοινών στοιχείων με τα Java applets. Ένα MIDlet έχει τρεις καταστάσεις: active, paused και destroyed. Επίσης, τα MIDlets παρέχουν τη δυνατότητα αποθήκευσης δεδομένων με τη μορφή εγγραφών (records).



Σχήμα 4.1.1.2 Κύκλος ζωής ενός MIDlet

Εκτός από τις κλάσεις που παρέχει το πρότυπο CLDC, το προφίλ MIDP εισάγει τα εξής πακέτα:

- `javax.microedition.midlet`: Περιέχει μία και μοναδική κλάση με το όνομα `MIDlet`, η οποία, περιέχει μεταξύ άλλων, τις μεθόδους για την δημιουργία, την απενεργοποίηση και την καταστροφή των `MIDlets`.
- `javax.microedition.lcdui`: Το πακέτο αυτό περιέχει μεθόδους για την κατασκευή γραφικού περιβάλλοντος.
- `javax.microedition.rms`: Αυτό το πακέτο παρέχει την δυνατότητα για την αποθήκευση δεδομένων με την βοήθεια των μεθόδων της κλάσης `RecordStore`. Επιπλέον, περιέχει διεπαφές για τη σύγκριση, το φιλτράρισμα και άλλες λειτουργίες.

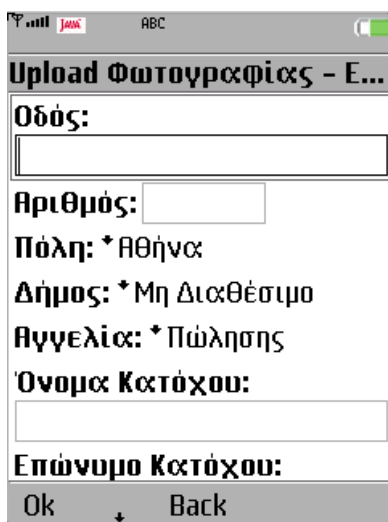
Εκτός από το προφίλ MIDP υπάρχει και το Foundation Profile, το οποίο συνδυάζεται με το πρότυπο CDC. Ουσιαστικά, το Foundation Profile επεκτείνει την λειτουργικότητα του MIDP, εξισώνοντας την με αυτήν που παρέχει η Java 2 Standard Edition (J2SE). Το Foundation Profile, σε αντίθεση με το MIDP, δεν περιέχει πακέτο για την δημιουργία γραφικού περιβάλλοντος χρήστη. Είναι εντυπωσιακό το γεγονός ότι η χρήση του είναι υποχρεωτική. Με άλλα λόγια, ακόμα και αν ο στόχος είναι η εργασία με κάποιο άλλο προφίλ, η παρουσία του Foundation Profile είναι υποχρεωτική. Παίζει το ρόλο του μεσολαβητή ανάμεσα στο πρότυπο CDC και το προς χρήση προφίλ.

Από την παραπάνω ανάλυση προκύπτει ότι για την υλοποίηση του υποσυστήματος πελάτη ο συνδυασμός τεχνολογιών που πρέπει να χρησιμοποιηθεί, ώστε να υπάρξει συμβατότητα του υποσυστήματος πελάτη με όσο το δυνατό μεγαλύτερο πλήθος συσκευών, είναι αυτός του προτύπου CLDC με το προφίλ MIDP.

4.1.2. Η κλάση MIDlet User

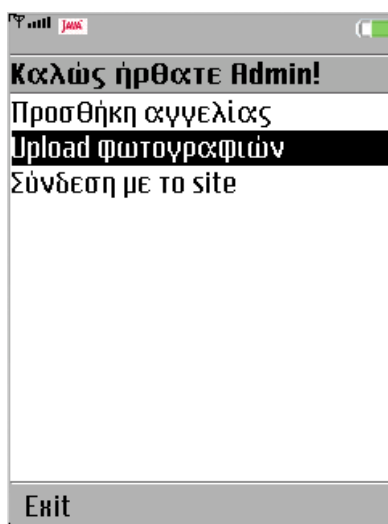
Η κλάση αυτή υλοποιείται με χρήση της έκδοσης 1.1 του προτύπου CLDC και της έκδοσης 2.1 του προφίλ MIDP. Αποτελεί τον κορμό του υποσυστήματος πελάτη, αφού αναλαμβάνει τη δημιουργία του γραφικού περιβάλλοντος διεπαφής με το χρήστη, επεξεργάζεται τις πληροφορίες που αυτός εισάγει και του επιστρέφει ως έξοδο στην οθόνη της συσκευής ανάλογες πληροφορίες. Αποτελείται από ένα σύνολο από φόρμες, λίστες και κουμπιά εντολών.

Οι φόρμες αποτελούν υποκλάση της κλάσης Screen, η οποία καθορίζει τι εμφανίζεται κάθε στιγμή στην οθόνη της συσκευής. Αποτελούνται από ένα σύνολο αντικειμένων: εικόνες, πεδία κειμένου, πεδία ημερομηνίας και ομάδες επιλογών για την εισαγωγή δεδομένων από το χρήστη ή την εμφάνιση αποτελεσμάτων και μηνυμάτων προς αυτόν.



Εικόνα 4.1.2.1 Παράδειγμα φόρμας του υποσυστήματος πελάτη

Οι λίστες χρησιμοποιούνται για τη δημιουργία λίστας επιλογών (menu) σε διάφορα σημεία της εφαρμογής-πελάτη. Μπορούν κι αυτές να περιλαμβάνουν οποιοδήποτε αντικείμενο επιτρέπεται και στις φόρμες.



Εικόνα 4.1.2.2 Παράδειγμα λίστας του υποσυστήματος πελάτη

Τέλος, τα κουμπιά εντολών χρησιμοποιούνται με σκοπό την επιβεβαίωση κάποιας ενέργειας. Αυτή μπορεί να είναι είτε εισαγωγή δεδομένων από το χρήστη, είτε κάποια εντολή πλοήγησης στην εφαρμογή. Οι τύποι των εντολών είναι αυστηρά καθορισμένοι και διακρίνονται σε:

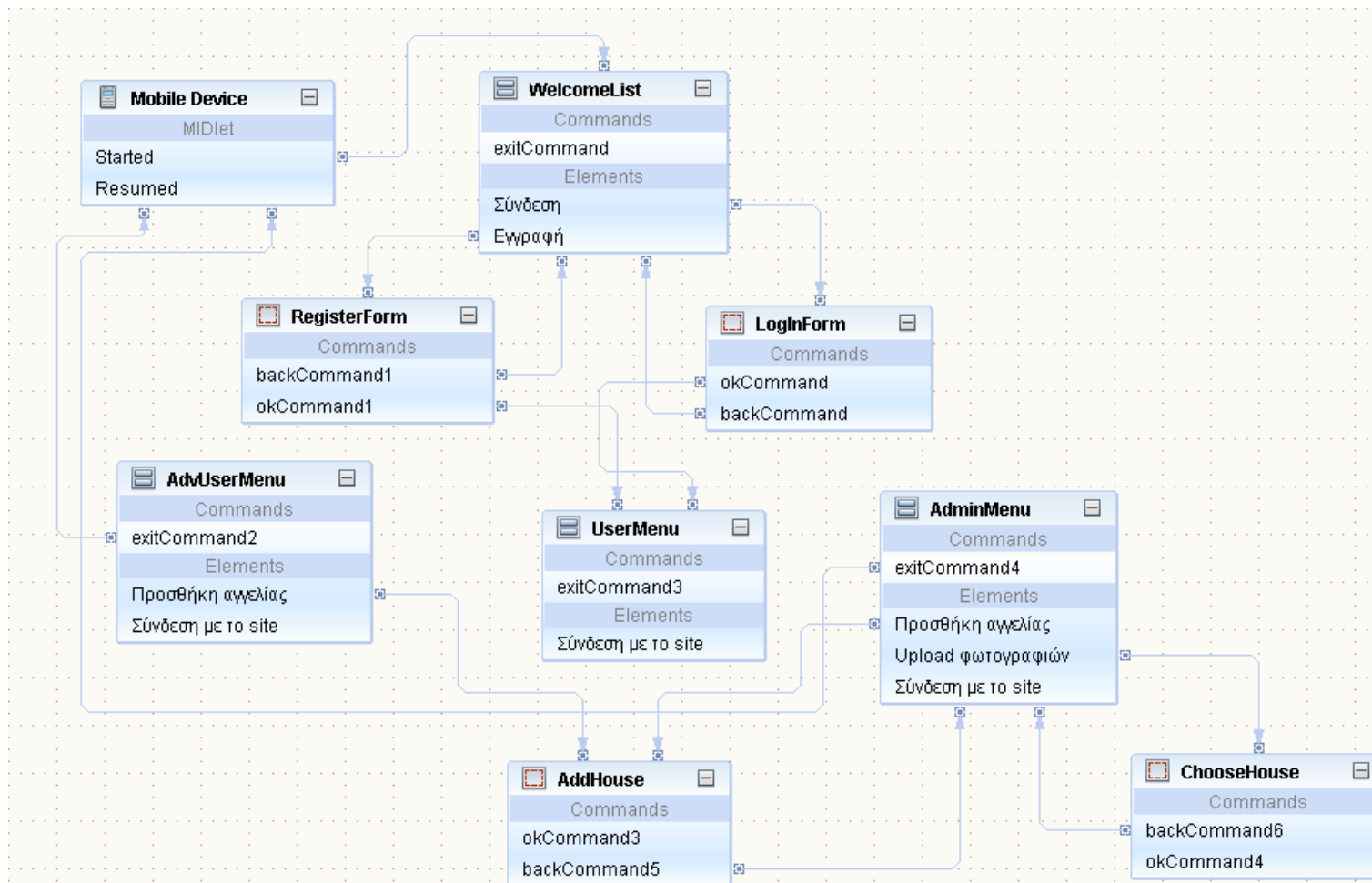
- **BACK:** Επιστροφή στην λογικά προηγούμενη (logically previous) οθόνη
- **CANCEL:** Αρνητική απάντηση στο διάλογο που υλοποιείται εκείνη τη στιγμή στην οθόνη.
- **EXIT:** Έξοδος από την εφαρμογή.
- **HELP:** Αναζήτηση βοήθειας μέσω διαδικτύου. Καμιά πληροφορία βοήθειας δεν εμφανίζεται αυτόματα από την υλοποίηση της εντολής **HELP**.
- **ITEM:** Η υλοποίηση της εντολής αυτής εξαρτάται από το αντικείμενο της οθόνης που είναι επιλεγμένο εκείνη τη στιγμή. Για παράδειγμα, αν το επιλεγμένο αντικείμενο είναι ένα πεδίο κειμένου, η εντολή **Item** θα μπορούσε να είναι η **Delete Written Text**.
- **OK:** Θετική απάντηση στο διάλογο που υλοποιείται εκείνη τη στιγμή στην οθόνη.
- **SCREEN:** Η υλοποίηση αυτής της εντολής εξαρτάται από την πληροφορία που εμφανίζεται εκείνη τη στιγμή στην οθόνη. Για παράδειγμα μια εντολή **Screen** θα μπορούσε να είναι η **Save**, αν στην οθόνη απεικονίζονταν μία εικόνα.
- **STOP:** Με την εντολή αυτή σταματά η διεργασία ή λειτουργία που εξυπηρετείται ή εκτελείται από τη συσκευή τη δεδομένη χρονική στιγμή.

Στη σελίδα που ακολουθεί, παρουσιάζεται ένα διάγραμμα ροής των οθόνων, μενού κι επιλογών που εμφανίζονται κατά την εκτέλεση του υποσυστήματος πελάτη.

Όσον αφορά τις μεθόδους που υλοποιούνται από την κλάση **Main**, αρχικά πρέπει να τονιστεί ότι η συγκεκριμένη κλάση αποτελεί υποκλάση της **MIDlet**. Οφείλει επομένως, να υλοποιεί και αυτή με τη σειρά της, τις μεθόδους **startApp()**, **pauseApp()** και **destroyApp(boolean unconditional)**. Επιπλέον, για την επίτευξη της δυνατότητας πλοήγησης δημιουργήθηκε η μέθοδος **commandAction(Command command, Displayable displayable)**. Η μέθοδος αυτή χρησιμοποιείται για να

υλοποιηθεί η ενέργεια που επιβάλλει η εντολή `command`, η οποία εμφανίζεται στην οθόνη `displayable`. Επίσης, για τη δυνατότητα χρήσης της κάμερας που ενδεχομένως να διαθέτει η φορητή συσκευή στην οποία έχει εγκατασταθεί το υποσύστημα, υλοποιήθηκαν δύο ακόμα μέθοδοι, η `showcamera()` και η `capture()`. Η πρώτη από αυτές εμφανίζει τη λήψη του φακού στην οθόνη κάνοντας χρήση της βοηθητικής κλάσης `cameraCanvas`, ενώ η δεύτερη αποθηκεύει την εικόνα του φακού όταν ενεργοποιηθεί η εντολή της λήψης φωτογραφίας. Αναλυτικά:

- **startApp()** : Η μέθοδος αυτή δε δέχεται ορίσματα κατά την κλήση της. Καλείται απαραίτητα κατά την εκκίνηση της εφαρμογής-πελάτη, αλλά μόνο όταν αυτή επανεκκινά μετά από παύση. Η λειτουργία της εξ'ορισμού περιορίζεται στην ενημέρωση του `MIDlet` ότι πλέον είναι ενεργό. Στην ενεργή κατάσταση το `MIDlet` έχει τη δυνατότητα να δεσμεύει πόρους συστήματος. Στην υλοποίηση της μεθόδου αυτής σε αυτή διπλωματική εργασία έχει γίνει και μία προσθήκη στη λειτουργία της. Το `MIDlet` ενημερώνεται και για την αρχική οθόνη που πρέπει να εμφανίσει στο χρήστη. Η οθόνη αυτή, είναι η `WelcomeList`, από την οποία ο χρήστης μπορεί να εισέλθει ή να εγγραφεί στο σύστημα.
- **pauseApp()** : Η μέθοδος αυτή δε δέχεται ορίσματα κατά την κλήση της. Καλείται κατά την παύση της εφαρμογής-πελάτη. Η λειτουργία της περιορίζεται στον σηματοδότηση του `MIDlet` ότι οφείλει να μπει σε κατάσταση παύσης, και να αποδεσμεύσει όποιο κοινό (`shared`) χώρο και πόρους είχε κρατήσει.
- **destroyApp(boolean unconditional)** : Η μέθοδος αυτή δέχεται ως όρισμα μία λογική πρόταση κατά την κλήση της. Μπορεί να κληθεί από όταν το `MIDlet` είναι σε ενεργή και σε κατάσταση παύσης. Ανάλογα με το αν το όρισμα της είναι αληθές ή ψευδές η λειτουργία της διαφοροποιείται. Αν είναι αληθές, τότε σηματοδοτεί το `MIDlet` ότι οφείλει να μπει σε κατάσταση καταστροφής, και να αποδεσμεύσει όποιο χώρο και πόρους έχει κρατήσει. Αν είναι ψευδές, κάνει `throw` μία `MIDletStateChangeException`, με σκοπό να επισημάνει ότι είναι επιθυμητό το `MIDlet` να παραμείνει στην ίδια κατάσταση όπου βρισκόνταν πριν την κλήση, είτε δηλαδή ενεργό, είτε σε παύση.



Σχήμα 4.1.2.1 Διάγραμμα ροής του MIDlet Main.

- **commandAction(Command command, Displayable displayable)** : Η μέθοδος αυτή δέχεται ως ορίσματα μία Command και ένα Displayable. Καλείται όταν σε μία οθόνη displayable επιλεγεί μία εντολή command. Στη συνέχεια, εκτελούνται οι ενέργειες που αντιστοιχούν στην επιλεγμένη εντολή, η οποία βρίσκεται στην οθόνη displayable. Οι ενέργειες αυτές είναι τεσσάρων ειδών. Αν το displayable αντιστοιχεί σε List, τότε η ενέργεια είναι η εμφάνιση στην οθόνη της συσκευής, της Screen που αντιστοιχεί στην επιλογή του ανάλογου στοιχείου της List. Αν το displayable αντιστοιχεί σε φόρμα (Form), τότε η ενέργεια είναι η συλλογή των δεδομένων που έχουν εισαχθεί σε αυτή από το χρήστη, η αποστολή τους στο υποσύστημα εξυπηρετητή και η επεξεργασία της απόκρισης αυτού, μέσω των μεθόδων της βοηθητικής κλάσης SupportMethods. Επίσης, αν το υποσύστημα βρίσκεται σε λειτουργία λήψης φωτογραφιών και η εντολή είναι για λήψη εικόνας, καλείται η μέθοδος capture. Τέλος, υπάρχει το ενδεχόμενο τα ορίσματα να είναι τέτοια ώστε να πρέπει να εμφανιστεί στην οθόνη το περιεχόμενο του φωτογραφικού φακού της συσκευής, να εισέλθει δηλαδή η συσκευή σε κατάσταση λήψης φωτογραφιών. Σε αυτή την περίπτωση, καλείται η μέθοδος showcamera().
- **showcamera()** : Η μέθοδος αυτή καλείται από τη μέθοδο commandAction, με σκοπό την έναρξη λειτουργίας λήψης φωτογραφιών από τη συσκευή. Για το σκοπό αυτό χρησιμοποιείται η βοηθητική κλάση CameraCanvas.
- **capture()** : Η μέθοδος αυτή καλείται από τη μέθοδο commandAction. Σκοπός της είναι η λήψη της εικόνας που αποτυπώνεται στο φακό της συσκευής τη στιγμή της κλήσης της, η μετατροπή της εικόνας σε κωδικοποιημένο αντικείμενο String μέσω της Base64 κωδικοποίησης και η αποστολή της στο υποσύστημα εξυπηρετητή προς αποθήκευση. Για τη λήψη της εικόνας γίνεται χρήση της μεθόδου getSnapshot(String encoding) της κλάσης VideoControl. Η μέθοδος αυτή δέχεται ως όρισμα τις παραμέτρους κωδικοποίησης της φωτογραφίας που θα ληφθεί και επιστρέφει ένα πίνακα από bytes, που αντιστοιχούν στα δεδομένα της εικόνας. Αν το όρισμα είναι null, τότε η φωτογραφία λαμβάνεται με τις εξ'ορισμού παραμέτρους που ορίζει η υλοποίηση της Java ME κάθε συσκευής. Μέσω επομένως του ορίσματος της παραπάνω μεθόδου, καθίσταται δυνατή η ομοιογένεια στα χαρακτηριστικά των φωτογραφιών που αποθηκεύονται στη βάση, που αποτελεί απαραίτητη

προϋπόθεση για την ομαλή λειτουργία του αλγόριθμου σύγκρισης εικόνων μέσω συζευγμένων ιστογραμμάτων. Με τη λήψη της φωτογραφίας με τα ζητούμενα από την εφαρμογή χαρακτηριστικά, εκτελείται η κωδικοποίηση και αποστολή της στον εξυπηρετητή με χρήση μεθόδων των βοηθητικών κλάσεων Base64 και SupportMethods αντίστοιχα.

Από τις παραπάνω κλάσεις καμία δεν έχει επιστρεφόμενες τιμές.

Άξια αναφοράς είναι και η μέθοδος που ακολουθήθηκε για την αντιμετώπιση των κλειδωμάτων (deadlocks) των συσκευών κατά την κλήση των μεθόδων επικοινωνίας μέσω HTTP πρωτοκόλλου με τον εξυπηρετητή καθώς και της κλήσης capture.

Όταν ξεκινά μια εφαρμογή να εκτελείται, δημιουργείται από το υποσύστημα του περιβάλλοντος χρήστη ένα νήμα (thread), το οποίο αναλαμβάνει την εξυπηρέτηση των επιλογών του χρήστη, που δημιουργούν γεγονότα (events). Το συγκεκριμένο νήμα ονομάζεται νήμα γεγονότων (event thread). Αποτέλεσμα της ύπαρξης αυτού του νήματος είναι πως κατά τη διάρκεια μιας διεργασίας, η οποία για λόγους ασφάλειας απαιτεί την συγκατάθεση του χρήστη, όπως για παράδειγμα η χρήση HTTP ή της κάμερας από μία τρίτη εφαρμογή, η συσκευή να σταματά να αποκρίνεται (deadlock). Αυτό συμβαίνει διότι στο νήμα γεγονότων εκτελείται η μέθοδος που αιτείται της άδειας του χρήστη, αλλά ο χρήστης αδυνατεί να δώσει τη συγκατάθεσή του, επειδή το γεγονός (event) της επιλογής θετικής απόφασης δε βρίσκει κάποιο νήμα να εκτελεστεί, μιας και το event thread είναι το μοναδικό νήμα εκείνη τη χρονική στιγμή.

Την λύση σε αυτό το πρόβλημα τη δίνει η χρήση νημάτων. Τα νήματα αναλαμβάνουν την εκτέλεση χρονοβόρων διαδικασιών ή διαδικασιών που απαιτούν ειδική άδεια από το χρήστη, αφήνοντας το event thread ελεύθερο να εκτελέσει τις ενέργειες που επιθυμεί ο χρήστης και να ανανεώνει το γραφικό περιβάλλον όποτε χρειάζεται.

Συνεπώς, για την αποφυγή των deadlocks στην υλοποίηση του υποσυστήματος-πελάτη, θεωρήθηκε απαραίτητο την εγκατάσταση HTTP συνδέσεων και τη λήψη φωτογραφιών να την αναλαμβάνουν νήματα. Πιο συγκεκριμένα, η μέθοδος poster της βοηθητικής κλάσης SupportMethods, η οποία είναι υπεύθυνη για την εγκατάσταση

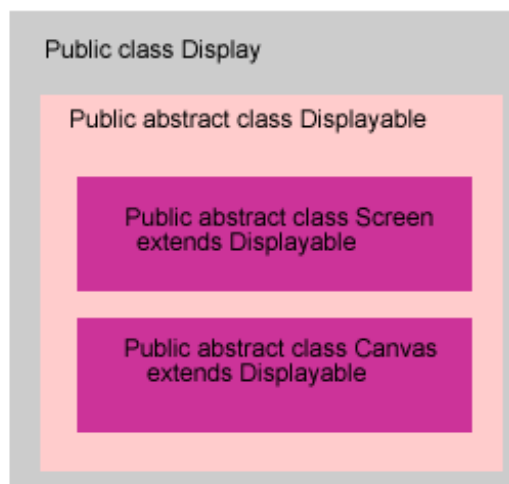
των HTTP συνδέσεων , δεν καλείται άμεσα. Αντίθετα, η μέθοδος `commandAction` δημιουργεί ένα νέο νήμα και καλεί την μέθοδο `poster` μέσα από τη μέθοδο `run` του εν λόγω νήματος. Η ίδια διαδικασία ακολουθείται και για τη μέθοδο `capture()`.

4.1.3. Η βοηθητική κλάση `CameraCanvas`

Η βοηθητική κλάση `CameraCanvas` σκοπό έχει την εμφάνιση στην οθόνη της συσκευής της εικόνας που λαμβάνεται από το φωτογραφικό φακό της, αν αυτός είναι διαθέσιμος. Η `CameraCanvas` αποτελεί υποκλάση της κλάσης `Canvas`.

Κάθε MIDlet έχει μπορεί να έχει μόνο ένα αντικείμενο της κλάσης `Display`, η οποία έχει παράγωγη την κλάση `Displayable`. Οτιδήποτε εμφανίζεται στην οθόνη της συσκευής, όπου εκτελείται η εφαρμογή, αποτελεί ένα στιγμιότυπο της κλάσης `Displayable`. Η κλάση `Displayable` έχει δύο υποκλάσεις, τις `Screen` και `Canvas`.

Η κλάση `Screen` και οι παράγωγες της χρησιμοποιούνται για την δημιουργία εργαλείων υψηλού επιπέδου. Η κλάση `Canvas` και οι παράγωγες αυτής χρησιμοποιούνται για την δημιουργία γραφικών χρησιμοποιώντας συναρτήσεις χαμηλού επιπέδου, που επιτρέπουν όμως την πλήρη διαχείριση της οθόνης της συσκευής. Με αυτό τον τρόπο κατασκευάζονται παντός τύπου γραφικά και δημιουργούνται εφαρμογές βασισμένες σε animation.



Σχήμα 4.1.3.1 Η κλάση `Display` και παράγωγες κλάσεις.

Για την κλάση Canvas, η οθόνη της συσκευής, είναι χωρισμένη σε τμήματα από ένα εικονικό πλέγμα, στο οποίο κάθε κελί αντιπροσωπεύει και ένα εικονοστοιχείο. Για τον προσδιορισμό αυτού, χρησιμοποιούνται συντεταγμένες γραμμής και στήλης μέσα στο πλέγμα. Λόγω μεγάλου πλήθους μοντέλων και χρήσεων των φορητών συσκευών, έπεται ότι και το μέγεθος της οθόνης διαφοροποιείται.

Απαραίτητος λοιπόν είναι πριν τον σχεδιασμό του γραφικού περιβάλλοντος, ο προσδιορισμός του μεγέθους της οθόνης της συσκευής (σε εικονοστοιχεία) που εκτελεί την εφαρμογή. Αυτό επιτυγχάνεται με δύο μεθόδους, που υλοποιούνται μέσα στην κλάση Canvas, τις `getWidth()` και `getHeight()`, οι οποίες επιστρέφουν το πλάτος και το ύψος της οθόνης της συσκευής αντίστοιχα.

Όσον αφορά τις μεθόδους που υλοποιούνται από την κλάση CameraCanvas, αρχικά πρέπει να τονιστεί ότι η συγκεκριμένη κλάση αποτελεί υποκλάση της Canvas. Συνεπώς, ήδη μπορεί να κάνει χρήση των υλοποιημένων στην κλάση Canvas μεθόδων `getWidth()` και `getHeight()`. Εκτός αυτών των μεθόδων όμως, υλοποιούνται ακόμα τρεις. Η μέθοδος κατασκευαστή (constructor method) `CameraCanvas(User midlet, VideoControl videoControl)`, η οποία καλείται κατά τη δημιουργία ενός αντικειμένου CameraCanvas και θέτει τη συσκευή όπου εκτελείται το MIDlet που δέχεται ως όρισμα σε κατάσταση λήψης φωτογραφιών. Επίσης, υλοποιείται η μέθοδος `paint()`, η οποία φροντίζει για την πλαισίωση της εικόνας που λαμβάνεται από το φακό της συσκευής με ένα πράσινου χρώματος πλαίσιο. Τέλος, η μέθοδος `keyPressed(int keycode)` φροντίζει για την κλήση της μεθόδου `capture()` της κλάσης Main, όταν πατηθεί οποιοδήποτε το κουμπί λήψης φωτογραφίας της συσκευής. Πιο αναλυτικά:

- **`getWidth()`** : Η μέθοδος αυτή δε δέχεται ορίσματα κατά την κλήση της. Καλείται κατά τη δημιουργία ενός αντικειμένου της κλάσης CameraCanvas. Η λειτουργία της εξ'ορισμού περιορίζεται στην επιστροφή της τιμής του πλάτους της οθόνης της συσκευής στην οποία εκτελείται.
- **`getHeight()`** : Η μέθοδος αυτή δε δέχεται ορίσματα κατά την κλήση της. Καλείται κατά τη δημιουργία ενός αντικειμένου της κλάσης CameraCanvas. Η

λειτουργία της εξ' ορισμού περιορίζεται στην επιστροφή της τιμής του ύψους της οθόνης της συσκευής στην οποία εκτελείται.

- **CameraCanvas(User midlet, VideoControl videoControl) :** Η μέθοδος αυτή δέχεται ως όρισμα ένα αντικείμενο ενός MIDlet κι ενός αντικειμένου VideoControl κατά την κλήση της. Αξίζει να σημειωθεί ότι το αντικείμενο VideoControl έχει οριστεί μέσα στο αντικείμενο midlet. Αρχικά, γίνεται προσδιορισμός των διαστάσεων της οθόνης μέσω των μεθόδων getHeight και getWidth. Στη συνέχεια, η μέθοδος αυτή θέτει τη συσκευή που εκτελείται το όρισμα midlet σε κατάσταση λήψης φωτογραφίας. Αυτό επιτυγχάνεται κάνοντας χρήση των μεθόδων setDisplayLocation, setDisplaySize και setVisible του αντικειμένου videocontrol. Μέσω της πρώτης από αυτές τις μεθόδους καθορίζεται η τοποθεσία του τμήματος της οθόνης που θα καταλαμβάνει η λήψη από το φακό. Έπειτα, η setDisplaySize καθορίζει το μέγεθος του τμήματος αυτού και τέλος επιλέγεται η εικόνα της λήψης να είναι ορατή από το χρήστη στην οθόνη της συσκευής του, μέσω της μεθόδου setVisible.
- **paint(Graphics g) :** Η μέθοδος αυτή δέχεται ως όρισμα κατά την κλήση της ένα αντικείμενο της κλάσης Graphics. Καλείται σε κάθε ανανέωση της οθόνης της συσκευής, η οποία βρίσκεται σε κατάσταση λήψης φωτογραφιών. Κύρια λειτουργία της είναι η σχεδίαση ενός πλαισίου, πράσινου χρώματος, γύρω από την εικόνα που λαμβάνεται από το φωτογραφικό φακό της συσκευής.
- **paint(int keyCode):** Η μέθοδος αυτή δέχεται ως όρισμα κατά την κλήση της έναν ακέραιο αριθμό, ο οποίος αντιστοιχεί σε ένα και μοναδικό πλήκτρο που πατήθηκε . Καλείται σε κάθε πάτημα ενός πλήκτρου από το χρήστη, όταν η συσκευή βρίσκεται σε κατάσταση λήψης φωτογραφιών. Κύρια λειτουργία της είναι ο έλεγχος, μέσω του κωδικού που δέχεται ως όρισμα, για το ποιο πλήκτρο πατήθηκε. Αν το πλήκτρο βρεθεί να είναι αυτό που αντιστοιχεί στη λήψη φωτογραφίας, τότε καλείται η μέθοδος capture() για την αποθήκευση της απεικονιζόμενης εκείνη τη στιγμή από το φακό εικόνας.

Από τις παραπάνω κλάσεις επιστρεφόμενες τιμές έχουν οι getWidth, getHeight, όπως αναφέρεται και στην περιγραφή τους, καθώς και η μέθοδος CameraCanvas, η οποία επιστρέφει ένα στιγμιότυπο της αντίστοιχης κλάσης.

4.1.4. Η βοηθητική κλάση SupportMethods

Η κλάση αυτή περιλαμβάνει τις μεθόδους, οι οποίες είναι υπεύθυνες για τη δημιουργία και εγκατάσταση συνδέσεων HTTP με τον εξυπηρετητή. Ουσιαστικά πρόκειται για τις μεθόδους, που εξασφαλίζουν τη δυνατότητα επικοινωνίας μεταξύ των υποσυστημάτων πελάτη και εξυπηρετητή.

Κύριο άξονα της επιλογής του πρωτοκόλλου HTTP αποτέλεσε το γεγονός ότι το υποσύστημα πελάτη, που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας, κάνει χρήση του προφίλ MIDP. Το προφίλ αυτό, υποστηρίζει αποτελεσματικά μόνο το πρωτόκολλο HTTP.

Οι μέθοδοι της κλάσης αυτής είναι τέσσερις. Η πρώτη μέθοδος, `poster(String post, String target)`, είναι υπεύθυνη για την εγκατάσταση της σύνδεσης μεταξύ πελάτη και υποσυστήματος εξυπηρετητή καθώς και για την αποστολή και λήψη δεδομένων από αυτόν. Η `checkRegister(String name)` ελέγχει σε περίπτωση προσπάθειας εγγραφής, αν αυτή είναι επιτυχής. Η τρίτη μέθοδος, `checkLogIn(String data)`, φροντίζει για τον έλεγχο της επιτυχούς εισόδου ενός χρήστη στο σύστημα. Τέλος, η `findHouse(String data)` αναλαμβάνει να επιστρέψει τον αριθμό ταυτότητας (house id) της οικίας που αντιστοιχεί στα δεδομένα που έχει εισάγει ο χρήστης στην εφαρμογή-πελάτη. Αναλυτικά:

- **`poster(String post, String target)`:** Η μέθοδος αυτή δέχεται ως ορίσματα δύο Strings. Η πρώτη αντιστοιχεί στα δεδομένα τα οποία θα αποσταλούν στο υποσύστημα του εξυπηρετητή. Η δεύτερη, ορίζει το στοιχείο του εξυπηρετητή, στο οποίο θα αποσταλούν τα δεδομένα. Αρχικά, η μέθοδος αυτή προβαίνει στην εγκατάσταση σύνδεσης HTTP μεταξύ πελάτη και εξυπηρετητή. Για το σκοπό αυτό χρησιμοποιήθηκε το `interface HTTPConnection` σε συνδυασμό με την κλάση `Connector` του πακέτου `javax.io`. Πιο συγκεκριμένα, η μέθοδος `open(String target)`, ανοίγει μια σύνδεση με την τοποθεσία που καθορίζεται από την τιμή της παραμέτρου `target`, ενώ η μέθοδος `setRequestMethod` καθορίζει τη μέθοδο αποστολής δεδομένων, η οποία μπορεί να είναι είτε GET, είτε POST. Η αποστολή των δεδομένων `post` επιτυγχάνεται με τη χρήση ενός `DataOutputStream` προς την

τοποθεσία που ορίζεται από το όρισμα `target`. Το άνοιγμα του `DataStream` αναλαμβάνει η μέθοδος `openDataOutputStream` της κλάσης `HttpConnection`. Με την ολοκλήρωση της αποστολής δεδομένων, γίνεται χρήση της μεθόδου `openInputStream`, ούτως ώστε να ληφθούν τα δεδομένα της απάντησης (`response`) του εξυπηρετητή μέσω ενός `InputStream`. Τα δεδομένα αυτά αποτελούν και την επιστρεφόμενη τιμή της μεθόδου. Ακολουθεί η επεξεργασία της απάντησης του εξυπηρετητή, μέσω των υπόλοιπων τριών μεθόδων της κλάσης.

- **`checkRegister(String name)`:** Η μέθοδος αυτή δέχεται ένα όρισμα, το οποίο αντιστοιχεί στο όνομα χρήστη προς έλεγχο. Ο έλεγχος που λαμβάνει χώρα έχει να κάνει με την ύπαρξη του ονόματος χρήστη στη βάση δεδομένων, εφ' όσον κάθε ένα από αυτά πρέπει να είναι μοναδικό. Για την εκτέλεση του ελέγχου αυτού, αποστέλλεται στο `Servlet` του εξυπηρετητή με όνομα `checkRegistering`, μέσω της μεθόδου `poster`, το όνομα χρήστη. Αν η απάντηση που ληφθεί είναι θετική, τότε η μέθοδος αυτή επιστρέφει `true`, αλλιώς επιστρέφει `false` και εμφανίζει αντίστοιχο πληροφοριακό μήνυμα στην οθόνη της συσκευής, όπου εκτελείται.
- **`checkLogIn(String data)`:** Η μέθοδος αυτή δέχεται ένα `String`, το οποίο αντιστοιχεί στο όνομα και κωδικό χρήστη προς έλεγχο. Ο έλεγχος που λαμβάνει χώρα, έχει να κάνει με την ταυτοποίηση του ονόματος και κωδικού χρήστη στη βάση δεδομένων. Για την εκτέλεση του ελέγχου αυτού, αποστέλλονται στο `Servlet` του εξυπηρετητή με όνομα `checkLogIn`, μέσω της μεθόδου `poster`, το όνομα κι ο κωδικός χρήστη. Μέσω επεξεργασίας της απάντησης που λαμβάνεται, αναγνωρίζεται το επίπεδο ιεραρχίας του χρήστη, δηλαδή αν είναι `user`, `advanced user` ή `administrator`, το οποίο αποτελεί και την επιστρεφόμενη τιμή της μεθόδου. Σε περίπτωση ανεπιτυχούς ταυτοποίησης, επιστρέφεται η τιμή `"none"` και εμφανίζεται αντίστοιχο πληροφοριακό μήνυμα στην οθόνη της συσκευής, όπου εκτελείται.
- **`findHouse(String data)`:** Η μέθοδος αυτή δέχεται ένα όρισμα, το οποίο αντιστοιχεί στα στοιχεία της οικίας που αναζητείται στη βάση δεδομένων των αγγελιών. Ο έλεγχος που λαμβάνει χώρα, έχει να κάνει με την ταυτοποίηση των δεδομένων του ορίσματος `data` στη βάση δεδομένων. Για την εκτέλεση του ελέγχου αυτού, αποστέλλονται στο `Servlet` με όνομα `findHouse` του

εξυπηρετητή, μέσω της μεθόδου `poster`, τα στοιχεία της οικίας προς αναζήτηση. Αν η οικία βρεθεί, τότε η μέθοδος αυτή επιστρέφει τον μοναδικό κωδικό αριθμό που αντιστοιχεί σε αυτή. Αλλιώς, επιστρέφει “none” και εμφανίζει αντίστοιχο πληροφοριακό μήνυμα στην οθόνη της συσκευής, όπου εκτελείται.

4.1.5. Η βοηθητική κλάση `Base64`

Η τελευταία κλάση του υποσυστήματος πελάτη εμπεριέχει μεθόδους για την κωδικοποίηση μιας ακολουθίας χαρακτήρων και συμβόλων, με βάση τη μέθοδο κωδικοποίησης `Base64`.

Η μέθοδος κωδικοποίησης `Base64` παίρνει τρία bytes, καθένα από τα οποία αποτελείται από οκτώ bits, και τα παρουσιάζει ως τέσσερις χαρακτήρες του προτύπου `ASCII`. Αυτό επιτυγχάνεται σε δύο στάδια:

Στο πρώτο στάδιο τα 3 bytes μετατρέπονται σε 4 εξαψήφιους αριθμούς. Κάθε χαρακτήρας του προτύπου `ASCII` αποτελείται από 7 bits. Η κωδικοποίηση `Base64` χρησιμοποιεί μόνο 6 bits, τα οποία αντιστοιχούν σε $2^6=64$ χαρακτήρες, ούτως ώστε να είναι βέβαιο ότι οι κωδικοποιημένοι χαρακτήρες είναι αντιληπτοί και αναγνώσιμοι από τον άνθρωπο. Κανένας ειδικός χαρακτήρας του `ASCII` δε χρησιμοποιείται. Οι 64 χαρακτήρες, από όπου προκύπτει και το όνομα `Base64`, είναι 10 νούμερα, 26 κεφαλαίοι χαρακτήρες, 26 μικροί χαρακτήρες και τα σύμβολα + και /.

Για παράδειγμα, αν τα τρία bytes είναι 155,162 και 233, η αντίστοιχη δυαδική ακολουθία είναι η 100110111010001011101001, η οποία με τη σειρά της αντιστοιχεί στις τιμές των 6 bit 38,58,11 και 41.

Οι τελευταίοι αυτοί αριθμοί μετατρέπονται σε χαρακτήρες `ASCII` στο δεύτερο στάδιο. Το στάδιο αυτό είναι η μετατροπή τιμών με βάση τον πίνακα κωδικοποίησης `Base64`. Οι αριθμοί αυτοί των 6 bit μεταφράζονται στην ακολουθία `ASCII` χαρακτήρων `m6Lp`.

Συγκεντρωτικά, έχουμε τις εξής μετατροπές:

- 155 -> 10011011
- 162 -> 10100010
- 233 -> 11101001

- 100110 -> 38
- 111010 -> 58
- 001011 -> 11
- 101001 -> 41

- 38 -> m
- 58 -> 6
- 11 -> L
- 41 -> p

Η ίδια ακολουθία των δύο παραπάνω βημάτων επαναλαμβάνεται, μέχρις ότου όλη η ακολουθία των bytes να έχει κωδικοποιηθεί.

Η βοηθητική κλάση Base64 αποτελείται από δύο μεθόδους, τις encode(String input) και decode(String output), οι οποίες αναλαμβάνουν την κωδικοποίηση και αποκωδικοποίηση αντίστοιχα της συμβολοσειράς που δέχονται ως όρισμα, με χρήση της μεθόδου Base64 που περιγράφηκε παραπάνω. Επιστρέφουν την κωδικοποιημένη και αποκωδικοποιημένη συμβολοσειρά, αντίστοιχα.

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Πίνακας 4.1.5.1 Πίνακας κωδικοποίησης μεθόδου Base64.

4.2. Υποσύστημα Εξυπηρετητή

Το υποσύστημα εξυπηρετητή υλοποιήθηκε με χρήση τεχνολογίας JSP και Servlet της Java. Παράλληλα, για την προσθήκη επιπλέον λειτουργικότητας και χαρακτηριστικών, έγινε χρήση και της γλώσσας σεναρίων Javascript.

Το υποσύστημα του εξυπηρετητή μπορεί να χωριστεί σε 2 τμήματα. Το πρώτο από αυτά, αποτελεί τη διεπαφή (front end) μεταξύ χρηστών της πλατφόρμας και του εξυπηρετητή. Πρόκειται για υλοποίηση μίας τυπικής ιστοσελίδας, που σκοπό έχει να επιδείξει τις δυνατότητες και χαρακτηριστικά του συστήματος. Το δεύτερο και τελευταίο τμήμα, είναι αυτό που ουσιαστικά υλοποιεί τα χαρακτηριστικά και τη λειτουργικότητα της πλατφόρμας. Παράλληλα, αναλαμβάνει την επίτευξη της επικοινωνίας μεταξύ του εξυπηρετητή και των υποσυστημάτων του πελάτη και της βάσης δεδομένων, με χρήση του πρωτοκόλλου HTTP και της γλώσσας MySQL αντίστοιχα. Πιο συγκεκριμένα, φροντίζει για την εξυπηρέτηση όλων των αιτήσεων της διεπαφής και του υποσυστήματος πελάτη, είτε παραδείγματος χάριν πρόκειται για αίτημα εισόδου ενός χρήστη στην πλατφόρμα, είτε επιστροφή αποτελεσμάτων από αναζήτηση στη βάση δεδομένων.

Η ιστοσελίδα, η οποία ουσιαστικά αποτελεί τη διεπαφή μεταξύ χρηστών και εξυπηρετητή, υλοποιείται από έναν αριθμό σελίδων Hyper Text Mark Up Language (HTML) καθώς και δυναμικών σελίδων JSP. Οι σελίδες αυτές ορίζουν τη μορφή της παραπάνω ιστοσελίδας, και επιτρέπουν τόσο την είσοδο δεδομένων από το χρήστη στην πλατφόρμα, όσο και την προβολή δεδομένων ως έξοδο. Η υλοποίηση της ιστοσελίδας ολοκληρώνεται με την προσθήκη ενός βοηθητικού αρχείου CSS, στο οποίο περιγράφονται οι κλάσεις CSS. Οι κλάσεις αυτές καθορίζουν το παρουσιαστικό (look and feel) της ιστοσελίδας, ενώ παράλληλα επιτρέπουν την εύκολη διαχείριση ή αλλαγή αυτού. Τέλος, επιπλέον χαρακτηριστικά της ιστοσελίδας, όπως έλεγχοι συμπλήρωσης φόρμας, έχουν προστεθεί μέσω ενός αρχείου Javascript, στο οποίο περιέχονται μέθοδοι που τα υλοποιούν.

Το δεύτερο τμήμα του υποσυστήματος εξυπηρετητή, αποτελείται από μεγάλο αριθμό υποκλάσεων της κλάσης Servlet, οι οποίες αναλαμβάνουν την διεκπεραίωση των αιτήσεων που προκύπτουν τόσο από την ιστοσελίδα, όσο και από το υποσύστημα της

πλατφόρμας πελάτη. Επίσης, αναλαμβάνουν τη διασύνδεση του εξυπηρετητή με τη βάση δεδομένων, στην οποία εκτελούν λειτουργίες ανάλογα με τις αιτήσεις που δέχονται. Τέλος, το τμήμα αυτό συμπληρώνεται από έναν αριθμό βοηθητικών κλάσεων, οι οποίες σκοπό έχουν την εκτέλεση συγκεκριμένων εξειδικευμένων εργασιών, όπως ψηφιακή επεξεργασία των εικόνων της βάσης και κωδικοποίηση συμβολοσειρών σύμφωνα με τη μέθοδο Base64.

Ακολουθεί μία γενική αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν, καθώς και αναλυτική παρουσίαση των προαναφερθέντων τμημάτων του υποσυστήματος εξυπηρετητή.

4.2.1. Τεχνολογία Web Components

Υπάρχουν δυο είδη Web components, τα Java Servlet και οι Java Server Pages (JSP). Και τα δυο είδη επιτρέπουν την δυναμική επεξεργασία και παραγωγή δεδομένων, ανταποκρινόμενα στην είσοδο. Τα servlets είναι ουσιαστικά ισοδύναμα με τα JSPs, μιας και τα JSPs μεταφράζονται από το σύστημα εξυπηρετητή στον οποίο είναι φορτωμένα, σε Servlet κατά τη διάρκεια της πρώτης φόρτωσής τους στη μνήμη. Συνεπώς, οποιαδήποτε λειτουργικότητα ή χαρακτηριστικό παρέχεται από ένα Servlet, μπορεί ισοδύναμα να παραχθεί και από JSP. Ωστόσο, δεν είναι καθ'όλα όμοια.

Η κάθε μορφή έχει συγκεκριμένο σκοπό και πλεονεκτεί έναντι της άλλης σε διαφορετικά σημεία. Πιο συγκεκριμένα, τα Servlet είναι καλύτερα στην παραγωγή δυαδικών δεδομένων, στη δημιουργία ιστοσελίδων με πολύ μεταβλητή δομή και στην εκτέλεση λειτουργιών που απαιτούν ελάχιστη ή και καθόλου έξοδο προς το χρήστη, όπως για παράδειγμα ανακατεύθυνση (redirection). Σε αντίθεση με τα Servlet, οι σελίδες JSP είναι ιδανικές για τη δημιουργία ιστοσελίδων, οι οποίες αποτελούνται από μεγάλα τμήματα σταθερά δομημένου κώδικα HTML ή άλλων δεδομένων. Γενικά, η απόφαση για τη χρήση Servlet ή JSP εξαρτάται κάθε φορά από τις ανάγκες της προς υλοποίηση εφαρμογής.

Τα Servlet είναι κλάσεις γραμμένες σε γλώσσα Java, οι οποίες μπορούν να επεξεργαστούν δυναμικά κλήσεις (requests) και να παράγουν αποκρίσεις (responses).

Ένα servlet εκτελείται σε ένα σύστημα εξυπηρετητή και όχι στα τερματικά των πελατών, δηλαδή είναι μία συνιστώσα λογισμικού της πλευράς του εξυπηρετητή (server-side component) και επομένως μπορεί να χρησιμοποιήσει όλα τα χαρακτηριστικά της γλώσσας Java. Αν και τα Servlet μπορούν να ανταποκριθούν σε κάθε είδους αίτηση και να παράγουν κάθε είδους απόκριση, η επικοινωνία λαμβάνει χώρα συνήθως μέσω του πρωτοκόλλου HTTP. Οι βιβλιοθήκες javax.servlet και javax.servlet.http παρέχουν τις κατάλληλες διεπαφές και κλάσεις για την υλοποίηση Servlet.

Τα Servlet παρουσιάζουν τα ίδια πλεονεκτήματα μεταφερσιμότητας με τη γλώσσα Java. Επίσης, αποδίδουν πολύ ικανοποιητικά σε θέματα διαχείρισης μνήμης, διότι χρησιμοποιούν πολυεπεξεργασία (multitasking). Αυτό σημαίνει ότι η παράλληλη χρήση ενός Servlet από μεγάλο πλήθος χρηστών οδηγεί στη δημιουργία αντίστοιχου αριθμού νημάτων (threads), και όχι στην καταχώρηση περισσότερων του ενός αντιτύπων του Servlet στη μνήμη.

4.2.2. Τμήμα Διεπαφής – Ιστοσελίδα Εξυπηρετητή

Το τμήμα της διεπαφής του υποσυστήματος εξυπηρετητή παρέχει όλες τις δυνατότητες εισόδου και εξόδου από και προς το χρήστη της πλατφόρμας αντίστοιχα. Πρόκειται ουσιαστικά για την υλοποίηση του περιβάλλοντος, στο οποίο ο χρήστης εισάγει, μέσω συγκεκριμένης κάθε φορά φόρμας, νέες αγγελίες στο σύστημα, εκτελεί αναζητήσεις και παρατηρεί τα αποτελέσματά τους, όπως αυτά εμφανίζονται. Πιο αναλυτικά:

- **Index.jsp:** Αποτελεί κεντρικό άξονα της υλοποίησης της διεπαφής. Όπως φαίνεται και στην εικόνα 4.2.2.1, η σελίδα αυτή χωρίζεται σε τρία τμήματα (DIVs). Το πρώτο από αυτά, βρίσκεται τοποθετημένο βόρεια. Πρόκειται για μία λεπτή λωρίδα, στην οποία εμφανίζεται το λογότυπο της ιστοσελίδας και έχει καθαρά πληροφοριακό χαρακτήρα. Το δεύτερο τμήμα τοποθετείται ανατολικά. Περιέχει τα πλήκτρα πλοήγησης στην ιστοσελίδα με τη μορφή ενός μενού. Μέσω αυτού, ο χρήσης μπορεί να μεταφερθεί ασύγχρονα στις υπόλοιπες σελίδες του συνόλου που αποτελούν τον διαδικτυακό τόπο του υποσυστήματος εξυπηρετητή, χωρίς να απαιτείται πλήρης φόρτωση των

στοιχείων του μενού και του λογότυπο ξανά. Το τρίτο και τελευταίο τμήμα καταλαμβάνει τον υπόλοιπο χώρο. Το περιεχόμενο του πλαισίου αυτού μεταβάλλεται ανάλογα με τις επιλογές πλοήγησης του χρήστη. Πρόκειται ουσιαστικά για το χώρο, στον οποίο εμφανίζονται όλες οι υπόλοιπες σελίδες του ιστοχώρου, καθώς και όλα τα πληροφοριακά μηνύματα. Αποτελεί δηλαδή, το τμήμα περιεχομένου (content DIV). Η σελίδα index.jsp είναι επιφορτισμένη με την εμφάνιση των κατάλληλων επιλογών πλοήγησης των χρηστών της πλατφόρμας. Έτσι, κατά τη φόρτωσή της γίνεται έλεγχος των παραμέτρων της εκάστοτε συνεδρίας. Εάν βρεθεί κάποιο όνομα χρήστη αποθηκευμένο στη συνεδρία, εμφανίζονται στο τμήμα μενού οι αντίστοιχες με το επίπεδο δικαιωμάτων του επιλογές πλοήγησης. Σε περίπτωση που δε βρεθεί κάποιος χρήστης, εμφανίζεται μήνυμα, στο οποίο προτρέπει να χρησιμοποιήσει το μενού, ούτως ώστε να εγγραφεί ή να συνδεθεί στην πλατφόρμα. Τέλος, το index.jsp είναι υπεύθυνο και για την εμφάνιση όλων των μηνυμάτων πληροφοριακής φύσεως, όπως για παράδειγμα σε περίπτωση ανεπιτυχούς σύνδεσης του υποσυστήματος εξυπηρετητή με τη βάση δεδομένων, όπου το γεγονός γνωστοποιείται στο χρήστη με αντίστοιχο μήνυμα.

- **about.html:** Πρόκειται για μία απλή στατική ιστοσελίδα κώδικα HTML, η οποία παρέχει πληροφοριακά στοιχεία για τον ιστοτόπο που υλοποιήθηκε.
- **addhouse.html:** Πρόκειται για απλή στατική ιστοσελίδα κώδικα HTML. Στη σελίδα αυτή, παρέχεται η φόρμα δεδομένων με τα απαραίτητα στοιχεία για την εισαγωγή μίας νέας αγγελίας χωρίς εικόνες στη βάση δεδομένων. Αφού τα στοιχεία συμπληρωθούν και γίνει ο απαραίτητος έλεγχος εγκυρότητας με χρήση ενός σεναρίου Javascript, μεταφέρονται μέσω μίας HTTP αίτησης στο Servlet AddHouseService, το οποίο αναλαμβάνει και την εισαγωγή τους στη βάση δεδομένων.
- **contact.html:** Πρόκειται για απλή στατική ιστοσελίδα κώδικα HTML. Στη σελίδα αυτή, παρέχεται η δυνατότητα επικοινωνίας με τους διαχειριστές του συστήματος εξυπηρετητή, μέσω δύο διαφορετικών μεθόδων. Ο πρώτος τρόπος επικοινωνίας, είναι μέσω της παρεχόμενης διεύθυνσης ηλεκτρονικού ταχυδρομείου. Ο δεύτερος τρόπος, είναι με χρήση της φόρμας που βρίσκεται στη σελίδα αυτή. Με τη συμπλήρωση της φόρμας, τα δεδομένα αυτής

αποστέλλονται στο servlet `contactService`, το οποίο αναλαμβάνει την εισαγωγή τους στον αντίστοιχο πίνακα της βάσης δεδομένων.

- **home.html:** Πρόκειται για μία απλή στατική ιστοσελίδα κώδικα HTML, η οποία λειτουργεί ως αρχική σελίδα του ιστοτόπου της διαδικτυακής πλατφόρμας που υλοποιήθηκε. Παρέχει οδηγίες στους χρήστες για την ευκολότερη χρήση της πλατφόρμας. Τέλος, είναι εμφανής στους χρήστες, μόνο μετά από επιτυχημένη σύνδεσή τους στο σύστημα.
- **home.jsp:** Η σελίδα `home.jsp` είναι επιφορτισμένη με τη διαχείριση των χρηστών της πλατφόρμας. Έτσι, κατά τη φόρτωσή της γίνεται έλεγχος των παραμέτρων της εκάστοτε συνεδρίας. Αν βρεθεί κάποιο όνομα χρήστη αποθηκευμένο στη συνεδρία, τότε εμφανίζεται πληροφοριακό μήνυμα κατάστασης εισόδου. Στο μήνυμα αυτό, ο χρήστης καλώς ορίζεται στην εφαρμογή και ταυτόχρονα γίνεται αναφορά στο επίπεδο δικαιωμάτων που ανήκει, αν δηλαδή αποτελεί απλό χρήστη, προχωρημένο χρήστη ή διαχειριστή. Επιπρόσθετα, παρέχεται σύνδεσμος για τη σελίδα `logout.jsp`, η οποία αναλαμβάνει να αποσυνδέσει το χρήστη από την πλατφόρμα, εάν και εφ'όσον εκείνος το επιθυμεί. Σε περίπτωση που δε βρεθεί κάποιος χρήστης, τότε παρέχεται σύνδεσμος προς τη σελίδα `index.jsp`, με τιμή της παραμέτρου συνεδρίας `login` ίση με ένα. Η τιμή της παραμέτρου αυτής, επιτάσσει στο `index.jsp` την εμφάνιση της φόρμας σύνδεσης και εγγραφής στο σύστημα με χρήση των servlet `LoginService` και `RegisterService` αντίστοιχα.
- **logout.jsp:** Κατά τη φόρτωση της σελίδας αυτής, διαγράφονται όλες οι προηγούμενες τιμές των παραμέτρων της συνεδρίας. Με τον τρόπο αυτό, επιτυγχάνεται η αποσύνδεση ενός χρήστη από το υποσύστημα εξυπηρετητή.
- **search.html:** Πρόκειται για απλή στατική ιστοσελίδα κώδικα HTML. Στη σελίδα αυτή, παρέχεται η φόρμα δεδομένων με τα απαραίτητα στοιχεία για τη διεξαγωγή αναζήτησης στη βάση δεδομένων, κάνοντας χρήση μόνο δεδομένων και όχι εικόνων. Αφού τα στοιχεία συμπληρωθούν και γίνει ο απαραίτητος έλεγχος εγκυρότητας με χρήση ενός σεναρίου Javascript, μεταφέρονται μέσω μίας HTTP αίτησης στη σελίδα `search.jsp`, η οποία αναλαμβάνει τη σύνταξη του ερωτήματος SQL, τη διεξαγωγή της αναζήτησης και την εμφάνιση των αποτελεσμάτων στο χρήστη.

- **search.jsp:** Η σελίδα `search.jsp` είναι επιφορτισμένη με τη διεξαγωγή αναζήτησης αγγελιών, σύμφωνα με τα δεδομένα που εισήγαγε ο χρήστης στη σελίδα `search.html`, καθώς και την εμφάνιση των αποτελεσμάτων αυτής στο χρήστη. Έτσι, κατά τη φόρτωσή της λαμβάνονται κι αποθηκεύονται προσωρινά τα δεδομένα που συμπληρώθηκαν στη φόρμα της `search.html`, μέσω των αντίστοιχων παραμέτρων της εκάστοτε συνεδρίας. Κατόπιν, δημιουργείται ένα SQL ερώτημα σύμφωνα με αυτά, το οποίο στη συνέχεια αποστέλλεται στη βάση δεδομένων. Κατόπιν, τα αποτελέσματα εμφανίζονται στο χρήστη. Πιο συγκεκριμένα, οι πληροφορίες που λαμβάνει ο χρήστης για τα αποτελέσματα της αναζήτησης αφορούν το ονοματεπώνυμο του κατόχου της οικίας κάθε αγγελίας, το μέγεθος, την ηλικία και τη διεύθυνση της οικίας. Παράλληλα, δίνεται η δυνατότητα προβολής μιας ολοκληρωμένης λίστας των χαρακτηριστικών κάθε οικίας, μέσω ενός συνδέσμου προς τη σελίδα `info.jsp`. Επίσης, ο χρήστης μπορεί να επιλέξει τυχόν αγγελίες που τον ενδιαφέρουν, και να λάβει πληροφορίες για τη γεωγραφική τους θέση, όπως αυτή απεικονίζεται σε ένθετο χάρτη. Τέλος, εμφανίζεται μία ενδεικτική φωτογραφία για κάθε αγγελία, εφόσον αυτή υπάρχει, και παρέχεται η δυνατότητα προβολής όλων των εικόνων που αντιστοιχούν σε αυτή, μέσω ενός συνδέσμου προς τη σελίδα `photos.jsp`.
- **photos.jsp:** Η σελίδα `photos.jsp` αναλαμβάνει την εμφάνιση όλων των φωτογραφιών, οι οποίες αντιστοιχούν σε μία αγγελία. Αυτό επιτυγχάνεται με την υλοποίηση ενός σεναρίου Javascript. Κατά τη φόρτωση της σελίδας, λαμβάνεται από την αίτηση το μονοπάτι, που αντιστοιχεί στον κατάλογο αρχείων του συστήματος εξυπηρετητή, όπου βρίσκονται αποθηκευμένες οι φωτογραφίες της εν λόγω αγγελίας. Κατόπιν, αναζητείται και καταγράφεται η πλήρης λίστα των ονομάτων των αρχείων εικόνων της αγγελίας, τα οποία βρίσκονται στο φάκελο που υποδεικνύει το μονοπάτι. Με είσοδο αυτή τη λίστα, επιτυγχάνεται η παρουσίαση των φωτογραφιών στο χρήστη, μέσω του σεναρίου Javascript που βρίσκεται στο αρχείο `jgalscroll.js`, με τον τρόπο που παρουσιάζεται στην εικόνα που ακολουθεί.
- **info.jsp:** Η σελίδα `info.jsp` αναλαμβάνει την εμφάνιση όλων των στοιχείων μιας αγγελίας, η οποία επιλέγεται από το χρήστη από τα αποτελέσματα μιας αναζήτησης. Επίσης, υλοποιεί το σύστημα συστάσεων του εξυπηρετητή, με

βάση τη μέθοδο που ήδη αναφέρθηκε σε προηγούμενη ενότητα. Κατά τη φόρτωση της σελίδας, λαμβάνεται ως παράμετρος της αίτησης ο μοναδικός κωδικός της αγγελίας. Κατόπιν, γίνεται σύνδεση με τη βάση δεδομένων και διενεργείται αναζήτηση όλων των δεδομένων της εν λόγω αγγελίας. Τα αποτελέσματα της παραπάνω αναζήτησης, εμφανίζονται δομημένα στο χρήστη. Παράλληλα με την αναζήτηση των δεδομένων, υλοποιείται και το σύστημα συστάσεων με χρήση μεθόδων της βοηθητικής κλάσης ImageMethods. Για το σκοπό αυτό επιλέγεται μία ενδεικτική φωτογραφία από το σύνολο των εικόνων της αγγελίας. Η εικόνα αυτή θα αποτελέσει το δείγμα προς σύγκριση με τις φωτογραφίες των υπόλοιπων αγγελιών. Η φθίνουσα σειρά προτίμησης για την επιλογή της εικόνας δείγματος είναι η εξής: καθιστικού, υπνοδωματίου, εξωτερικού χώρου, κουζίνας και μπάνιου. Στη συνέχεια διεξάγονται συγκρίσεις της εικόνας δείγματος με αντίστοιχες εικόνες των υπόλοιπων αγγελιών, που βρίσκονται αποθηκευμένες στο υποσύστημα της βάσης δεδομένων. Τα τέσσερα πρώτα αποτελέσματα αγγελιών, οι οποίες έχουν φωτογραφίες που πληρούν τις προϋποθέσεις ομοιότητας, παρουσιάζονται στο χρήστη, με τη μορφή τεσσάρων ενδεικτικών φωτογραφιών, μίας για κάθε αγγελία. Τέλος, δίνεται η δυνατότητα προβολής λεπτομερειών για κάθε μία από τις παραπάνω αγγελίες, μέσω του συνδέσμου προς τη σελίδα info.jsp, που δημιουργείται σε κάθε φωτογραφία που παρουσιάζεται από το σύστημα συστάσεων.

- **smartadd.html:** Πρόκειται για απλή στατική ιστοσελίδα κώδικα HTML. Στη σελίδα αυτή, παρέχεται η δυνατότητα επιλογής μίας εικόνας οικίας και η αποθήκευσή του στο σύστημα αρχείων του υποσυστήματος εξυπηρετητή. Στη συνέχεια, δίνεται η δυνατότητα επιλογής του τύπου δωματίου στον οποίο αντιστοιχεί η εικόνα, δηλαδή αν πρόκειται για καθιστικό, εξωτερικό χώρο, υπνοδωμάτιο, μπάνιο ή κουζίνα. Αφού τα στοιχεία συμπληρωθούν μεταφέρονται μέσω μίας HTTP αίτησης πολλών τμημάτων (multipart) στο servlet ImageUpload, με σκοπό την αποθήκευση της φωτογραφίας σε κατάλληλη θέση στο σύστημα αρχείων του εξυπηρετητή. Κατόπιν γίνεται ανακατεύθυνση στη σελίδα smartaddhouse.jsp.
- **smartaddhouse.jsp:** Η σελίδα smartaddhouse.jsp αναλαμβάνει τη δημιουργία μίας φόρμας εισαγωγής νέας αγγελίας στη βάση δεδομένων, η οποία έχει προ-

επιλεγμένες τις κατάλληλες τιμές στα πεδία Επιπλωμένο, Τύπος Πατώματος και Εξωτερικό Χρώμα Οικίας. Οι τιμές αυτές, επιλέγονται σύμφωνα με φωτογραφία που «ανέβασε» ο χρήστης στο σύστημα από τη σελίδα smartadd.html. Η επιλογή των κατάλληλων τιμών επιτυγχάνεται με χρήση των μεθόδων της βοηθητικής κλάσης ImageMethods, οι οποίες υλοποιούν τους αλγόριθμους που αναλύθηκαν στο τέταρτο κεφάλαιο. Κατά τη φόρτωση της σελίδας, λαμβάνεται από τις παραμέτρους της εκάστοτε συνεδρίας, η τιμή που αντιστοιχεί στο μονοπάτι του συστήματος αρχείων, στο οποίο βρίσκεται αποθηκευμένη η εικόνα-δείγμα. Με βάση το μονοπάτι αυτό, λαμβάνεται η εικόνα, στην οποία εφαρμόζονται οι μέθοδοι isFurnished, floorType και extColor για τη διαπίστωση αν ο απεικονιζόμενος χώρος είναι επιπλωμένος, για την εύρεση του τύπου πατώματος και του εξωτερικού χρώματος της οικίας αντίστοιχα. Με το πέρας της συμπλήρωσης και των υπόλοιπων πεδίων της φόρμας, καθώς και τη διόρθωση τυχόν λαθών στα αυτόματα συμπληρωμένα πεδία, τα δεδομένα της φόρμας αποστέλλονται στο servlet SmartAddHouseService, από όπου και εισάγονται στη βάση δεδομένων.

- **smartsearch.html:** Πρόκειται για απλή στατική ιστοσελίδα κώδικα HTML. Στη σελίδα αυτή, παρέχεται η δυνατότητα επιλογής μίας εικόνας οικίας και η χρήση της για αναζήτηση αγγελιών, των οποίων οι φωτογραφίες παρυσιάζουν ομοιότητες με αυτή. Παράλληλα, μέσω μίας φόρμας εισαγωγής στοιχείων είναι δυνατή η παραμετροποίηση της αναζήτησης, με βάση συγκεκριμένα δεδομένα. Τα δεδομένα αυτά, αφορούν τα χαρακτηριστικά που οφείλουν να πληρούν οι αγγελίες, ούτως ώστε να συγκριθούν οι φωτογραφίες τους με το δείγμα του χρήστη. Αφού όλα τα στοιχεία συμπληρωθούν, μεταφέρονται μέσω μίας HTTP αίτησης πολλών τμημάτων (multipart) στη σελίδα smartsearch.jsp, με σκοπό τη διεξαγωγή της αναζήτησης και την εμφάνιση των αποτελεσμάτων στο χρήστη.
- **smartsearch.jsp:** Η σελίδα smartsearch.jsp είναι επιφορτισμένη με τη διεξαγωγή αναζήτησης αγγελιών, σύμφωνα με τα δεδομένα και τη φωτογραφία-δείγμα που εισήγαγε ο χρήστης στη σελίδα smartsearch.html, καθώς και την εμφάνιση των αποτελεσμάτων αυτής στο χρήστη. Έτσι, κατά τη φόρτωσή της λαμβάνονται κι αποθηκεύονται προσωρινά τα δεδομένα που συμπληρώθηκαν στη φόρμα της smartsearch.html, μέσω των αντίστοιχων

παραμέτρων της εκάστοτε αίτησης. Κατόπιν, δημιουργείται ένα SQL ερώτημα σύμφωνα με αυτά, το οποίο στη συνέχεια αποστέλλεται στη βάση δεδομένων. Από τα αποτελέσματα αυτά επιλέγονται εκείνα τα οποία περιέχουν φωτογραφίες προς σύγκριση με το δείγμα. Οι φωτογραφίες των επιλεγμένων αγγελιών συγκρίνονται ως προς την ομοιότητά τους με το δείγμα, με χρήση των μεθόδων της κλάσης ImageMethods. Από τα αποτελέσματα της σύγκρισης, επιλέγονται οι αγγελίες με όμοιες με το δείγμα εικόνες και εμφανίζονται ενδεικτικά δεδομένα τους στο χρήστη. Πιο συγκεκριμένα, οι πληροφορίες που λαμβάνει ο χρήστης για τα αποτελέσματα της αναζήτησης αφορούν το ονοματεπώνυμο του κατόχου της οικίας κάθε αγγελίας, το μέγεθος, την ηλικία και τη διεύθυνση της οικίας. Παράλληλα, δίνεται η δυνατότητα προβολής μιας ολοκληρωμένης λίστας των χαρακτηριστικών κάθε οικίας, μέσω ενός συνδέσμου προς τη σελίδα info.jsp. Επίσης, ο χρήστης μπορεί να επιλέξει τυχόν αγγελίες που τον ενδιαφέρουν, και να λάβει πληροφορίες για τη γεωγραφική τους θέση, όπως αυτή απεικονίζεται σε ένθετο χάρτη. Τέλος, εμφανίζεται μία ενδεικτική φωτογραφία για κάθε αγγελία, εφ'όσον αυτή υπάρχει, και παρέχεται η δυνατότητα προβολής όλων των εικόνων που αντιστοιχούν σε αυτή, μέσω ενός συνδέσμου προς τη σελίδα photos.jsp.

Real Estate Server

- Αρχική
- Προσθήκη νέας αγγελίας
- Έξυπνη προσθήκη νέας αγγελίας
- Έξυπνη αναζήτηση αγγελίας
- Αναζήτηση αγγελίας
- Επικοινωνία
- About

Εύρεση Οικίας:

Ενοικίαση Αγορά

Επιλογή πηγής

Ελάχιστη: Ελάχιστη Διαθέσιμη ▾
Μέγιστη: Μέγιστη Διαθέσιμη ▾

Επιλογή τετραγωνικών

Ελάχιστα: Ελάχιστα Διαθέσιμα ▾
Μέγιστα: Μέγιστα Διαθέσιμα ▾

Επιλογή ηλικίας σπιτιού

Ελάχιστη: Νέο ▾
Μέγιστη: Μέγιστη Διαθέσιμη ▾

Επιλογή Τοποθεσίας

Νομός: Νομός ▾
Περιοχή Νομού: Περιοχή ▾
Πόλη: Πόλη ▾
Δήμος: Δήμος ▾

Εικόνα 4.2.2.1 Στιγμιότυπο από τη χρήση της ιστοσελίδας

4.2.3. Τμήμα Servlet Εξυπηρετητή

Το τμήμα αυτό αποτελείται από ένα μεγάλο αριθμό servlet, που σκοπό έχουν τη διεκπεραίωση των αιτήσεων που αποστέλλονται, τόσο από το τμήμα διεπαφής του εξυπηρετητή, όσο και από το υποσύστημα πελάτη. Οι αιτήσεις αυτές περιλαμβάνουν κυρίως την αποστολή ερωτημάτων (queries) SQL στη βάση δεδομένων για εισαγωγή νέων στοιχείων σε αυτή ή τη διεξαγωγή αναζήτησης σε κάποιον από τους πίνακές της. Το τμήμα αυτό συμπληρώνεται από έναν αριθμό βοηθητικών κλάσεων, οι οποίες χρησιμοποιούνται κυρίως από τα Servlet, αλλά και σε μικρότερο βαθμό απευθείας από τις σελίδες jsp του τμήματος διεπαφής. Σκοπός των βοηθητικών αυτών κλάσεων, είναι η εκτέλεση εξειδικευμένων λειτουργιών, όπως για παράδειγμα ψηφιακή επεξεργασία εικόνων της βάσης ή κωδικοποίηση και αποκωδικοποίηση με βάση τη μέθοδο Base64.

Η επικοινωνία μεταξύ των σελίδων jsp καθώς και του υποσυστήματος εξυπηρετητή με τα servlet γίνεται με χρήση αιτήσεων αποστολής δεδομένων. Υπάρχουν δυο είδη αιτήσεων αποστολής δεδομένων μέσω του URL : η GET και η POST.

Μια **αίτηση GET** τοποθετεί όλα τα προς μεταφορά δεδομένα στο τέλος του URL, το οποίο έχει σε αυτή την περίπτωση την παρακάτω μορφή:

```
http://<host>:<port><request_path>?<par_1=value_1>&...&<par_n=value_n>
```

,όπου:

- **host:** η διεύθυνση IP του Web Server
- **port:** η θύρα στην οποία «ακούει» ο Web server
- **request_path:** το μονοπάτι όπου βρίσκεται το servlet μέσα στο σύστημα εξυπηρετητή
- **par_n:** το όνομα της παραμέτρου n
- **value_n:** η τιμή της παραμέτρου n

Μια **αίτηση POST** περιλαμβάνει τα δεδομένα προς αποστολή σε μία επικεφαλίδα, η οποία στέλνεται ξεχωριστά από το URL, μέσω ενός διαφορετικού stream εξόδου.

Τα Java servlet χειρίζονται τα δυο αυτά είδη αιτήσεων μέσω μεθόδων που κληρονομούν από την κλάση `HttpServlet`. Οι μέθοδοι αυτές είναι οι: `doGet(HttpServletRequest, HttpServletResponse)` και `doPost(HttpServletRequest, HttpServletResponse)`. Οι μέθοδοι αυτές έχουν δυο ορίσματα. Το πρώτο από αυτά είναι ένα αντικείμενο `HttpServletRequest`, ενώ το δεύτερο είναι αντικείμενο `HttpServletResponse`.

Κατά τη μεταφορά δεδομένων σε ένα servlet, αυτά αποθηκεύονται με τη μορφή string από την κλάση `HttpServletRequest`. Τα δεδομένα που εστάλησαν είναι δυνατό να ληφθούν ξανά από το servlet με κλήση της μεθόδου `getParameter(String name)`, όπου το όρισμα `name` αντιστοιχεί στο όνομα της παραμέτρου. Σε περίπτωση έλλειψης παραμέτρου με όνομα `name`, η μέθοδος αυτή επιστρέφει `null`.

Ένα servlet επικοινωνεί με το σύστημα που του έστειλε κάποια αίτηση κάνοντας χρήση των μεθόδων της κλάσης `HttpServletResponse`. Αρχικά, ορίζεται το είδος του περιεχομένου που θα στείλει το servlet σαν απόκριση με κλήση της μεθόδου `setContentType(String type)`, όπου `type` ο τύπος του περιεχομένου. Στην συνέχεια, δημιουργείται ένα stream εξόδου servlet που σχετίζεται με το πρόγραμμα που κάλεσε το servlet. Τέλος, για την αποστολή των δεδομένων καλείται η μέθοδος `println(String)`, σε αυτό το stream. Το stream εξόδου servlet παριστάνεται από την κλάση `ServerOutputStream`, που είναι μέρος του πακέτου `javax.servlet`.

Αναλυτικά τα Servlet του υποσυστήματος εξυπηρετητή είναι τα παρακάτω:

- **RegisterService:** Σκοπός του εν λόγω Servlet είναι η διεξαγωγή αναζήτησης στη βάση δεδομένων, ούτως ώστε να ελεγχθεί η διαθεσιμότητα ενός ονόματος χρήστη προς εγγραφή, εφ'όσον κάθε όνομα χρήστη στη βάση οφείλει να είναι μοναδικό. Επίσης, φροντίζει να καταχωρήσει το όνομα μαζί με τα υπόλοιπα στοιχεία του χρήστη στη βάση δεδομένων, αν ο έλεγχος είναι επιτυχής. Κατά τη φόρτωσή του, το Servlet λαμβάνει το πιθανό όνομα χρήστη, όπως επίσης και τα υπόλοιπα δεδομένα του νέου χρήστη, από τις παραμέτρους της αίτησης

που του γίνεται. Στη συνέχεια, διενεργεί αναζήτηση του ονόματος στη βάση δεδομένων. Σε περίπτωση που το όνομα βρεθεί στη βάση, ανακατευθύνει στη σελίδα `index.jsp`, με κατάλληλες τιμές παραμέτρων ώστε να εμφανιστεί ανάλογο μήνυμα λάθους. Αν η αναζήτηση ολοκληρωθεί χωρίς εύρεση του ονόματος, τότε εισάγει στον πίνακα χρηστών `users` του υποσυστήματος βάσης, όλα τα στοιχεία του νέου χρήστη. Κατόπιν, προσθέτει τα στοιχεία αυτά στις παραμέτρους της συνεδρίας, εισάγοντας έτσι το χρήστη στο σύστημα, και ανακατευθύνει στη σελίδα `index.jsp`.

- **LoginService:** Σκοπός του εν λόγω Servlet είναι ο έλεγχος της δυνατότητας εισαγωγής ενός χρήστη στο σύστημα, με βάση το όνομα χρήστη και τον κωδικό που έχει εισάγει. Αυτό επιτυγχάνεται, με έλεγχο της ύπαρξης του εν λόγω ονόματος και κωδικού χρήστη στη βάση δεδομένων. Επομένως, κατά τη φόρτωσή του, το Servlet λαμβάνει το όνομα χρήστη και τον κώδικο πρόσβασης που ο χρήστης έχει εισάγει, από τις παραμέτρους της αίτησης που του γίνεται. Στη συνέχεια, διενεργεί αναζήτηση ύπαρξης αυτών στη βάση δεδομένων. Σε περίπτωση που δε βρεθούν στη βάση, ανακατευθύνει στη σελίδα `index.jsp`, με κατάλληλες τιμές παραμέτρων ώστε να εμφανιστεί ανάλογο μήνυμα λάθους. Αν υπάρξει ταυτοποίηση με τα αποθηκευμένα στοιχεία της βάσης, προσθέτει τα στοιχεία που έλαβε από την αίτηση στις αντίστοιχες παραμέτρους της συνεδρίας και ανακατευθύνει στη σελίδα `index.jsp`.
- **AddHouseService:** Σκοπός του εν λόγω Servlet είναι η εισαγωγή μίας νέας αγγελίας χωρίς εικόνες στη βάση δεδομένων. Η διαδικασία που ακολουθείται και εδώ είναι όμοια με τις προηγούμενες περιπτώσεις. Το Servlet λαμβάνει τα στοιχεία της αγγελίας, τα οποία πρέπει να εισάγει στη βάση, από τις παραμέτρους της αίτησης που του γίνεται. Στη συνέχεια, εισάγει τα εν λόγω δεδομένα στη βάση. Σε περίπτωση αποτυχίας κατά τη διάρκεια εισαγωγής των δεδομένων, ανακατευθύνει στη σελίδα `index.jsp`, με κατάλληλες τιμές παραμέτρων ώστε να εμφανιστεί ανάλογο μήνυμα λάθους. Σε αντίθετη περίπτωση, ανακατευθύνει στη σελίδα `index.jsp`, με τιμές παραμέτρων τέτοιες, ούτως ώστε να εμφανιστεί μήνυμα επιτυχούς εισαγωγής της αγγελίας.
- **SmartAddHouseService:** Η λειτουργία του εν λόγω Servlet είναι καθ' όλα όμοια με του `AddHouseService`. Η διαφορά τους έγκειται στο γεγονός ότι εδώ

είναι δυνατός ο χειρισμός εισαγωγής αγγελιών με φωτογραφίες. Έτσι, τα δεδομένα της αγγελίας λαμβάνονται μέσω των αντίστοιχων τιμών παραμέτρων της αίτησης και εισάγονται στη βάση. Παράλληλα, υπολογίζεται από αυτά το μονοπάτι, στο οποίο θα αποθηκευτεί η φωτογραφία της αγγελίας. Με την ολοκλήρωση του υπολογισμού του μονοπατιού, η εν λόγω εικόνα μετακινείται από την προσωρινή της θέση, όπου έχει τοποθετηθεί από το Servlet UploadImage, στην τελική της θέση, ανάλογη της αγγελίας που αντιστοιχεί. Σε περίπτωση, είτε αποτυχίας εισαγωγής της αγγελίας, είτε αποτυχίας αποθήκευσης της εικόνας στην τελική της θέση, το Servlet ανακατευθύνει στη σελίδα index.jsp, με κατάλληλες τιμές παραμέτρων ώστε να εμφανιστεί μήνυμα λάθους αντίστοιχο της κάθε περίπτωσης. Τέλος, το ίδιο συμβαίνει και στην περίπτωση επιτυχημένης ολοκλήρωσης της παραπάνω διαδικασίας, με την εμφάνιση αντίστοιχου πληροφοριακού μηνύματος στη σελίδα index.jsp.

- **UploadImage:** Το Servlet αυτό έχει σκοπό την προσωρινή αποθήκευση μίας εικόνας, την οποία επιθυμεί κάποιος χρήστης να εισάγει στην πλατφόρμα μαζί με αντίστοιχη αγγελία. Για την υλοποίηση του παραπάνω χαρακτηριστικού, το servlet δέχεται μία αίτηση δύο τμημάτων (multipart request). Το πρώτο τμήμα αυτής, περιέχει το αρχείο της εικόνας προς αποθήκευση. Το δεύτερο τμήμα περιέχει πληροφορίες, σχετικά με τον τύπο οικίας που απεικονίζεται, αν δηλαδή είναι φωτογραφία εξωτερικής όψης, υπνοδωματίου, καθιστικού, μπάνιου ή κουζίνας. Το όνομα του αρχείου της εικόνας που θα αποθηκευτεί χωρίζεται σε δύο μέρη. Ανάλογα με τον τύπο απεικόνισης, το πρώτο από τα δύο μέρη του ονόματος της εικόνας λαμβάνει διαφορετική τιμή. Το δεύτερο μέρος του ονόματος, αποτελείται από 10 τυχαία δημιουργημένα ψηφία, μοναδικά για τον κατάλογο αρχείων, στον οποίο αποθηκεύεται η εικόνα. Το μονοπάτι αρχείων, που δείχνει τη θέση αποθήκευσης της φωτογραφίας, είναι προ-ρυθμισμένο και αντιστοιχεί στην τιμή της global μεταβλητής path του UploadImage Servlet, καθιστώντας δυνατή την εύκολη αλλαγή του.

Σε πλήρη αντιστοιχία με τα προαναφερθέντα Servlet LoginService, RegisterService, AddHouseService και UploadImage βρίσκονται τα Servlet checkLogin, checkRegistering, insertServlet και imageReceiver. Οι ενέργειες που επιτελούν, είναι όμοιες με τα αντίστοιχα Servlet που αναλύθηκαν παραπάνω. Διαφέρουν μόνο στο

γεγονός ότι δέχονται αιτήσεις από το υποσύστημα του πελάτη, τροποποιώντας ανάλογα και τις αποκρίσεις (response) τους.

Επίσης, λόγω της έλλειψης δυνατότητας αποστολής του αρχείου εικόνας με αίτηση πολλών τμημάτων, που πηγάζει από τους περιορισμούς της Java ME, η αποστολή της εικόνας από το υποσύστημα πελάτη προς το `imageReceiver` γίνεται σε μορφή κωδικοποιημένου με Base64 String. Συνεπώς, πριν την επιλογή του μονοπατιού και ονόματος αποθήκευσης, το αρχείο της εικόνας αποκωδικοποιείται στο `imageReceiver`. Αυτό επιτυγχάνεται με χρήση της μεθόδου αποκωδικοποίησης με τη μέθοδο `Base64`, η οποία υλοποιείται σε αντίστοιχη μέθοδο στη βοηθητική κλάση `Base64`.

Τέλος, αξίζει να σημειωθεί ότι τόσο στην αποστολή αιτήσεων και αποκρίσεων μέσω HTTP πρωτοκόλλου, όσο και στην εισαγωγή δεδομένων ή εξαγωγή στοιχείων από τη βάση δεδομένων ακολουθήθηκε το στάνταρ κωδικοποίησης χαρακτήρων UTF-8. Η κωδικοποίηση αυτή, επιτρέπει την επεξεργασία από το σύστημα, μη λατινογενών χαρακτήρων και γλωσσών, προσφέροντας έτσι απολύτη συμβατότητα και με χαρακτήρες του ελληνικού αλφάβητου.

4.2.4. Βοηθητική Κλάση `ImageMethods`

Η κλάση αυτή περιλαμβάνει όλες τις μεθόδους, οι οποίες είναι υπεύθυνες για τη επεξεργασία και σύγκριση των εικόνων της βάσης. Ουσιαστικά πρόκειται για τις μεθόδους, που υλοποιούν αλγόριθμους επεξεργασίας εικόνας που περιγράφηκαν σε προηγούμενο κεφάλαιο, οι οποίες αποτελούν και τον σχεδιαστικό πυρήνα του συστήματος συστάσεων της εφαρμογής.

Οι μέθοδοι της κλάσης αυτής είναι δεκατέσσερις, από τις οποίες πέντε είναι βασικές, δηλαδή εκτελούν τις κύριες λειτουργίες και οι υπόλοιπες είναι βοηθητικές σε αυτές. Η πρώτη κύρια μέθοδος, `iterate(String rootpath,String samplepath,String limitation,String pathlimit)`, είναι υπεύθυνη για τη σύγκριση μίας εικόνας-δείγμα με όλες τις εικόνες, οι οποίες βρίσκονται αποθηκευμένες στη βάση δεδομένων. Δεύτερη κύρια μέθοδος είναι η `checkImages(int[][] ch1,File file2,CannyEdgeDetector`

det,BufferedImage scr2), η οποία και συγκρίνει δύο εικόνες ως προς την ομοιότητά τους, με βάση τη μέθοδο των συζευγμένων ιστογραμμάτων. Ακολουθούν οι τρεις τελευταίες κύριες μέθοδοι. Κάθε μία από αυτές αναλαμβάνει να παράξει αποτέλεσμα, το οποίο θα οδηγεί στην αυτόματη συμπλήρωση δεδομένων της φόρμας, στην αντίστοιχη ιστοσελίδα του τμήματος της διεπαφής. Συνεπώς, η μέθοδος isFurnished(String path) αναλαμβάνει τον εντοπισμό εικόνων, οι οποίες παρουσιάζουν επιπλωμένους χώρους. Η μέθοδος floorType(String path) αποφαινεται για το αν ο απεικονιζόμενος στη φωτογραφία χώρος, έχει πάτωμα από πλακάκια, παρκέ ή μωσαϊκό. Τέλος, η μέθοδος extColor(String path) εντοπίζει το κυρίαρχο χρώμα της εξωτερικής όψης μιας οικίας, όπως αυτή απεικονίζεται σε μία φωτογραφία. Αναλυτικά:

- **iterate(String rootpath,String samplepath,String limitation,String pathlimit):** Η μέθοδος αυτή δέχεται ως ορίσματα τέσσερις παραμέτρους. Η πρώτη αντιστοιχεί μονοπάτι, όπου βρίσκονται αποθηκευμένες όλες οι φωτογραφίες της πλατφόρμας. Η δεύτερη, ορίζει το μονοπάτι, το οποίο αντιστοιχεί στο αρχείο της εικόνας-δείγμα, η οποία και θα συγκριθεί με όλες τις υπόλοιπες. Η τρίτη παράμετρος αντιπροσωπεύει τον περιορισμό στο περιεχόμενο των φωτογραφιών, που θα συγκριθούν με το δείγμα. Ο περιορισμός αυτός σκοπό έχει να συγκρίνονται όμοιου περιεχομένου εικόνες, για παράδειγμα φωτογραφίες καθιστικών έναντι εικόνων καθιστικών και μόνο, αυξάνοντας την απόδοση της όλης μεθόδου. Η ιδιότητα αυτή των εικόνων λαμβάνεται από τους αρχικούς χαρακτήρες, οι οποίοι αποτελούν το όνομά τους. Τέλος, η τέταρτη παράμετρος ορίζει κάποιο μονοπάτι, που αντιστοιχεί στον κατάλογο με τα αρχεία των εικόνων, οι οποίες αναφέρονται στην ίδια αγγελία με την εικόνα δείγμα. Σκοπός της παραμέτρου αυτής, είναι η επίγνωση από το σύστημα του φακέλου αρχείων, στον οποίο πρέπει να αποφευχθούν οι συγκρίσεις εικόνων. Αυτό συμβαίνει διότι συγκρίσεις μεταξύ εικόνων ίδιας αγγελίας δεν αποφέρουν αποτελέσματα με ενδιαφέρον για τη συγκεκριμένη υλοποίηση. Με την κλήση της εν λόγω μεθόδου, υπολογίζεται το συζευγμένο ιστόγραμμα της εικόνας δείγματος με χρήση μεθόδων της βοηθητικής κλάσης CannyEdgeDetector. Στη συνέχεια, διατρέχεται ο φάκελος με τα αρχεία εικόνων, που αντιστοιχεί στην παράμετρο rootpath, και σε περίπτωση που αυτά ικανοποιούν τα επιβαλλόμενα από τις παραμέτρους

limitation και pathlimit κριτήρια, συγκρίνονται με την εικόνα που αντιστοιχεί στο μονοπάτι samplepath μέσω της μεθόδου checkImages(int[][] ch1,File file2,CannyEdgeDetector det,BufferedImage scr2), η οποία αναλύεται στη συνέχεια. Τελικά, η μέθοδος επιστρέφει ένα πίνακα από Strings, όπου περιέχονται όλα τα μονοπάτια των αρχείων εικόνων, οι οποίες είναι όμοιες με το δείγμα.

- **checkImages(int[][] ch1,File file2,CannyEdgeDetector det,BufferedImage scr2):** Η μέθοδος αυτή δέχεται τέσσερα ορίσματα και υλοποιεί τη σύγκριση δύο εικόνων με βάση τη μέθοδο των συζευγμένων ιστογραμμάτων. Έτσι, ως πρώτη είσοδος σε αυτή παρέχεται το συζευγμένο ιστόγραμμα της εικόνας, η οποία αποτελεί και το δείγμα, όταν η εν λόγω μέθοδος καλείται από την iterate. Επίσης δέχεται ως είσοδο τα αντικείμενα file2 και scr2 των κλάσεων File και BufferedImage, όπως αυτά προκύπτουν για το αρχείο της δεύτερης εικόνας προς σύγκριση. Τελικό όρισμα της μεθόδου αποτελεί μία υπόσταση της βοηθητικής κλάσης CannyEdgeDetector. Από τα παραπάνω ορίσματα, η μέθοδος κάνει χρήση των μεθόδων της βοηθητικής κλάσης CannyEdgeDetector, ούτως ώστε να υπολογίσει το συζευγμένο ιστόγραμμα της δεύτερης εικόνας προς σύγκριση. Τελικά, για την ομοιότητα των εικόνων αποφαιίνεται με βάση την έξοδο της μεθόδου comparePics, την οποία και χρησιμοποιεί ως όρισμα στη μέθοδο getResult. Το αποτέλεσμα της τελευταίας αυτής μεθόδου, οδηγεί την checkImages στην επιστροφή true ή false.
- **comparePics(int[][] a,int[][] b,CannyEdgeDetector det):** Η μέθοδος αυτή δέχεται τρία ορίσματα, από τα οποία τα δύο πρώτα αντιστοιχούν στα συζευγμένα ιστογράμματα δύο εικόνων ενώ το τρίτο σε ένα αντικείμενο της κλάσης CannyEdgeDetector. Κατά την κλήση της, υπολογίζει τη μετρική L1 μεταξύ των αντίστοιχων τιμών των δύο πινάκων. Κατόπιν, λαμβάνεται ο μέσος όρος των τιμών των αποστάσεων L1 που προέκυψαν, για κάθε μία από τις τέσσερις συνιστώσες του συζευγμένου ιστογράμματος. Ακολουθεί αναγωγή των τιμών αυτών σε κλίμακα 0-100 με χρήση της μεθόδου scalearray. Ο πίνακας, ο οποίος επιστρέφεται από τη μέθοδο αυτή, έχοντας ανηγμένες τις τιμές του από 0 έως 100, αποτελεί και την επιστροφή της μεθόδου.

- **getResults(double[] a):** Η μέθοδος αυτή δέχεται ένα και μόνο όρισμα, το οποίο αντιστοιχεί στον ανηγμένο σε κλίμακα 0-100 πίνακα των μέσων τιμών των μετρικών L1, όπως αυτές προκύπτουν από τις συνιστώσες των συζευγμένων ιστογραμμάτων των εικόνων προς σύγκριση. Από τις τέσσερις αυτές τιμές, υπολογίζεται ο γενικός μέσος όρος. Στη συνέχεια, η τιμή αυτή συγκρίνεται με μία τιμή κατωφλίου. Σε περίπτωση που είναι μεγαλύτερη ή ίση της τιμής κατωφλίου, οι εικόνες θεωρούνται μεταξύ τους όμοιες και η μέθοδος επιστρέφει true. Σε αντίθετη περίπτωση, επιστρέφεται false.
- **Quantize(String path):** Η μέθοδος αυτή δέχεται ένα όρισμα, το οποίο είναι το μονοπάτι της εικόνας που θα υποστεί κβαντοποίηση χρωμάτων. Μέσω της πληροφορίας του μονοπατιού, εντοπίζεται το αρχείο της εικόνας και τα δεδομένα αυτού συγκεντρώνονται σε ένα δισδιάστατο πίνακα, με τιμές τις τιμές των εικονοστοιχείων της εικόνας στις αντίστοιχες συντεταγμένες. Ο πίνακας αυτός, χρησιμοποιείται ως όρισμα στη μέθοδο `quantizeImage` της βοηθητικής κλάσης `Quantize`, η οποία εκτελεί την επιθυμητή ενέργεια. Η έξοδος της μεθόδου `quantizeImage`, μετατρέπεται σε `BufferedImage` αντικείμενο, το οποίο αποτελεί και το επιστρεφόμενο από τη μέθοδο `quantize` αντικείμενο.
- **isFurnished(String path):** Η μέθοδος αυτή δέχεται ένα όρισμα, το οποίο είναι το μονοπάτι της εικόνας, η οποία θα ελεγχθεί για το αν απεικονίζει επιπλωμένο χώρο. Μέσω της πληροφορίας του μονοπατιού, εντοπίζεται το αρχείο της εικόνας. Αρχικά, εκτελείται στο αρχείο αυτό ο αλγόριθμος κατάτμησής της σε περιοχές, όπως αυτός περιγράφηκε σε προηγούμενη ενότητα, οπότε και υπολογίζεται ο αριθμός των περιοχών στις οποίες χωρίζεται. Για το σκοπό αυτό γίνεται χρήση μεθόδων της βοηθητικής κλάσης `SimpleRegionGrowing`. Ο αριθμός αυτός, συγκρίνεται με μία τιμή κατωφλίου. Αν είναι μεγαλύτερος αυτής, τότε η μέθοδος επιστρέφει false. Σε διαφορετική περίπτωση, εκτελείται στην εικόνα ανίχνευση ακμών με τον αλγόριθμο του `Canny`. Κάτι τέτοιο επιτυγχάνεται με χρήση της αντίστοιχης μεθόδου της κλάσης `CannyEdgeDetector`. Μετά το επιτυχές πέρας της ανίχνευσης ακμών, υπολογίζεται το ποσοστό των εικονοστοιχείων που αποτελούν ακμές έναντι του συνόλου των εικονοστοιχείων της εικόνας. Κατόπιν, έχουμε σύγκριση της υπολογισμένης τιμής με μία νέα τιμή κατωφλίου. Σε περίπτωση που το

ποσοστό ακμών είναι μεγαλύτερο αυτής, η μέθοδος επιστρέφει false. Αν είναι μικρότερο, τότε ο απεικονιζόμενος χώρος είναι επιπλωμένος και η μέθοδος επιστρέφει την τιμή true.

- **floorType(String path):** Η μέθοδος αυτή δέχεται ένα όρισμα, το οποίο είναι το μονοπάτι ενός αρχείου εικόνας. Η εικόνα αυτή, θα ελεγχθεί σχετικά με τον τύπο του πατώματος του δωματίου που απεικονίζει. Αρχικά, μέσω της πληροφορίας του μονοπατιού, εντοπίζεται το αρχείο της εικόνας. Στη συνέχεια, εκτελείται στο νότιο τμήμα της εικόνας, το οποίο καταλαμβάνει το 1/3 της συνολικής της επιφάνειας, ο αλγόριθμος εύρεσης κυρίαρχου χρώματος, όπως αυτός περιγράφηκε σε προηγούμενη ενότητα, οπότε και υπολογίζεται το κυρίαρχο χρώμα. Για το σκοπό αυτό γίνεται χρήση της μεθόδου colorFinder. Ανάλογα με το κυρίαρχο χρώμα, η μέθοδος αποφαινεται και επιστρέφει τον τύπο του πατώματος.
- **extColor(String path):** Η μέθοδος αυτή είναι σε μεγάλο βαθμό όμοια στη λειτουργία της με τη μέθοδο floorType. Έτσι, δέχεται και αυτή ένα όρισμα, το οποίο είναι το μονοπάτι ενός αρχείου εικόνας. Αυτή τη φορά όμως, η εικόνα ελέγχεται σχετικά με το χρώμα της εξωτερική όψης της οικίας που απεικονίζει. Επίσης, μία ακόμα διαφορά τους έγκειται στο γεγονός ότι ο αλγόριθμος εύρεσης κυρίαρχου χρώματος εκτελείται στο μεσαίο τμήμα της εικόνας, το οποίο καταλαμβάνει τα 3/5 της συνολικής της επιφάνειας. Επιπλέον, η λίστα των χρωμάτων που χρησιμοποιούνται από τη μέθοδο colorFinder είναι εκτενέστερη στην περίπτωση κλήσης της από την extColor. Τελικά, το κυρίαρχο χρώμα, όπως αυτό επιστρέφεται από την κλήση της μεθόδου colorFinder, είναι και η επιστρεφόμενη τιμή της extColor.
- **colorFinder(String path, int mode):** Η μέθοδος αυτή, υλοποιεί τον αλγόριθμο εύρεσης κυρίαρχου χρώματος εικόνας, όπως αυτός παρουσιάστηκε σε προηγούμενη ενότητα. Τα ορίσματά της είναι δύο. Το πρώτο από αυτά, αντιστοιχεί στο μονοπάτι του αρχείου της προς εξέταση εικόνας. Η δεύτερη παράμετρος, καθορίζει την περιοχή στην οποία θα γίνει ο έλεγχος, καθώς επίσης και τη λίστα των χρωμάτων που θα χρησιμοποιηθεί για τη σύγκριση με το χρώμα κάθε εικονοστοιχείου. Αρχικά, από την τιμή path των παραμέτρων της, εντοπίζεται το αρχείο της εικόνας, από όπου λαμβάνονται οι rgb τιμές των εικονοστοιχείων της, από τη μέθοδο getColorHistogram της βοηθητικής

κλάσης CannyEdgeDetector. Στη συνέχεια, με χρήση της μεθόδου rgb2hsv μετατρέπονται οι εν λόγω τιμές, στις αντίστοιχές τους με βάση το χρωματικό μοντέλο HSV. Η μέθοδος αυτή, λαμβάνει μία τριάδα τιμών rgb και, με βάση τον τύπο μετατροπής των τιμών rgb σε hsv, υπολογίζει και επιστρέφει τις αντίστοιχες τιμές hsv. Με βάση τις τιμές αυτές και χρήση της μετρικής L1, υπολογίζεται από την colorFinder το χρώμα, το οποίο προσεγγίζει καλύτερα το χρώμα κάθε εικονοστοιχείου της περιοχής της εικόνας που καθορίζει η τιμή της παραμέτρου mode. Ακολουθεί ένας τυπικός υπολογισμός του χρώματος που προσεγγίζεται από τα περισσότερα εικονοστοιχεία. Το χρώμα αυτό, αποτελεί το κυρίαρχο χρώμα καθώς επίσης και την επιστρεφόμενη τιμή της μεθόδου.

4.2.5. Βοηθητική Κλάση CannyEdgeDetector

Η συγκεκριμένη κλάση αποτελεί τη δεύτερη βοηθητική κλάση του υποσυστήματος εξυπηρετητή, το οποίο υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας. Σκοπός της κλάσης αυτής δεν είναι μόνο η παροχή της μεθόδου που υλοποιεί τον αλγόριθμο ανίχνευσης ακμών του Canny. Επιπρόσθετα, παρέχει μεθόδους για την εξαγωγή πλήθους άλλων χαρακτηριστικών μίας εικόνας. Έτσι, στην κλάση αυτή υλοποιούνται οι εξής μέθοδοι:

- **setSourceImage(BufferedImage image):** Η μέθοδος αυτή δέχεται ως όρισμα ένα στιγμιότυπο της κλάσης BufferedImage, η οποία αντιστοιχεί σε μία εικόνα. Την εικόνα αυτή την ορίζει ως το δείγμα, το οποίο θα χρησιμοποιείται από το εκάστοτε στιγμιότυπο για την εξαγωγή των ζητούμενων αποτελεσμάτων.
- **computeGM():** Η μέθοδος αυτή δε δέχεται κανένα όρισμα. Η λειτουργία της έγκειται στον υπολογισμό της βαθμίδας κλίσης κάθε εικονοστοιχείου της εικόνας δείγματος. Ο τρόπος υπολογισμού περιγράφηκε σε προηγούμενο κεφάλαιο. Οφείλει να χρησιμοποιείται μετά την κλήση είτε της μεθόδου process(), είτε της μεθόδου computeCH() της ίδιας κλάσης, ούτως ώστε να έχουν αρχικοποιηθεί οι global μεταβλητές της κλάσης, που αντιστοιχούν στο ιστόγραμμα χρώματος της εικόνας.

- **process():** Η μέθοδος αυτή δε δέχεται κανένα όρισμα. Αποτελεί την κύρια μέθοδο της κλάσης CannyEdgeDetector. Με την κλήση της, εντοπίζει την εικόνα δείγμα, υπολογίζει τις rgb τιμές των εικονοστοιχείων της, το ιστόγραμμα χρώματός της καθώς και τις ακμές της με βάση τον αλγόριθμο ανίχνευσης ακμών του Canny, όπως αυτός παρουσιάστηκε σε προηγούμενη ενότητα. Όλα τα παραπάνω αποτελέσματα, τα αποθηκεύει σε αντίστοιχες global μεταβλητές της κλάσης, δίνοντας έτσι τη δυνατότητα προσπέλασής τους μέσω getter μεθόδων οποιαδήποτε στιγμή.

Αξιοσημείωτο είναι το γεγονός ότι η ύπαρξη της μεθόδου process δεν αποτελεί μονόδρομο στον υπολογισμό του ιστογράμματος χρώματος και των rgb τιμών των εικονοστοιχείων της εικόνας-δείγμα. Για το σκοπό αυτό υπάρχουν και οι βοηθητικές μέθοδοι computeCH και readLuminance της εν λόγω κλάσης, οι οποίες υπολογίζουν και αρχικοποιούν τις global μεταβλητές colorhistogram και rgbvalues αντίστοιχα. Όταν όμως κρίνεται απαραίτητος ο υπολογισμός όλων των παραπάνω παραμέτρων, η μέθοδος process προσφέρει αποδοτικότερη χρήση της μνήμης και ταχύτερα αποτελέσματα, σε σχέση με το αν κάθε υπολογισμός εκτελούνταν μεμονωμένα από τις αντίστοιχες μεθόδους.

4.2.6. Βοηθητική Κλάση SimpleRegionGrowing

Η τρίτη βοηθητική κλάση του υποσυστήματος εξυπηρετητή, έχει ως σκοπό την υλοποίηση του αλγόριθμου κατάτμησης εικόνας σε περιοχές, όπως αυτός περιγράφηκε στο κεφάλαιο Μέθοδος. Για την υλοποίηση του παραπάνω αλγόριθμου, η κλάση SimpleRegionGrowing υλοποιεί δύο μεθόδους:

- **preprocess(PlanarImage input):** Η μέθοδος αυτή δέχεται ως όρισμα ένα στιγμιότυπο της κλάσης PlanarImage, το οποίο αντιστοιχεί σε μία εικόνα. Στη συνέχεια, ελέγχει αν η εικόνα αυτή είναι σε τόνους του γκρι. Αν ισχύει κάτι τέτοιο, τότε επιστρέφει την ίδια την εικόνα. Σε αντίθετη περίπτωση, διατρέχει κάθε ένα από τα εικονοστοιχεία της και μετατρέπει το χρώμα τους σε αντίστοιχες τιμές σε τόνους του γκριζου. Με το πέρας της μετατροπής όλων των εικονοστοιχείων, επιστρέφει τη νέα εικόνα.

- **runprocess(PlanarImage input):** Η μέθοδος αυτή δέχεται ως όρισμα ένα στιγμιότυπο της κλάσης PlanarImage, το οποίο αντιστοιχεί σε μία εικόνα με τιμές εικονοστοιχείων σε τόνους του γκρι,. Η εικόνα αυτή έχει προκύψει από το στάδιο προεπεξεργασίας, δηλαδή τη μέθοδο preprocess. Με δεδομένη την παραπάνω εικόνα, κάνει χρήση μίας στοίβας για τον υπολογισμό των περιοχών στις οποίες χωρίζεται αυτή. Πιο συγκεκριμένα τα βήματα της μεθόδου είναι τα εξής:

1. Εύρεση ενός εικονοστοιχείου, το οποίο δεν έχει ακόμα σημειωθεί. Μόλις αυτό βρεθεί, σημειώνεται και εισάγονται οι συντεταγμένες του στη στοίβα.
2. Όσο υπάρχουν εικονοστοιχεία στη στοίβα:
 - a. Λαμβάνεται ένα εικονοστοιχείο από τη στοίβα
 - b. Ελέγχονται τα γειτονικά σε αυτό σημεία για το αν έχουν σημειωθεί. Αν όχι, σημειώνονται και εισάγονται στη στοίβα.

Τα παραπάνω βήματα επαναλαμβάνονται μέχρις ότου να μην υπάρχουν άλλα εικονοστοιχεία, που να μην έχουν ελεγχθεί και σημειωθεί. Μόλις ολοκληρωθεί η εκτέλεση αυτών, η εικόνα που προκύπτει έχει καταταμηστεί, οπότε και επιστρέφεται από τη μέθοδο. Πριν την επιστροφή της εικόνας όμως, υπολογίζεται ο αριθμός των περιοχών και αποθηκεύεται ως ιδιότητα της κλάσης, ούτως ώστε να είναι διαθέσιμη μέσω μίας getter μεθόδου.

4.2.7. Βοηθητική Κλάση Base64

Η τελευταία βοηθητική κλάση είναι η Base64. Αποτελεί πανομοιότυπη κλάση με την ομώνυμη του υποσυστήματος πελάτη, υλοποιώντας ακριβώς τις ίδιες δύο μεθόδους. Σκοπός της υλοποίησής της και σε αυτό το υποσύστημα, είναι η παροχή της δυνατότητας στο servlet ImageReceiver, να δεχτεί την κωδικοποιημένη εικόνα, όπως αυτή αποστέλλεται από το υποσύστημα πελάτη και να την αποκωδικοποιήσει. Μέσω της αποκωδικοποίησης αυτής, επιτυγχάνεται η δυνατότητα ανάπλασης της εικόνας. Η ανακατασκευασμένη εικόνα είναι αυτή που τελικά αποθηκεύεται στο υποσύστημα της βάσης δεδομένων.

4.3. Υποσύστημα Βάσης Δεδομένων

Το υποσύστημα της Βάσης Δεδομένων αποτελεί το τελευταίο υποσύστημα της πλατφόρμας που υλοποιήθηκε. Ο ρόλος του επικεντρώνεται στη συγκέντρωση, κατηγοριοποίηση, ευρετηρίαση και βελτιστοποίηση των πινάκων δεδομένων, τόσο όσον αφορά τον πίνακα χρηστών της πλατφόρμας, όσο και τους πίνακες για τις αγγελίες ενοικιάσεων και πωλήσεων.

Κύριο στέλεχος του υποσυστήματος αυτού αποτελεί μία βάση δεδομένων, στην οποία αποθηκεύονται τα πληροφορίες για τις αγγελίες και τους χρήστες της πλατφόρμας. Κάθε φορά που εισάγεται μία νέα αγγελία στο σύστημα, ή εγγράφεται ένας νέος χρήστης, γίνεται μια καταχώρηση στη βάση δεδομένων.

Το υποσύστημα αυτό, σκοπό έχει τη συγκέντρωση των αγγελιών και χρηστών της πλατφόρμας, προσφέροντας παράλληλα αποδοτικές δυνατότητες αναζήτησης μέσω ευρετηρίασης. Παράλληλα, τα δεδομένα που εισάγονται στη βάση έχουν συγκεκριμένη δομή και μέγεθος, καθιστώντας εφικτή την οργάνωση, διαχείριση και την απόδοση αυτών σε περίπτωση ζήτησης.

Με βάση τα παραπάνω, εύκολα γίνεται αντιληπτή η μεγάλη αναγκαιότητα της ύπαρξης του υποσυστήματος Βάσης Δεδομένων. Χωρίς αυτό, τα δεδομένα που θα έπρεπε να αποθηκευτούν χρησιμοποιώντας το σύστημα αρχείων του λειτουργικού συστήματος, θα ήταν δυσανάλογα ογκώδη ως προς την πληροφορία που μετέφεραν. Παράλληλα, η αξιοποίησή τους θα ήταν δυσχερέστερη και λιγότερο αποδοτική ως προς την ταχύτητα.

Επίσης, από τη μέχρι τώρα ανάλυση του υλοποιηθέντος συστήματος, προκύπτει άμεση η ανάγκη αποθήκευσης μεγάλου όγκου πολυμεσικών αρχείων, και πιο συγκεκριμένα εικόνων, μαζί με τα όποια δεδομένα των αγγελιών.

Είναι γεγονός ότι τα μοντέρνα συστήματα διαχείρισης βάσεων δεδομένων προσφέρουν επιπλέον ευκολίες στην αποθήκευση πολυμεσικών δεδομένων, όπως τελεστές για την υλοποίηση Content Based Image Retrieval. Για παράδειγμα, ο

τελεστής «like», επιτρέπει την αναζήτηση εικόνων με παρόμοιο περιεχόμενο με αυτό κάποιου δείγματος. Η χρήση όμως της βάσης δεδομένων για απευθείας αποθήκευση των εικόνων σε αυτή δε συνίσταται. Κι αυτό διότι κάτι τέτοιο στη συγκεκριμένη υλοποίηση δεν είναι αποδοτικό.

Τούτο συμβαίνει επειδή η αποθήκευση και η διαχείριση μεγάλου όγκου εικόνων σε πολυμεσικές βάσεις δεδομένων παρουσιάζει πολλά προβλήματα. Η αποθήκευση τους γίνεται με δυαδικό τρόπο, δηλαδή τοποθετούνται στη βάση δυαδικά δεδομένα, με αποτέλεσμα να χρειάζεται η μετατροπή της εικόνας από ένα πίνακα εικονοστοιχείων σε σειρά δυαδικών ψηφίων πριν την αποθήκευσή της στη βάση. Κατά τη φάση της ανάκλησης της εικόνας πρέπει να πραγματοποιηθεί η ακριβώς αντίθετη διαδικασία. Σε ό,τι αφορά τη διαχείριση των αποθηκευμένων δεδομένων και τις δυνατότητες CBIR, οι διάφορες υπηρεσίες που προσφέρονται είναι ακόμη σε πειραματικό στάδιο και δεν είναι αποτελεσματικές. Όλα αυτά οδήγησαν σε πειραματικές δοκιμές, οι οποίες κατέστησαν σαφώς αποδοτικότερη την αποθήκευση των εικόνων, όχι απευθείας στη βάση, καθώς έτσι παρατηρούνται πολύ μεγάλες καθυστερήσεις, αλλά σε καταλόγους αρχείων δενδρικής μορφής του εκάστοτε λειτουργικού συστήματος.

Αυτή είναι και η κατεύθυνση, στην οποία κινήθηκε η υλοποίηση του υποσυστήματος αυτού, στα πλαίσια της διπλωματικής εργασίας. Όλη η πληροφορία και τα δεδομένα των αγγελιών εισάγονται στη βάση, ενώ οι εικόνες που συνοδεύουν κάθε αγγελία, αποθηκεύονται σε καταλόγους μέσω του συστήματος αρχείων του εκάστοτε λειτουργικού συστήματος. Παράλληλα, τα όποια μεταδεδομένα απαιτούνται για τη διευκόλυνση στην εκτέλεση λειτουργιών που αφορούν τις εικόνες, βρίσκονται αποθηκευμένα, είτε στη βάση δεδομένων, είτε ανακτούνται από τα ονόματα των αρχείων των εικόνων.

Πιο συγκεκριμένα, οι φωτογραφίες που αφορούν κάθε συγκεκριμένη αγγελία αποθηκεύονται στον ίδιο κατάλογο. Το όνομα του καταλόγου αυτού, ακολουθεί την εξής σύμβαση:

FolderName = type+id

,όπου το type δηλώνει αν πρόκειται για αγγελία ενοικίασης ή πώλησης και το id δηλώνει τον μοναδικό αριθμητικό κωδικό-κλειδί, που αντιστοιχεί σε κάθε καταχώρηση αγγελίας στη βάση.

Σημειώνεται ότι λόγω περιορισμών στην ονοματολογία των αρχείων και καταλόγων από διάφορα λειτουργικά συστήματα, το σύμβολο «+» δεν χρησιμοποιείται, και οι χαρακτήρες των type και id βρίσκονται ενωμένοι μεταξύ τους.

Ακόμη, κάθε φωτογραφία αντιστοιχεί σε ένα αρχείο με συγκεκριμένο όνομα. Ακολουθείται και εδώ σύμβαση στην ονομασία:

ImageName = imgtype+id

,όπου το imgtype δηλώνει τον τύπο της εικόνας, αν δηλαδή απεικονίζει εξωτερικό χώρο, υπνοδωμάτιο, καθιστικό, μπάνιο ή κουζίνα και το id είναι ένας δεκαψηφίος τυχαίος αριθμός, μοναδικός για την ομάδα εικόνων μιας συγκεκριμένης αγγελίας.

Για παράδειγμα, έστω ότι το μονοπάτι για μία αποθηκευμένη εικόνα είναι: /rent2/exterior1234567890. Σε αυτή την περίπτωση μπορούν να εξαχθούν τα εξής μεταδεδομένα για την εικόνα. Από το όνομα του φακέλου προκύπτει ότι η εικόνα αντιστοιχεί σε αγγελία ενοικίασης με αριθμό-κλειδί 2. Επίσης, από το όνομα του αρχείου της εικόνας προκύπτει ότι πρόκειται για εικόνα εξωτερικού χώρου. Καθίσταται έτσι εύκολη πρακτικά η ανάκτηση όλων των σχετικών με τη φωτογραφία δεδομένων της αγγελίας, καθώς και των στοιχείων της ίδιας της αγγελίας, από το όνομα του αρχείου της εν λόγω εικόνας και μέσω της υποβολής ενός απλού ερωτήματος στη βάση.

Η βάση δεδομένων που υλοποιήθηκε αποτελείται από τρεις πίνακες, οι οποίοι για χάρη απλότητας θα αναφέρονται ως Π1, Π2 και Π3. Το σχήμα της βάσης παριστάνεται στο σχήμα 4.3.1 που ακολουθεί.

house_rents	house_sales	users
contact_name VARCHAR(30)	contact_name VARCHAR(30)	username VARCHAR(20)
contact_surname VARCHAR(30)	contact_surname VARCHAR(30)	password VARCHAR(30)
contact_mail VARCHAR(30)	contact_mail VARCHAR(30)	privileges VARCHAR(30)
contact_phone BIGINT(20)	contact_phone BIGINT(20)	name VARCHAR(30)
street VARCHAR(30)	street VARCHAR(30)	surname VARCHAR(30)
streetnum INT(11)	streetnum INT(11)	mail VARCHAR(30)
city VARCHAR(30)	city VARCHAR(30)	phone BIGINT(20)
area VARCHAR(30)	area VARCHAR(30)	street VARCHAR(30)
municipality VARCHAR(30)	municipality VARCHAR(30)	streetnum INT(11)
county VARCHAR(30)	county VARCHAR(30)	city VARCHAR(30)
postcode INT(11)	postcode INT(11)	area VARCHAR(30)
date TIMESTAMP	date TIMESTAMP	municipality VARCHAR(30)
squares INT(11)	squares INT(11)	county VARCHAR(30)
cost BIGINT(20)	cost BIGINT(20)	postcode INT(11)
size VARCHAR(30)	size VARCHAR(30)	reged_since TIMESTAMP
at_floor VARCHAR(16)	at_floor VARCHAR(16)	user_id INT(11)
with_floors VARCHAR(16)	with_floors VARCHAR(16)	
beds VARCHAR(16)	beds VARCHAR(16)	
rooms VARCHAR(16)	rooms VARCHAR(16)	
wc VARCHAR(16)	wc VARCHAR(16)	
balcony VARCHAR(16)	balcony VARCHAR(16)	
garden VARCHAR(16)	garden VARCHAR(16)	
bottoms VARCHAR(16)	bottoms VARCHAR(16)	
heat VARCHAR(30)	heat VARCHAR(30)	
dima VARCHAR(16)	dima VARCHAR(16)	
fireplace VARCHAR(16)	fireplace VARCHAR(16)	
parking VARCHAR(3)	parking VARCHAR(3)	
elevator VARCHAR(6)	elevator VARCHAR(6)	
storage VARCHAR(16)	storage VARCHAR(16)	
furniture VARCHAR(16)	furniture VARCHAR(16)	
commons VARCHAR(16)	commons VARCHAR(16)	
extclr VARCHAR(16)	extclr VARCHAR(16)	
images VARCHAR(6)	images VARCHAR(6)	
type VARCHAR(30)	type VARCHAR(30)	
extras VARCHAR(300)	extras VARCHAR(300)	
age INT(11)	age INT(11)	
renovated VARCHAR(6)	renovated VARCHAR(6)	
pets VARCHAR(6)	pets VARCHAR(6)	
walled_apps VARCHAR(16)	walled_apps VARCHAR(16)	
rent_id INT(11)	sale_id INT(11)	

Σχήμα 4.3.1 Το σχήμα της βάσης δεδομένων

Ο πρώτος πίνακας Π1, περιέχει τα στοιχεία κάθε χρήστη, ο οποίος εγγράφεται στο σύστημα. Πιο συγκεκριμένα, διατηρούνται στοιχεία του επιθυμητού κωδικού

πρόσβασης και ονόματος χρήστη, δεδομένα της διεύθυνσής του, των πραγματικών ονομαστικών στοιχείων του καθώς και πληροφορίες για ενδεχόμενη επικοινωνία μαζί του. Επίσης κάθε εγγραφή στον πίνακα Π1, αντιπροσωπεύεται από μοναδικό κωδικό αριθμό, που αποτελεί και το κύριο κλειδί.

Ο δεύτερος πίνακας, ο Π2, περιέχει τα στοιχεία για κάθε αγγελία ενοικίασης. Πιο συγκεκριμένα, στον πίνακα αυτό αποθηκεύονται πληροφορίες σχετικά με τα χαρακτηριστικά της οικίας, τη διεύθυνσή του και το αν συνοδεύεται από εικόνες ή όχι. Επιπρόσθετα, κρατούνται τα ονομαστικά στοιχεία του ιδιοκτήτη της οικίας, καθώς και τρόποι πιθανής επικοινωνίας μαζί του. Επίσης, το σύστημα της βάσης δεδομένων φροντίζει κατά τη διάρκεια της εισαγωγής μιας νέας αγγελίας να κρατά και την ημερομηνία και ώρα, που αυτή εισήχθηκε. Τέλος, κάθε αγγελία χαρακτηρίζεται από ένα μοναδικό κωδικό αριθμό, ο οποίος αποτελεί και το κύριο κλειδί του Π2.

Ο τρίτος και τελευταίος πίνακας της βάσης, Π3, είναι πανομοιότυπος με τον Π2. Η διαφορά τους έγκειται στο γεγονός ότι ο Π2 χρησιμοποιείται για αποθήκευση αγγελιών ενοικίασης, ενώ ο Π3 αποθηκεύει αγγελίες πώλησης.

Συνεπώς, δύο αγγελίες μπορούν να έχουν τον ίδιο κωδικό αριθμό, αν και μόνο αν η μία αποτελεί αγγελία πώλησης και η άλλη ενοικίασης.

Τέλος, η βάση δεδομένων δεν είναι ανάγκη να αποθηκευτεί στον ίδιο χώρο μνήμης με το υποσύστημα εξυπηρετητή, μιας και οι μέθοδοι επικοινωνίας του τελευταίου επιτρέπουν την αποστολή και λήψη δεδομένων μέσω διαδικτύου. Το γεγονός αυτό καθιστά το υλοποιηθέν σύστημα περισσότερο ευέλικτο.

5. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα των πειραματικών μετρήσεων της αποδοτικότητας και ευστοχίας των μεθόδων ψηφιακής επεξεργασίας εικόνας που περιγράφηκαν στα προηγούμενα κεφάλαια.

Πιο συγκεκριμένα, παρουσιάζονται τα αποτελέσματα του αλγόριθμου σύγκρισης εικόνων μέσω συζευγμένων ιστογραμμάτων, του αλγόριθμου εύρεσης τύπου πατώματος, του αλγόριθμου εύρεσης χρώματος εξωτερικής όψης οικίας και τέλος, του αλγόριθμου εύρεσης επιπλωμένου χώρου.

Για το σκοπό αυτό χρησιμοποιήθηκε ο φωτογραφικό υλικό από τον ιστοχώρο Google Images, από τον οποίο και συλλέχτηκαν 160 φωτογραφίες. Κατόπιν, οι φωτογραφίες αυτές χωρίστηκαν στις παρακάτω κατηγορίες:

A) 30 φωτογραφίες απεικόνιζαν άδειους χώρους, κυρίως σαλόνια και υπνοδωμάτια, οι τοίχοι των οποίων είχαν διάφορα χρώματα. Παράλληλα, υπήρχε ποικιλία και στον τύπο του πατώματος που απεικονίζονταν σε κάθε εικόνα. Πιο συγκεκριμένα, 14 εξ' αυτών απεικόνιζαν πάτωμα τύπου παρκέ, άλλες 10 πλακάκια και οι υπόλοιπες 6 μωσαϊκό.

B) Στις επόμενες 20 φωτογραφίες απεικονίζονταν εξωτερικές όψεις σπιτιών, από τις οποίες 8 περιείχαν και κάποιας μορφής κήπο.

Γ) Επίσης, επιλέχτηκαν 35 φωτογραφίες από χώρους κουζίνας. Από αυτές, 10 εμφάνιζαν πλήρως επιπλωμένους χώρους, ενώ στις υπόλοιπες 25 υπήρχαν μόνο τα βοηθητικά εντοιχισμένα έπιπλα, όπως για παράδειγμα ντουλάπια.

Δ) Στην τέταρτη κατηγορία βρίσκονται 20 εικόνες μάνιων. Όλα τα πατώματα σε αυτές τις φωτογραφίες είναι πλακάκια.

E) Οι 40 φωτογραφίες αυτής της κατηγορίας απεικονίζουν σαλόνια. Σε όλες αυτές τις εικόνες, υπάρχουν έπιπλα, με μεγάλο πλήθος διαφορετικών χρωμάτων και σχεδίων. Επίσης, 5 από αυτές έχουν πάτωμα τύπου παρκέ, 10 πλακάκια, 5 μωσαϊκό ενώ στις υπόλοιπες 20 δεν είναι ευδιάκριτο το μεγαλύτερο μέρος του πατώματος, λόγω παρουσίας χαλιών.

ΣΤ) Στην έκτη και τελευταία κατηγορία απεικονίζονται επιπλωμένα υπνοδωμάτια, 15 στον αριθμό. Στα 5 εξ'αυτών παρατηρήθηκαν πλακάκια για πάτωμα, ενώ οι υπόλοιπες 10 είχαν παρκέ.

Σημειώνεται ότι συγκεντρωτικά, από τις 160 φωτογραφίες, 55 θεωρήθηκε ότι απεικονίζουν μη επιπλωμένους χώρους, ενώ 65 μη επιπλωμένους χώρους. Συνεπώς οι εικόνες με μπάνια ή εξωτερικές όψεις σπιτιών δε λήφθηκαν υπ' όψιν όσον αφορά την απεικόνιση επιπλωμένου χώρου.

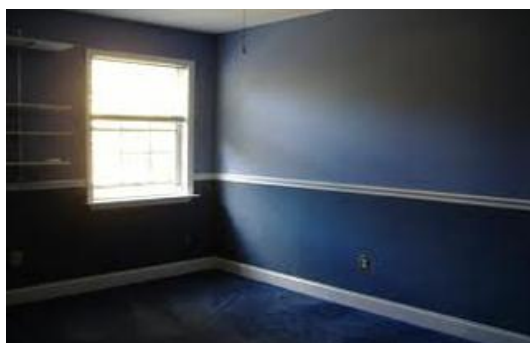
Η διαδικασία που ακολουθήθηκε για τη μέτρηση της ορθότητας του αλγόριθμου σύγκρισης εικόνων ήταν η εξής:

Από το σύνολο των αποθηκευμένων φωτογραφιών επιλέγονταν κάθε φορά μία μόνο κατηγορία για κάθε δοκιμή, για την οποία ο αλγόριθμος μπορούσε να παράξει ορθά αποτελέσματα. Τουτέστιν, συγκρίνονταν κάθε φορά ως προς την ομοιότητα φωτογραφίες που απεικόνιζαν ίδιους χώρους, όπως για παράδειγμα υπνοδωμάτια. Κάθε εικόνα αποτελούσε διαδοχικά είσοδο για τον αλγόριθμο, ο οποίος αποφαινόταν ως προς την ομοιότητά της σε σύγκριση με όλες τις υπόλοιπες.

Τα κριτήρια που χρησιμοποιήθηκαν για τον προσδιορισμό της ομοιότητας μεταξύ δύο εικόνων, ούτως ώστε να προκύψουν συμπεράσματα σχετικά με την αποδοτικότητα της μεθόδου ήταν τόσο χρωματικά όσο και χωρικά. Πιο συγκεκριμένα, ανόμοιοι θεωρήθηκαν δύο χώροι, οι οποίοι αποτελούνται από διαφορετικού χρώματος τοίχους, μεγάλη διαφορά στην πόσοτητα, χρώμα και διάταξη επίπλων σε αυτούς καθώς μεγάλη απόκλιση στη γωνία από την οποία έχει τραβηχτεί η φωτογραφία. Δύο τέτοια παραδείγματα παρουσιάζονται στις εικόνες 5.1 και 5.2.



Εικόνα 5.1 Παράδειγμα όμοιων χώρων



Εικόνα 5.2 Παράδειγμα ανόμοιων χώρων

Συνολικά έγιναν 4750 συγκρίσεις. Τα αποτελέσματα αυτών παρουσιάζονται αναλυτικά στον παρακάτω πίνακα.

Πίνακας 5.1 Πειραματικά αποτελέσματα για τη μέτρηση της απόδοσης του αλγόριθμου σύγκρισης εικόνων

Κατηγορία Εικόνων	Σύνολο Φωτογραφιών	Σύνολο Συγκρίσεων	Σφάλματα Συγκρίσεων	Απόδοση Επί τους Εκατό
Άδειοι Χώροι	30	900	124	86,2%
Εξωτερική Όψη	20	400	56	86%
Κουζίνα	35	1225	105	91,4%
Μπάνιο	20	400	236	41%
Σαλόνι	40	1600	485	69,7%
Υπνοδωμάτιο	15	225	49	78,2%
Συνολικά	160	4750	1055	77,8%

Στη συνέχεια, έγινε μέτρηση της ορθότητας του αλγόριθμου εύρεσης επιπλωμένων χώρων με την εξής διαδικασία:

Από το σύνολο των αποθηκευμένων φωτογραφιών όλες εκείνες, οι οποίες απεικόνιζαν επιπλωμένους και μη χώρους. Όπως δηλαδή προαναφέρθηκε, από τη δοκιμή εξαιρέθηκαν οι φωτογραφίες μπάνιων. Κάθε μία από τις επιλεγμένες εικόνες αποτελούσε διαδοχικά είσοδο για τον αλγόριθμο, ο οποίος αποφαινόταν σχετικά με το αν ο απεικονιζόμενος χώρος ήταν επιπλωμένος ή όχι.

Τα κριτήρια που χρησιμοποιήθηκαν για την κατάταξη ενός χώρου σε επιπλωμένο ή όχι ήταν δύο. Αφ' ενός απαιτούνταν πλήρης απουσία οποιουδήποτε επίπλου, αφ'ετέρου εντοιχισμένα έπιπλα όπως ντουλάπια κουζίνας δε λαμβάνονταν υπ' όψιν. Ένα παράδειγμα επιπλωμένου και μη χώρου παρουσιάζεται στην εικόνα 5.3.



Εικόνα 5.3 Παράδειγμα επιπλωμένου (αριστερά) και μη (δεξιά) χώρου

Τα αποτελέσματα του αλγόριθμου, έχοντας ως είσοδο τις παραπάνω 120 εικόνες αναφέρονται στον παρακάτω πίνακα:

Πίνακας 5.2 Πειραματικά αποτελέσματα για τη μέτρηση της απόδοσης του αλγόριθμου εύρεσης επιπλωμένων χώρων

Κατηγορία Εικόνων	Σύνολο Φωτογραφιών	Σφάλματα	Απόδοση Επί τοις Εκατό
Άδειοι Χώροι	30	1	96,7%
Κουζίνα	35	3	91,4%
Σαλόνι	40	5	87,5%
Υπνοδωμάτιο	15	2	86,7%
Συνολικά	120	11	90,8%

Για τις μετρήσεις της απόδοσης του αλγόριθμου εύρεσης τύπου πατώματος χρησιμοποιήθηκαν όλες οι εικόνες, πλην αυτών που απεικόνιζαν εξωτερικές όψεις. Αξιοσημείωτο είναι ότι για καλύτερη προσέγγιση της απόδοσης του εν λόγω αλγόριθμου, όταν αυτός χρησιμοποιείται από την πλατφόρμα που υλοποιήθηκε για αυτή τη διπλωματική εργασία, χρησιμοποιήθηκαν και οι εικόνες σαλονιών, στις

οποίες δε διακρίνεται καθαρά ο τύπος δαπέδου, λόγω χαλιών ή επίπλωσης. Ένα τέτοιο παράδειγμα παρουσιάζεται στην εικόνα 5.4.



Εικόνα 5.4 Παράδειγμα εικόνας όπου δε διακρίνεται καθαρά το δάπεδο

Τα αποτελέσματα του αλγόριθμου, έχοντας ως είσοδο τις παραπάνω 140 εικόνες αναφέρονται στον παρακάτω πίνακα:

Πίνακας 5.3 Πειραματικά αποτελέσματα για τη μέτρηση της απόδοσης του αλγόριθμου εύρεσης τύπου δαπέδου

Κατηγορία Εικόνων	Σύνολο Φωτογραφιών	Σφάλματα	Απόδοση Επί τους Εκατό
Άδειοι Χώροι	30	1	96,7%
Κουζίνα	35	11	68,6%
Μπάνια	20	3	85%
Σαλόνι	40	18	55%
Υπνοδωμάτιο	15	3	80%
Συνολικά	140	36	74,3%

Τέλος, για τον έλεγχο της αποδοτικότητας του αλγόριθμου εύρεσης χρώματος εξωτερικής όψης, χρησιμοποιήθηκαν μόνο οι 20 αντίστοιχες φωτογραφίες από το

σύνολο των 160. Τα αποτελέσματα του εν λόγω αλγόριθμου παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 5.4 Πειραματικά αποτελέσματα για τη μέτρηση της απόδοσης του αλγόριθμου εύρεσης χρώματος εξωτερικού χώρου

Κατηγορία Εικόνων	Σύνολο Φωτογραφιών	Σφάλματα	Απόδοση Επί τοις Εκατό
Εξωτερική Όψη	20	3	85%
Συνολικά	20	3	85%

6. ΕΠΙΛΟΓΟΣ

Στα πλαίσια αυτής της διπλωματικής εργασίας, υλοποιήθηκε ευφυής φορητή πλατφόρμα ηλεκτρονικού εμπορίου, με έμφαση σε χαρακτηριστικά που ενσωματώνουν τεχνικές και τεχνολογία ψηφιακής επεξεργασίας εικόνας. Πιο συγκεκριμένα, η πλατφόρμα υλοποιεί σύστημα συστάσεων για τους χρήστες, ενώ παράλληλα εμφανίζει δυνατότητες αυτόματης συμπλήρωσης πεδίων. Βάση για τις παραπάνω δυνατότητες, αποτέλεσε σε κάθε περίπτωση μετα-πληροφορία, η οποία εξάγεται από εικόνες και την επεξεργασία αυτών. Οι τεχνικές ψηφιακής επεξεργασίας εικόνας που χρησιμοποιήθηκαν αποτελούνται από την ανίχνευση ακμών του Canny, την κατάτμηση εικόνας με τη μέθοδο ανάπτυξης περιοχών και τη μέθοδο σύγκρισης εικόνων με βάση τα συζευγμένα ιστογράμματα.

Οι παραπάνω μέθοδοι, έδωσαν τη δυνατότητα στο χρήστη να εισάγει αγγελίες στο σύστημα, οι οποίες περιέχουν και εικόνες. Στην περίπτωση εισαγωγής αγγελίας με εικόνες, ήταν δυνατή η αυτόματη εξαγωγή συμπερασμάτων για τον τύπο πατώματος, το χρώμα της εξωτερικής όψης και το βαθμό επίπλωσης της οικίας που απεικονίζονταν, όπως αυτά προέκυπταν από την επεξεργασία των εκάστοτε εικόνων. Με βάση αυτά τα συμπεράσματα, συμπληρώνονταν αυτόματα πεδία στη φόρμα εισαγωγής αγγελίας, συμβάλλοντας έτσι στη διευκόλυνση των χρηστών. Κατόπιν, με βάση είτε δεδομένα, είτε φωτογραφίες, ο κάθε χρήστης μπορούσε να αναζητήσει αγγελίες που τον ενδιέφεραν. Τα αποτελέσματα εμφανίζονταν προσφέροντας συνοπτικές πληροφορίες για κάθε αγγελία, καθώς και γεωγραφικά στοιχεία μέσω ηλεκτρονικών χαρτών. Παράλληλα, δίνονταν η δυνατότητα πλήρους προβολής των στοιχείων των αγγελιών. Τέλος, υλοποιήθηκε και σύστημα συστάσεων, το οποίο έχοντας ως δεδομένες τις αναζητήσεις των χρηστών, παρουσίαζε στους χρήστες αγγελίες, οι οποίες συνοδεύονταν από φωτογραφίες όμοιες με τις εικόνες των αποτελεσμάτων της αναζήτησης.

Η πλατφόρμα χωρίστηκε σε τρία επιμέρους υποσυστήματα, του πελάτη, του εξυπηρετητή και της βάσης δεδομένων. Καθένα από αυτά περιγράφηκε ενδελεχώς στα προηγούμενα κεφάλαια. Οι υλοποίησή των υποσυστημάτων πελάτη και εξυπηρετητή έγινε με χρήση τεχνολογίας Java, ενώ για το υποσύστημα της βάσης

δεδομένων επιλέχθηκε η MySQL γλώσσα. Σε κάθε περίπτωση, το υλοποιηθέν σύστημα διακρίθηκε για τη φορητότητά του, την αποδοτική διαχείριση μνήμης και την πλήρη ανεξαρτησία από το περιβάλλον εκτέλεσης.

Τέλος, το σύστημα ελέγχθηκε με μεγάλο αριθμό αγγελιών και των αντίστοιχων φωτογραφιών τους, που απεικόνιζαν πληθώρα διαφορετικών χώρων των οικιών, τόσο εσωτερικών όσο και εξωτερικών. Οι φωτογραφίες και αγγελίες αυτές εισήχθησαν στη βάση δεδομένων με χρήση προσομοιωτών των αντίστοιχων συσκευών των μεγαλύτερων και δημοφιλέστερων κατασκευαστών κινητών τηλεφώνων. Τα αποτελέσματα του ελέγχου κατέδειξαν την ασφάλεια, σταθερότητα και αποδοτικότητα του συστήματος.

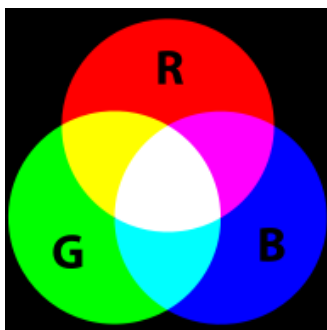
Προτάσεις για μελλοντική επέκταση της διπλωματικής αφορούν τόσο την υλοποίηση νέων χαρακτηριστικών, όσο και τη βελτίωση των ήδη υπάρχοντων. Πιο συγκεκριμένα, η ευφυής εισαγωγή αγγελιών μπορεί να βελτιωθεί με χρήση μεταναζήτησης τιμών σε διαδικτυακούς τόπους αγγελιών. Με τον τρόπο αυτό, θα προσφέρεται ένα σύστημα συστάσεων τιμών ανάλογα με τα χαρακτηριστικά της οικίας στην οποία αναφέρεται η αγγελία, όπως το μέγεθος, η παλαιότητα και η περιοχή. Επίσης, είναι δυνατή η προσθήκη περαιτέρω τεχνικών για την αύξηση του αριθμού των αυτόματα συμπληρωμένων πεδίων στη φόρμα εισαγωγής αγγελίας. Επιπρόσθετα, η αναζήτηση αγγελιών μπορεί να εμπλουτιστεί με χρήση αλγόριθμου αντίχνευσης μοτίβου (pattern matching algorithm). Με αυτό τον τρόπο θα καταστεί δυνατή η εύρεση αγγελιών, οι φωτογραφίες των οποίες περιέχουν συγκεκριμένα αντικείμενα, όπως τζάκι, πισίνα ή φυτά.

7. ΠΑΡΑΡΤΗΜΑΤΑ

7.1. Χρωματικά μοντέλα εικόνων

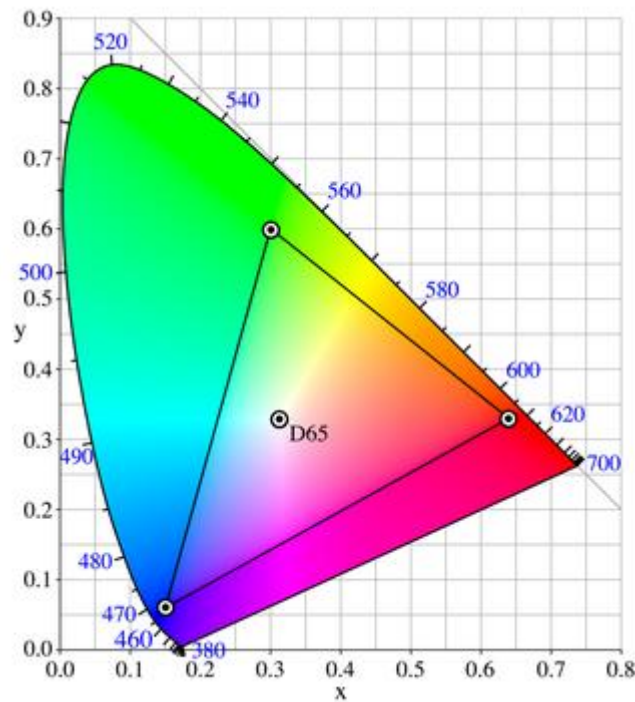
Κάθε εικονοστοιχείο της εικόνας ενδέχεται να περιέχει τρία χρώματα (κόκκινο, πράσινο, μπλε - RGB) σε διαφορετικές αναλογίες, διαμορφώνοντας το RGB μοντέλο. Στα πλαίσια της υλοποίησης αυτής της διπλωματικής εργασίας έγινε χρήση και το χρωματικού μοντέλου HSV. Στη συνέχεια γίνεται σύντομη αναφορά σε αυτά τα χρωματικά μοντέλα.

Η ιδιότητα του **RGB** μοντέλου της προσομοίωσης του μοντέλου της ανθρώπινης όρασης, το καθιστά ιδιαίτερα εύχρηστο. Τα δύο πιο συνηθισμένα μοντέλα RGB που χρησιμοποιούνται στην πράξη, είναι το sRGB και το Adobe RGB. Από το 2007 το sRGB είναι το πιο δημοφιλές, ειδικά για τις ψηφιακές κάμερες, λόγω της επάρκειάς του σε ό,τι αφορά τις τυπικές εφαρμογές. Οι τρεις τιμές των εικονοστοιχείων μίας εικόνας, σύμφωνα με αυτό, αντιστοιχούν στις αναλογίες κόκκινου, πράσινου και μπλε χρώματος. Από τα bit βάθους χρώματος που αποτελείται κάθε εικονοστοιχείο εξαρτάται το πλήθος των χρωμάτων του. Με βάθος χρώματος 24 bit, δηλαδή 8 bit ανά κανάλι (R, G και B), αναπαριστώνται 8 εκατομμύρια χρώματα. Επομένως, η τιμή [0,0,0] αναπαριστά το μαύρο χρώμα, η [255,0,0] το κόκκινο, η [0,255,0] το πράσινο, η [0,0,255] το μπλε και η τιμή [255,255,255] αντιστοιχεί στο λευκό χρώμα.



Εικόνα 7.1.1 : Μίξη χρωμάτων στο RGB

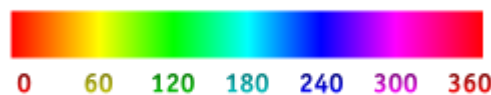
Στην εικόνα 7.1.2 παρουσιάζεται το sRGB μοντέλο.



Εικόνα 7.1.2 : Το sRGB χρωματικό μοντέλο

Το **HSV**, που αλλιώς αναφέρεται και ως HSB (απόχρωση - Hue, κορεσμός - Saturation, φωτεινότητα - Brightness), μετασχηματίζει μη γραμμικά το RGB μοντέλο. Χρησιμοποιείται στις εφαρμογές όπου χρειάζεται ο ταχύτερος υπολογισμός των συνεχών περιοχών χρωμάτων μίας εικόνας. Αποτελείται από τις εξής συνιστώσες:

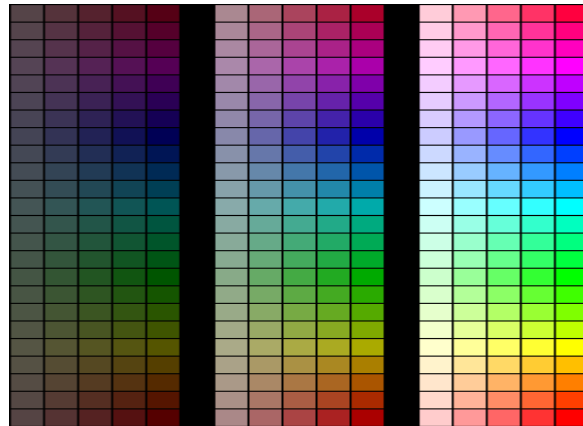
- Την απόχρωση Hue, που δηλώνει τον τύπο του χρώματος (όπως κόκκινο, πράσινο ή κίτρινο) με τιμές από 0 έως 360 (αν και σε πολλές εφαρμογές γίνεται κανονικοποίηση στο 0-100)



- Τον κορεσμό Saturation του χρώματος, με τιμές από 0 έως 100%. Όσο μικρότερη τιμή λαμβάνει το H τόσο πιο γκρίζο είναι ένα χρώμα.

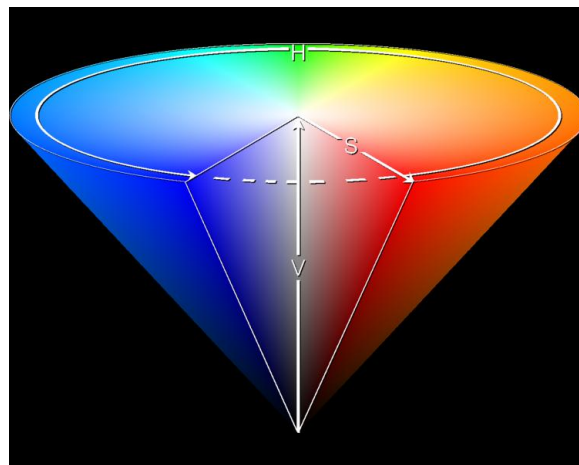
- Το Value, η τιμή της φωτεινότητας, με τιμές από 0 έως 100%. Η φωτεινότητα αναφέρεται στην ποσότητα φωτός σε ένα χρώμα. Το RGB μοντέλο δίνεται από τη σχέση $V = \frac{R + G + B}{3}$.

Στο παρακάτω σχήμα δίνονται οι τιμές του S για τρεις τιμές του V.



Εικόνα 7.1.3

Το HSV μοντέλο παρουσιάζεται στην εικόνα 7.1.4



Εικόνα 7.1.4 : Το HSV μοντέλο

7.2. Διαμορφώσεις εικόνων

Το σύστημα που υλοποιήθηκε σε αυτή τη διπλωματική εργασία λαμβάνει ως είσοδο προς επεξεργασία εικόνες διάφορων διαμορφώσεων, ανάλογα την πηγή προέλευσής τους. Στη συνέχεια, αναφέρονται οι διαμορφώσεις που η υλοποίηση του συστήματος είναι ικανή να διαχειριστεί. Αυτές είναι η JPEG, η GIF και η PNG.

Η **JPEG** διαμόρφωση είναι ένας τυποποιημένος μηχανισμός συμπίεσης εικόνας. Έλαβε το όνομα της από τα αρχικά της επιτροπής που την πρότεινε (Joint Photographic Experts Group).

Η JPEG συμπιέζει την εικόνα με παράγοντα 20 προς 1, δηλαδή συμπιέζει μία εικόνα μεγέθους 2MB σε εικόνα 100KB. Παρ' όλα αυτά, κατά τη συμπίεση παρουσιάζει απώλεια των χαμηλά μεταβαλλόμενων συνιστωσών χρώματος της εικόνας. Η απώλεια αυτή δεν είναι εμφανής, διότι αξιοποιείται η ιδιότητα της ευαισθησίας του ανθρώπινου μάτιου στις υψηλές συχνότητες μεταβολής των εντάσεων των εικονοστοιχείων (για παράδειγμα στις ακμές της εικόνας). Προσφάτως αναπτύχθηκαν βελτιώσεις του προτύπου JPEG, ούτως ώστε να μην εμφανίζονται απώλειες κατά τη συμπίεση, όπως το JPEG-LS (LossLess). Τέλος, χρησιμοποιείται 24 bit βάθος χρωμάτων. Το αρχείο μιας εικόνας JPEG έχει κατάληξη .jpg ή .jpeg, .jpe, .jfif, .jfi και .jif.

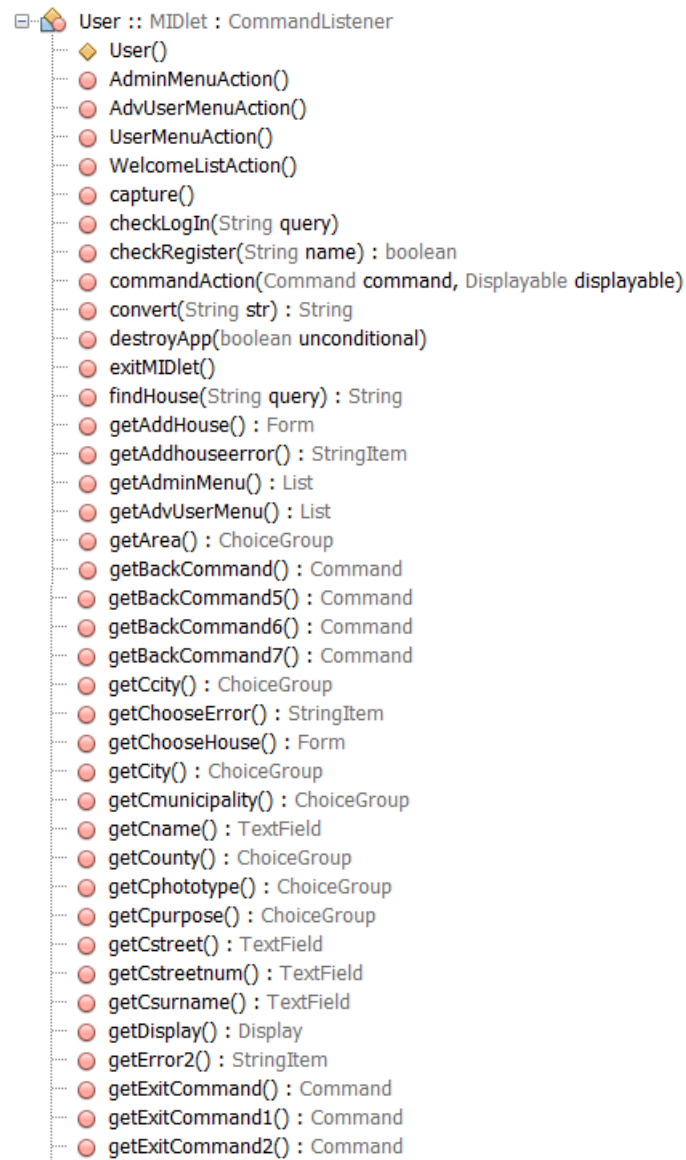
Το **GIF** (Graphics Interchange Format), είναι μια διαμόρφωση εικόνων που είναι πολύ δημοφιλής στον παγκόσμιο ιστό. Χρησιμοποιεί τον αλγόριθμο LZW και προσφέρει συμπίεση χωρίς απώλειες, σε αντίθεση με το JPEG, με αποτέλεσμα τη μη παραμόρφωση των ακμών της εικόνας. Χρησιμοποιεί 8 bit βάθος χρωμάτων. Είναι η ιδανική διαμόρφωση για μεταφορά εικόνων μέσω δικτύου, ειδικά για τις εικόνες των αποτελεσμάτων που αποστέλλονται σε ένα κέντρο ελέγχου. Για λόγους παροχών αδειών χρήσης του GIF, αναπτύχθηκε από την Unisys (η οποία έχασε τη διαμάχη για τη χρήση του GIF) το **PNG** (Portable Network Graphics), το οποίο αποτελεί βελτίωση του GIF, αφού υποστηρίζει μεγαλύτερο από 8 bit βάθος χρωμάτων.

Η κατάληξη των GIF αρχείων είναι .gif και των png .png.

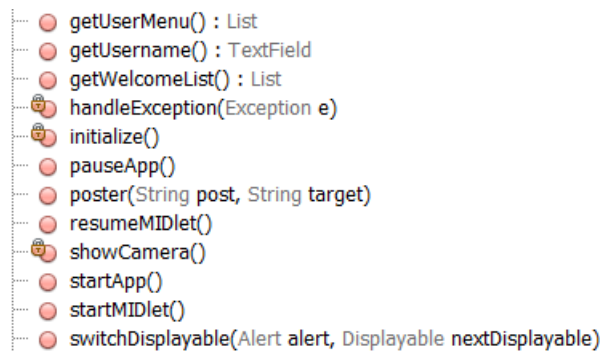
7.3. Κατάλογος Κλάσεων και Μεθόδων

Υποσύστημα Πελάτη:

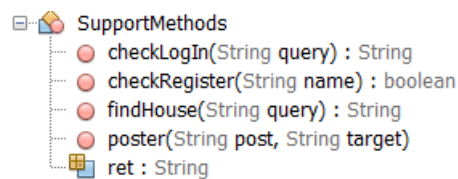
Κλάση User:



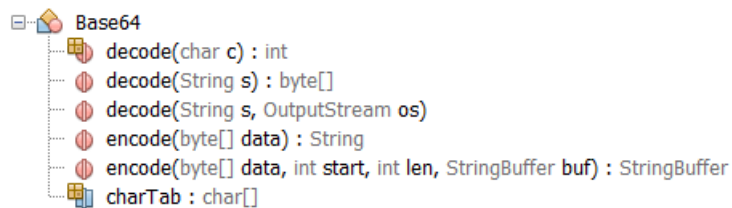
- getExitCommand3() : Command
- getExitCommand4() : Command
- getExitCommand5() : Command
- getHage() : TextField
- getHappliances() : ChoiceGroup
- getHarea() : ChoiceGroup
- getHatfloor() : ChoiceGroup
- getHbalconies() : ChoiceGroup
- getHbaths() : ChoiceGroup
- getHbeds() : ChoiceGroup
- getHbottoms() : ChoiceGroup
- getHcity() : ChoiceGroup
- getHclima() : ChoiceGroup
- getHcommons() : ChoiceGroup
- getHcounty() : ChoiceGroup
- getHelevator() : ChoiceGroup
- getHextras() : TextField
- getHextype() : ChoiceGroup
- getHfireplace() : ChoiceGroup
- getHfloors() : ChoiceGroup
- getHfurniture() : ChoiceGroup
- getHgarden() : ChoiceGroup
- getHheat() : ChoiceGroup
- getHimages() : ChoiceGroup
- getHmail() : TextField
- getHmunicipality() : ChoiceGroup
- getHname() : TextField
- getHparking() : ChoiceGroup
- getHpets() : ChoiceGroup
- getHphone() : TextField
- getHpostal() : TextField
- getHprice() : TextField
- getHrenovated() : ChoiceGroup
- getHrooms() : ChoiceGroup
- getHsize() : ChoiceGroup
- getHsquares() : TextField
- getHstorage() : ChoiceGroup
- getHstreet() : TextField
- getHstreetnum() : TextField
- getHsurname() : TextField
- getHtype() : ChoiceGroup
- getItemCommand() : Command
- getLogInForm() : Form
- getLogname() : TextField
- getLogpass() : TextField
- getMail() : TextField
- getMunicipality() : ChoiceGroup
- getName() : TextField
- getOkCommand() : Command
- getOkCommand1() : Command
- getOkCommand2() : Command
- getOkCommand3() : Command
- getOkCommand4() : Command
- getOkCommand5() : Command
- getPassword() : TextField
- getPhone() : TextField
- getPostalcode() : TextField
- getPurpose() : ChoiceGroup
- getRegerror() : StringItem
- getRegisterForm() : Form
- getSpacer() : Spacer
- getStreet() : TextField
- getStreetnumber() : TextField
- getSurname() : TextField
- getUserMenu() : List
- getUsername() : TextField
- getWelcomeList() : List



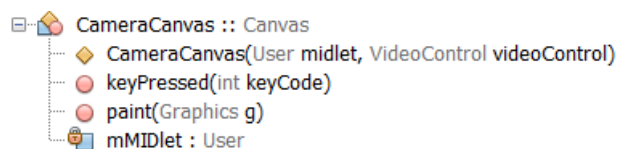
Κλάση SupportMethods:



Κλάση Base64:

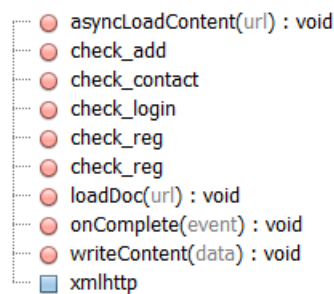


Κλάση CameraCanvas:

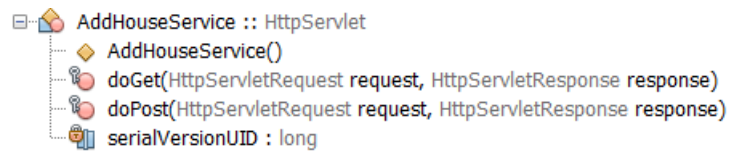


Υποσύστημα Εξυπηρετητή:

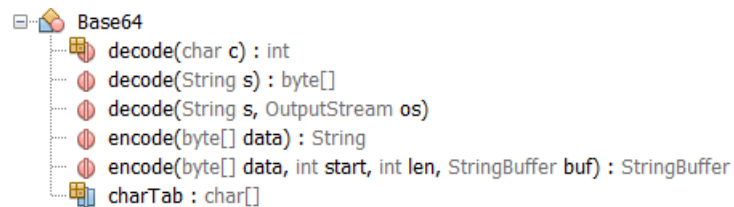
Javascript Αρχείο Code.js:



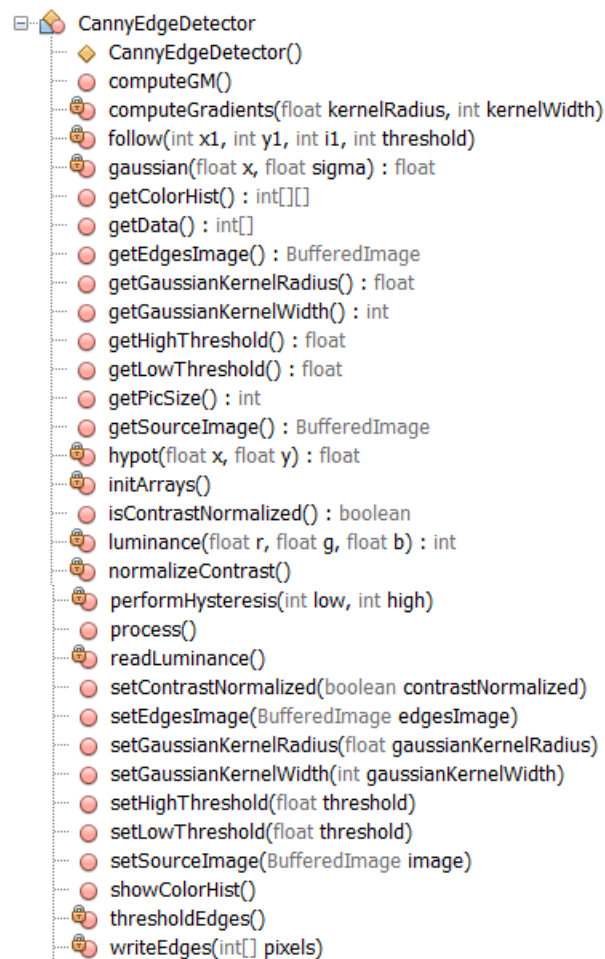
Servlet AddHouseService:



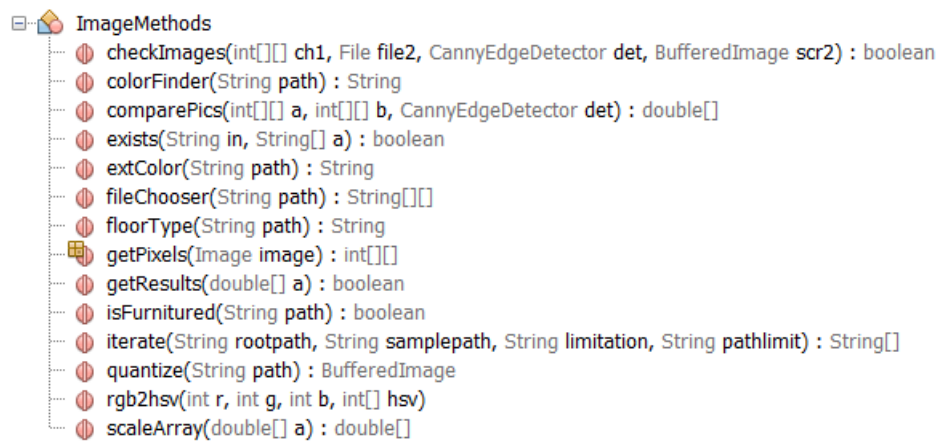
Κλάση Base64:



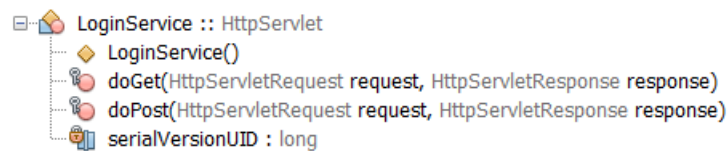
Κλάση CannyEdgeDetector:



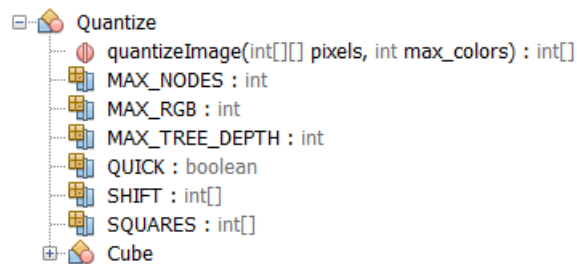
Κλάση ImageMethods:



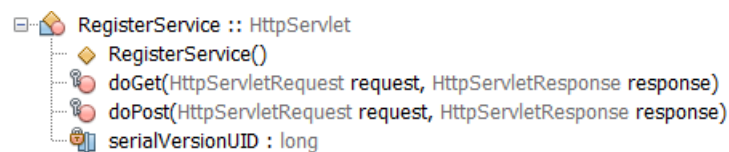
Servlet LoginService:



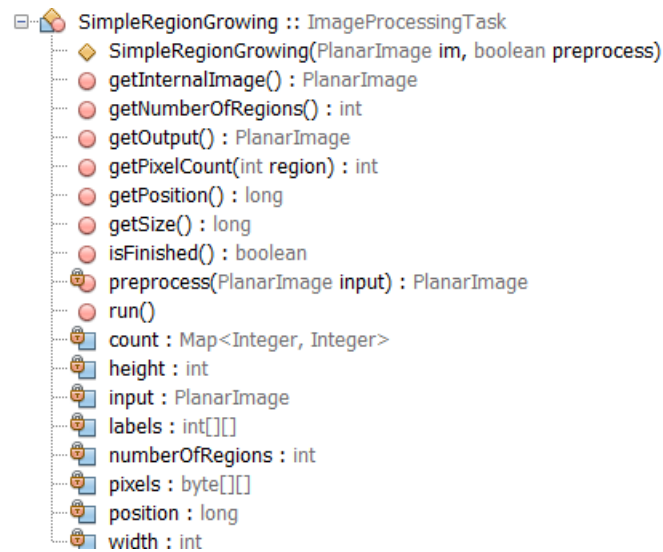
Κλάση Quantize:



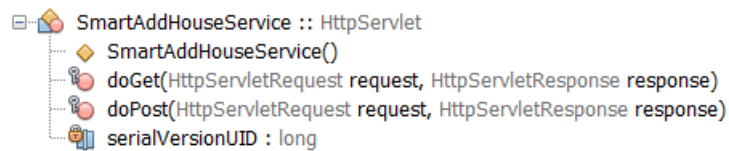
Servlet RegisterService:



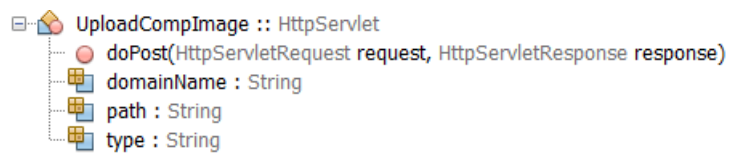
Κλάση SimpleRegionGrowing:



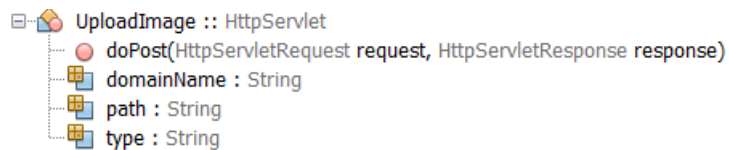
Servlet SmartAddHouseService:



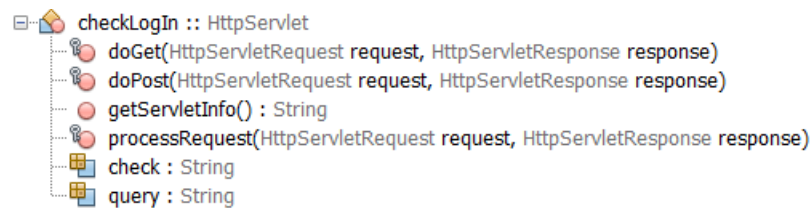
Servlet UploadCompImage:



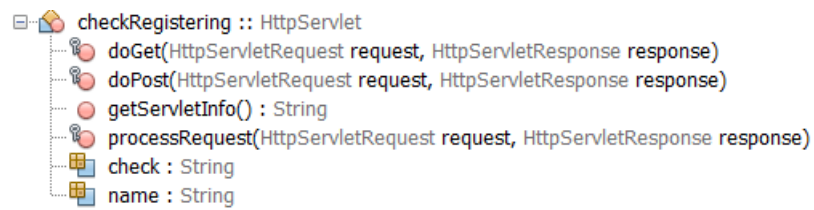
Servlet UploadImage:



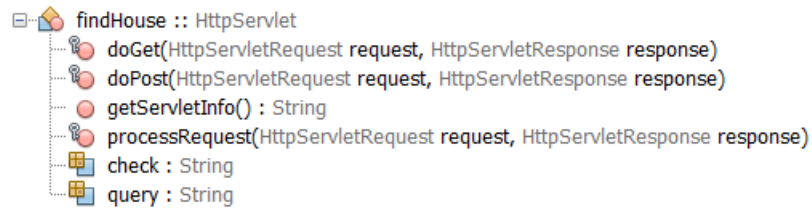
Servlet CheckLogIn:



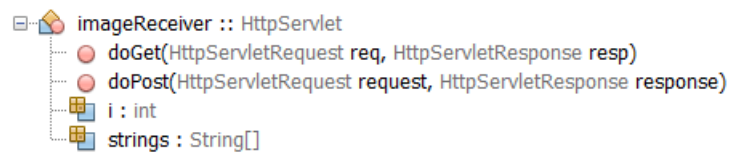
Servlet checkRegistering:



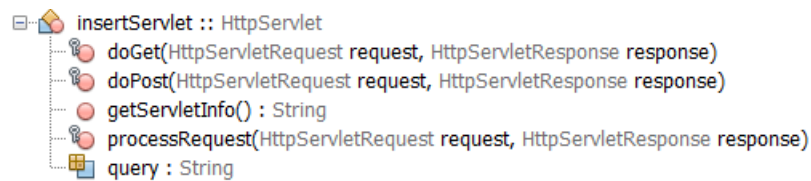
Servlet findHouse:



Servlet imageReceiver:



Servlet insertServlet:



7.4. Βιβλιογραφία

1. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", Gediminas Adomavicius, Alexander Tuzhilin, *IEEE Transactions On Knowledge And Data Engineering*, Vol. 17, No 6, 2005
2. "Analysis of Recommendation Algorithms for E-Commerce", Sarwar Badrul, George Karypis, Joseph Konstan, John Riedl, *In proceedings of ACM E-Commerce 1999 conference*
3. <http://en.wikipedia.org/wiki/Base64>, Base64 Encoding, τελευταία μέρα επίσκεψης 19/1/2011
4. "Histograms Analysis for image retrieval", R. Brunelli, O. Mich, *Pattern Recognition*, Vol. 34, Issue 6, 2001
5. http://en.wikipedia.org/wiki/Canny_edge_detector, Canny Edge Detection, τελευταία μέρα επίσκεψης 19/1/2011
6. http://en.wikipedia.org/wiki/Color_histogram, Color Histogram, τελευταία μέρα επίσκεψης 9/1/11
7. http://en.wikipedia.org/wiki/Color_quantization, Color Quantization, τελευταία μέρα επίσκεψης 16/1/11
8. "Content Based Retrieval of Still Images ", John P. Eakins, Graham, *Technical Report – University of Northumbria at Newcastle*, 1999
9. Digital Image Processing - An Algorithmic Introduction Using Java, Nick Effort, Addison-Wesley Publishing Company
10. http://en.wikipedia.org/wiki/Edge_detection, Edge Detection Algorithms, τελευταία μέρα επίσκεψης 19/1/2011

11. <http://code.google.com/apis/maps/documentation/javascript/>, Google Maps Javascript API version 3, τελευταία μέρα επίσκεψης 19/1/11
12. *Servlets And Java Server Pages, Volume One: Core Technologies*, Marty Hall and Larry Brown, Prentice Hall
13. "Image Segmentation Techniques", R. M. Haralick, L. G. Shapiro, *Comp. Vis. Graph Image Process.*, Vol. 29, pp 100-132, 1985
14. <http://en.wikipedia.org/wiki/HSV>, HSV Color Model, τελευταία μέρα επίσκεψης 19/1/11
15. <http://www.oracle.com/technetwork/java/javame/documentation/index.html>, Java ME Reference, τελευταία μέρα επίσκεψης 17/1/11
16. http://en.wikipedia.org/wiki/JavaServer_Pages, Java Server Pages, τελευταία μέρα επίσκεψης 16/1/11
17. <http://java.sun.com/products/jsp/reference/api/index.html>, JSP API Reference, τελευταία μέρα επίσκεψης 13/1/11
18. "Lightweight security for mobile commerce transactions", Kwok-Yan Lam, Siu-Leung Chung, Ming Gu, Jia-Guang Sun, *Computer Communications* 26, pp 2052-2063, 2003
19. <http://www.marketingcharts.com/interactive/mobile-device-popularity-surges-12020/comscore-smartphone-penetration-marketshare-feb-2010.jpg>, Marketing Charts, τελευταία μέρα επίσκεψης 21/1/11
20. <http://dev.mysql.com/doc/refman/5.5/en/index.html>, MySQL version 5.5 Reference Manual, τελευταία μέρα επίσκεψης 19/1/11

21. "Comparing Images using Joint Histograms", G. Pass, R. Zabih, *Multimedia Systems*, Vol. 75, No 1, pp 73-85, 2000
22. http://en.wikipedia.org/wiki/Recommender_system, Recommender Systems, τελευταία μέρα επίσκεψης 11/1/11
23. http://en.wikipedia.org/wiki/Region_growing, Region Growing algorithms, τελευταία μέρα επίσκεψης 19/1/2011
24. http://en.wikipedia.org/wiki/RGB_color_model, RGB Color Model, τελευταία μέρα επίσκεψης 19/1/11
25. "Recommender Systems In E-Commerce", J. Ben Schafer, Joseph Konstan, John Riedl, *In proceedings of ACM E-Commerce 1999 conference*
26. <http://www.medialab.ntua.gr/education/MultimediaTechnology/MultimediaTechnologyNotes/index.htm>, η εικόνα, medialab, τελευταία μέρα επίσκεψης 19/1/11
27. Ψηφιακή Επεξεργασία & Ανάλυση Εικόνας, Νικόλαου Παπαμάρκου, Αθήνα 2005, Β. Γκιούρδας Εκδοτική
28. <http://el.wikipedia.org/wiki/RGB>, Το χρωματικό μοντέλο RGB, τελευταία μέρα επίσκεψης 19/1/11