



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ  
ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ**

**Παράλληλη επεξεργασία εικόνων υψηλής ανάλυσης σε σύστημα πολλών  
επεξεργαστικών πυρήνων  
για προβολή σε φορητές συσκευές - πελάτες**

**Μαυρίδης Ηλίας**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΥΠΕΥΘΥΝΟΣ  
Κακαρούντας Αθανάσιος**

Λαμία, 2012



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ  
ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ**

**Παράλληλη επεξεργασία εικόνων υψηλής ανάλυσης σε σύστημα  
πολλών επεξεργαστικών πυρήνων  
για προβολή σε φορητές συσκευές - πελάτες**

**Μαυρίδης Ηλίας**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΥΠΕΥΘΥΝΟΣ  
Κακαρούντας Αθανάσιος**

## Περίληψη

Οι φορητές συσκευές έχουν πλέον μετατραπεί σε προσωπικές πολύ-λειτουργικές συσκευές πολυμέσων. Ωστόσο, αν και αυτές οι συσκευές εκτελούν με αποτελεσματικότητα πολλές λειτουργίες, έχουν κάποια βασικά μειονεκτήματα όπως η περιορισμένη υπολογιστική ισχύ, οι λιγοστοί ενεργειακοί πόροι, η μικρή οθόνη. Αυτές οι αδυναμίες γίνονται εμφανείς όταν η συσκευή αναλαμβάνει να εκτελέσει υπολογιστικά απαιτητικούς αλγορίθμους, όπως οι αλγόριθμοι επεξεργασίας εικόνας και να προβάλλει εικόνες υψηλής ανάλυσης.

Η παρούσα εργασία έχει σκοπό να αντιμετωπίσει αυτά τα προβλήματα και να παρέχει ένα εύχρηστο περιβάλλον για επεξεργασία εικόνων. Θα γίνει ανάπτυξη ενός συστήματος πελάτη-εξυπηρετητή, το οποίο θα παρέχει τη δυνατότητα επεξεργασίας εικόνων υψηλής ανάλυσης μέσω διαδικτύου και την προβολή σε φορητές συσκευές αλλά και σε κάθε συσκευή που διαθέτει φυλλομετρητή. Η επεξεργασία των εικόνων θα γίνεται σε εξυπηρετητή που περιλαμβάνει προγραμματιζόμενο σύστημα πολλών πυρήνων· συγκεκριμένα θα μελετηθεί η χρήση μονάδας επεξεργασία γραφικών (GPGPU - *General Purpose computation on Graphics Processing Units*) ενώ η συσκευή του χρήστη θα αναλαμβάνει λειτουργίες μεταφοράς και προβολής των εικόνων.

Η σχεδίαση έχει πραγματοποιηθεί με βάση τα βήματα της Ανάπτυξης Λογισμικού, δηλαδή Ανάλυση Απαιτήσεων, Σχεδίαση και Υλοποίηση ενώ ο έλεγχος ήταν συνεχής σε όλα τα βήματα. Η υλοποίηση του συστήματος έχει πραγματοποιηθεί με βασικό γνώμονα τις παραμέτρους που έχουν προκαθοριστεί και έχει δοθεί έμφαση στη μείωση της κίνησης του δικτύου και στη μείωση του υπολογιστικού φόρτου στις φορητές συσκευές.

Για την αύξηση της ταχύτητας της εφαρμογής και της μείωσης της κίνησης του δικτύου χρησιμοποιήθηκαν μέθοδοι προεπεξεργασίας και επιλογής σημείων ενδιαφέροντος από τις εικόνες. Επιπλέον στο κομμάτι της ασφάλειας ο κάθε χρήστης προκειμένου να εισέλθει στο σύστημα θα περνά από το στάδιο της αυθεντικοποίησης, θα χρησιμοποιηθούν τεχνικές κρυπτογράφησης, ενώ θα γίνεται ροή πληροφοριών από τον εξυπηρετητή στον πελάτη χωρίς να μεταφέρεται η πληροφορία στο σύνολό της.

Όσον αφορά την επικοινωνία πελάτη με εξυπηρετητή, η ταχύτατη εξάπλωση των ασύρματων δικτύων και η ολοένα και αυξανόμενη ταχύτητα εγγυάται την πρόσβαση στον εξυπηρετητή και την ικανοποιητική ταχύτητα μεταφοράς δεδομένων.

### **Λέξεις-Κλειδιά**

Επεξεργασία εικόνας, Παράλληλη επεξεργασία, Φορητές συσκευές, Υπηρεσίες διαδικτύου, έξυπνες συσκευές, διαμοιρασμός αρχείων, ασφάλεια

# Abstract

Nowadays mobile devices have become multi-functional personal media devices. Although these devices perform effectively many functions, have some major disadvantages such as limited computing power, few energy resources, small screen. These weaknesses become apparent, when the device performs computational demanding algorithms such as image processing algorithms and display high resolution images.

This study aims to address these problems and provides a friendly environment for image processing. Will be developed an internet based client-server system, for high resolution image processing for portable devices and any device with internet browser. The image processing will be take place on server side which is being fitted with graphics processing unit (GPGPU - General Purpose computation on Graphics Processing Units), on the other hand user's device will perform transfer and viewing image functions.

The design has been done according to the steps of software development that includes Requirements Analysis, Design and Implementation with monitoring in all steps. The system implementation is based on parameters that are predefined and emphasized the reducing of network traffic and the computational load on mobile devices.

To increase the running speed of the system and for network traffic reduction have been used pre-processing methods and selection of regions of interest from images. For security each user has to pass authentication process to enter the system, also will be used encryption techniques and the information will send from server to client like data flow, without transferring entire image.

On client-server communication, the rapid spread of wireless networks and the increasing speed guarantees the access to server and high speed of data transfer.

## Key-Words

Image processing, Parallel processing, Mobile devices, Internet services, smart devices, file sharing, security

## Ευχαριστίες

Με αυτή την εργασία ολοκληρώνεται ο κύκλος των προπτυχιακών σπουδών, και νιώθω την ανάγκη να εκφράσω την ευγνωμοσύνη μου προς όλους εκείνους που με βοήθησαν.

Κατ' αρχάς να ευχαριστήσω τον επιβλέποντα καθηγητή, κύριο Κακαρούντα Αθανάσιο, για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου αυτή την εργασία, για την εποικοδομητική συνεργασία, και τις ουσιαστικές του υποδείξεις.

Επιπλέον θα ήθελα να ευχαριστήσω ιδιαίτερα την οικογένειά μου για την υποστήριξη, την αμέριστη συμπαράσταση, και την κατανόηση που επέδειξαν κατά την εκπόνηση της πτυχιακής εργασίας.

Τέλος, θέλω να ευχαριστήσω τους φίλους, τους συμφοιτητές, αλλά και τους καθηγητές του Πανεπιστημίου Στερεάς Ελλάδος, για τις γνώσεις αλλά και τις εμπειρίες που μου χάρισαν αυτά τα τέσσερα χρόνια που παρεπιδημούσα στη πόλη της Λαμίας.

# Περιεχόμενα

<b>1 ΕΙΣΑΓΩΓΗ.....</b>	<b>1</b>
1.1 ΓΕΝΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ, ΑΝΑΓΚΕΣ.....	1
1.2 ΣΚΟΠΟΣ.....	2
1.3 ΣΤΟΧΟΣ.....	3
<b>2 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ.....</b>	<b>5</b>
2.1 ΕΙΣΑΓΩΓΗ.....	5
2.2 ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ.....	5
2.3 ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ.....	7
2.4 ΠΙΘΑΝΑ ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ.....	8
<b>3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....</b>	<b>10</b>
3.1 ΚΕΝΤΡΙΚΟΠΟΙΗΜΕΝΗ ΔΟΜΗ ΠΕΛΑΤΗ-ΕΞΥΠΗΡΕΤΗΤΗ.....	10
3.1.1 Τα στοιχεία του συστήματος.....	10
3.1.2 Τύποι των εξυπηρετητών.....	11
3.1.3 Αξιοποίηση της αρχιτεκτονικής πελάτη-εξυπηρετητή.....	17
3.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΠΙΚΟΙΝΩΝΙΑΣ ΣΥΣΚΕΥΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	18
3.3 ΜΕΤΑΔΟΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ.....	19
3.4 ΧΡΗΣΗ ΜΟΝΑΔΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΓΡΑΦΙΚΩΝ ΓΙΑ ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ.....	21
<b>4 ΣΧΕΔΙΑΣΗ.....</b>	<b>26</b>
4.1 ΕΙΣΑΓΩΓΗ.....	26
4.2 ΑΡΧΕΣ ΣΧΕΔΙΑΣΜΟΥ ΔΙΕΠΑΦΗΣ ΧΡΗΣΤΗ.....	26
4.2.1 Γενικές αρχές σχεδιασμού χρηστικότητα.....	27
4.2.2 Ειδικές αρχές σχεδιασμού.....	28
4.3 ΣΧΕΔΙΑΣΜΟΣ ΣΕΛΙΔΑΣ.....	29
4.4 ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ.....	31
4.5 ΔΙΑΓΡΑΜΜΑ ΧΡΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	37
4.6 ΔΙΑΜΟΙΡΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΠΕΛΑΤΗ-ΕΞΥΠΗΡΕΤΗΤΗ.....	40
4.6.1 Λειτουργίες που θα εκτελεί η συσκευή του χρήστη.....	40
4.6.2 Λειτουργίες που θα εκτελεί ο εξυπηρετητής.....	41
<b>5 ΥΛΟΠΟΙΗΣΗ.....</b>	<b>44</b>
5.1 ΕΙΣΑΓΩΓΗ.....	44
5.2 ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ.....	44
5.2.1 Αλγόριθμος αλλαγής φωτεινότητας.....	46
5.2.2 Αλγόριθμος αλλαγής αντίθεσης.....	48
5.2.3 Αλγόριθμος αλλαγής μεγέθους.....	51
5.2.4 Αλγόριθμος μετακίνησης εικόνας.....	52
5.2.5 Αλγόριθμος εύρεσης ακμών εικόνας.....	53
5.2.6 Αλγόριθμος χρωματικής αντιστροφής εικόνας.....	56

5.3	ΦΟΡΜΕΣ ΤΕΛΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ .....	57
5.4	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ .....	71
5.5	ΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΣΥΣΤΗΜΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ .....	72
5.5.1	<i>Κάρτα Γραφικών Nvidia Geforce 9500 Gt .....</i>	73
5.5.2	<i>Επεξεργαστής Intel Pentium D 930.....</i>	73
<b>6 ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....</b>		<b>74</b>
6.1	ΑΞΙΟΛΟΓΗΣΗ .....	74
6.2	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	75
6.3	ΜΕΛΛΟΝΤΙΚΑ .....	75
<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>		<b>77</b>
A.1	ΕΙΣΑΓΩΓΗ.....	77
A.2	ΟΡΓΑΝΩΣΗ ΤΩΝ ΕΚΤΕΛΕΣΙΜΩΝ ΝΗΜΑΤΩΝ .....	77
A.3	ΠΥΡΗΝΕΣ-ΕΠΕΞΕΡΓΑΣΤΕΣ ΤΗΣ ΚΑΡΤΑΣ ΓΡΑΦΙΚΩΝ .....	80
A.4	ΕΚΤΕΛΕΣΗ ΤΩΝ ΝΗΜΑΤΩΝ .....	82
A.5	ΘΕΣΕΙΣ ΜΝΗΜΗΣ ΤΗΣ ΚΑΡΤΑΣ .....	83
A.5.1	<i>Global Memory.....</i>	86
A.5.2	<i>Shared Memory .....</i>	86
A.5.3	<i>Local Memory .....</i>	91
A.5.4	<i>Registers .....</i>	92
A.5.5	<i>Constant Memory .....</i>	92
A.5.6	<i>Texture Memory .....</i>	93
<b>ΠΑΡΑΡΤΗΜΑ Β.....</b>		<b>94</b>
B.1	ΕΙΣΑΓΩΓΗ.....	94
B.2	ΠΡΟΕΚΤΑΣΗ ΤΗΣ C .....	95
B.2.1	<i>Προσδιοριστές Υπορουτινών.....</i>	96
B.2.2	<i>Προσδιοριστές Μεταβλητών.....</i>	97
B.2.3	<i>Προσδιοριστικά Κλήσης Υπορουτίνας.....</i>	98
B.2.4	<i>Μεταβλητές Θέσης και Διαστάσεων.....</i>	98
B.3	ΠΡΟΣΘΗΚΕΣ ΒΙΒΛΙΟΘΗΚΗΣ ΕΚΤΕΛΕΣΙΜΩΝ ΕΝΤΟΛΩΝ.....	99
B.3.1	<i>Βιβλιοθήκη Κοινών Εντολών.....</i>	99
B.3.2	<i>Βιβλιοθήκη Εντολών Συσκευής.....</i>	99
B.3.3	<i>Βιβλιοθήκη Εντολών Συσκευής.....</i>	100
<b>ΠΑΡΑΡΤΗΜΑ Γ.....</b>		<b>101</b>
<b>ΑΝΑΦΟΡΕΣ.....</b>		<b>106</b>



# ΣΧΗΜΑΤΑ

Σχήμα 1: Αρχιτεκτονική δύο στρωμάτων [25] .....	15
Σχήμα 2: Αρχιτεκτονική τριών στρωμάτων [25] .....	17
Σχήμα 3: Αρχιτεκτονική του συστήματος .....	18
Σχήμα 4: Διαφορετικοί τύποι εικόνας κατά τη μεταφορά και επεξεργασία εικόνων ..	21
Σχήμα 5: Εξέλιξη των υπολογιστικών δυνατοτήτων, κεντρικής μονάδας επεξεργασίας και μονάδας επεξεργασία γραφικών [26] .....	23
Σχήμα 6: Αρχιτεκτονική κεντρική μονάδα επεξεργασίας (CPU) και μονάδας επεξεργασίας γραφικών (GPU) [19] .....	24
Σχήμα 7: Εύρος ζώνης μεταφοράς δεδομένων στη μνήμη (memory bandwidth) [26]	25
Σχήμα 8: Σχεδίαση του πρώτου προτύπου ιστοσελίδας .....	30
Σχήμα 9: Εικόνα του δεύτερου προτύπου ιστοσελίδας .....	31
Σχήμα 10: Διάγραμμα ροής δεδομένων Επιπέδου 0 .....	33
Σχήμα 11: Διάγραμμα ροής δεδομένων Επιπέδου 1 .....	34
Σχήμα 12: Διάγραμμα ροής δεδομένων Επιπέδου 2 για τη διεργασία 0. 5 του Επιπέδου 1 .....	36
Σχήμα 13: Διάγραμμα ροής δεδομένων Επιπέδου 2 για τη διεργασία 0.7 του Επιπέδου 1 .....	37
Σχήμα 14: Διάγραμμα περίπτωσης χρήσης διαχειριστή .....	38
Σχήμα 15: Διάγραμμα περίπτωσης χρήσης χρήστη .....	39
Σχήμα 16: Επιλογή περιοχής ενδιαφέροντος από εικόνα προς επεξεργασία [12] .....	43
Σχήμα 17: Διαδικασία εκτέλεσης προγραμμάτων [27] .....	45
Σχήμα 18: Μεγέθυνση εικόνας (a). Με γραμμική παρεμβολή (b) και με παρεμβολή γειτονικού εικονοστοιχείου (c) [30] .....	52
Σχήμα 19: Αρχική σελίδα εισαγωγής στο σύστημα .....	57
Σχήμα 20: Σελίδα εγγραφής χρήστη στο σύστημα .....	58
Σχήμα 21: Σελίδα με βασικό μενού .....	59
Σχήμα 22: Σελίδα αποστολής αρχείου εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος .....	60
Σχήμα 23: Αριστερή εικόνα, σελίδα επιλογής εικόνας για προβολή. Δεξιά εικόνα, σελίδα προβολής εικόνας που επιλέχθηκε από τον χρήστη .....	61
Σχήμα 24: Σελίδα επεξεργασίας εικόνας .....	62
Σχήμα 25: Σελίδα επιλογής εικόνας προς επεξεργασία, από κατάλογο .....	63
Σχήμα 26: Σελίδα επεξήγησης των εικονιδίων επεξεργασίας εικόνας στον χρήστη ..	64
Σχήμα 27: Σελίδα προβολής και επεξεργασίας της επιλεγμένης εικόνας. Με προβολή του αναδύομένου μενού (δεξιά εικόνα) και με απόκρυψη του αναδύομένου μενού (αριστερή εικόνα) .....	65
Σχήμα 28: Αριστερή εικόνα, σελίδα αλλαγής μεγέθους εικόνας. Δεξιά εικόνα, σελίδα μετακίνησης εικόνας .....	67
Σχήμα 29: Αριστερή εικόνα, σελίδα αλλαγής φωτεινότητας εικόνας. Δεξιά εικόνα, σελίδα αλλαγής αντίθεσης εικόνας .....	68
Σχήμα 30: Αριστερή εικόνα, σελίδα εύρεσης ακμών εικόνας. Δεξιά εικόνα, σελίδα χρωματικής αντιστροφής εικόνας .....	68
Σχήμα 31: Σελίδα επικοινωνίας με τους διαχειριστές του συστήματος .....	69
Σχήμα 32: Περιβάλλον phpmyadmin .....	71
Σχήμα 33: Πίνακας με χρήστες .....	71
Σχήμα 34: Πίνακας χρήστη .....	72

Σχήμα 35: Πυρήνες κεντρικής μονάδας επεξεργασίας και μονάδας επεξεργασίας γραφικών [28] .....	78
Σχήμα 36: ομαδοποίηση και οργάνωση των νημάτων [29].....	79
Σχήμα 37: Το μοντέλο των επεξεργαστών [19].....	81
Σχήμα 38: Λειτουργία DRAM [29] .....	83
Σχήμα 39: Το Μοντέλο της Μνήμης [29].....	84
Σχήμα 40: Η Χρήση της κάθε Μνήμης [19] .....	85
Σχήμα 41: Χρήση Κοινής Μνήμης [29] .....	87
Σχήμα 42: Στα αριστερά έχουμε ομαλή γραμμική καταχώρηση μιας λέξης για κάθε ένα νήμα σε κάθε ένα bank, ενώ στα δεξιά έχουμε ομαλή τυχαία καταχώρηση μιας λέξης για κάθε ένα νήμα σε κάθε ένα bank [19].....	88
Σχήμα 43: Ομαλή γραμμική καταχώρηση 3 λέξεων για κάθε νήμα σε θέσεις μνήμης [19].....	89
Σχήμα 44: - Γραμμική καταχώρηση δύο λέξεων ανά νήμα που προκαλεί σύγκρουση διευθύνσεων 2η τάξης και γραμμική καταχώρηση 8 λέξεων ανά νήμα που προκαλεί σύγκρουση διευθύνσεων 8ης τάξης [19] .....	90
Σχήμα 45: Στο αριστερό σχήμα οι καταχωρήσεις της μνήμης γίνονται χωρίς σύγκρουση, μιας και όλα τα νήματα διαβάζουν από την ίδια θέση μνήμης με χρήση αναμετάδοσης, ενώ στο δεξί δεν θα προκληθεί σύγκρουση αν η λέξη στο bank 5 είναι αυτή που θα αναμεταδοθεί, αλλά σε αντίθετη περίπτωση θα υπάρξει σύγκρουση 2ης τάξης [19].....	91
Σχήμα 46: Πακέτο Λογισμικού CUDA [19].....	95

## Πίνακες

Πίνακας 1: Τα τέσσερα κύρια στοιχεία των διαγραμμάτων ροής δεδομένων.....	32
Πίνακας 2: Αντιστοιχία ιστογραμμάτων με εικόνες διαφορετικής φωτεινότητας .....	46
Πίνακας 3: Ψευδοκόδικας και η υλοποίηση σε cuda της συνάρτησης αλλαγής φωτεινότητας .....	48
Πίνακας 4: Αντιστοιχία ιστογραμμάτων με εικόνες διαφορετικής αντίθεσης .....	49
Πίνακας 5: Ψευδοκόδικας και η υλοποίηση σε cuda της συνάρτησης αλλαγής αντίθεσης.....	51
Πίνακας 6: Εικόνες που προέκυψαν από τις μεθόδους ανίχνευσης ακμών (β)Prewitt, (γ)Roberts, (δ)Sobel, (ε)Kirsch [15] .....	53
Πίνακας 7: Ψευδοκόδικας και η υλοποίηση σε cuda της μεθόδου Sobel .....	55
Πίνακας 8: Ψευδοκόδικας και η υλοποίηση σε cuda της συνάρτησης χρωματικής αντιστροφής .....	56
Πίνακας 9: Υποστηριζόμενοι τύποι εικόνας.....	101

# Κεφάλαιο 1

## Εισαγωγή

---

1.1 Γενικές απαιτήσεις, ανάγκες

1.2 Σκοπός

1.3 Στόχος

---

### 1.1 Γενικές απαιτήσεις, ανάγκες

Το σύστημα αυτό αναπτύχθηκε με σκοπό να καλύψει τις ανάγκες για ασφαλή, γρήγορη και αξιόπιστη διαμοίραση και επεξεργασία εικόνων υψηλής ανάλυσης, από οποιαδήποτε συσκευή που συνδέεται στο διαδίκτυο και διαθέτει φυλλομετρητή. Απευθύνεται κυρίως στην αγορά του φορητού υπολογισμού (mobile computing) και δευτερευόντως του διάχυτου υπολογισμού (Pervasive Computing), φιλοδοξώντας να αντιμετωπίσει τα προβλήματα που προκύπτουν από τα ιδιαίτερα χαρακτηριστικά των φορητών συσκευών.

Ιδιαίτερη έμφαση δόθηκε στην ανάγκη για συμβατότητα του συστήματος με τους διάφορους τύπους εικόνας που υπάρχουν. Υποστηρίζοντας παραπάνω από εκατόν ενενήντα τύπους εικόνας, καταφέρνει να καλύπτει ένα ευρύ φάσμα εικόνων που προκύπτουν από διάφορες εφαρμογές. Π.χ. η υποστήριξη του τύπου DICOM [22] (Digital Imaging and Communications in Medicine) που χρησιμοποιείται στην ιατρική απεικόνιση, δίνει τη δυνατότητα στο χρήστη να επεξεργαστεί εικόνες από ιατρικούς ηλεκτρονικούς φακέλους ή εικόνες που προέκυψαν απευθείας από ιατρικά απεικονιστικά μηχανήματα.

Καθώς αυτή η εφαρμογή θα εξυπηρετεί κυρίως φορητές συσκευές, προκειμένου να είναι λειτουργική, απαιτείται η ταχύτητα επεξεργασίας των δεδομένων να είναι υψηλή, τα αποτελέσματα να έχουν τη βέλτιστη απεικόνιση για τη

κάθε συσκευή και να γίνεται οικονομία χρήσης υπολογιστικών και ενεργειακών πόρων. Επίσης η ταχύτητα επεξεργασίας θα πρέπει να είναι ανεξάρτητη της υπολογιστικής ισχύς των φορητών συσκευών και σταθερή για κάθε συσκευή. Το μέγεθος της οθόνης των φορητών συσκευών, δημιουργεί τις προϋποθέσεις για τη βέλτιστη αξιοποίηση των υπολογιστικών πόρων. Για αυτό το λόγο κατά την επεξεργασία και προβολή των εικόνων, θα πρέπει να λαμβάνονται υπόψη τα χαρακτηριστικά της οθόνης της συσκευής και να παράγονται αποτελέσματα για τη συγκεκριμένη οθόνη.

Στο κομμάτι της ασφάλειας το σύστημα θα πρέπει να εγγυάται την ασφάλεια των δεδομένων που βρίσκονται αποθηκευμένα στους εξυπηρετητές του συστήματος, την ασφαλή μεταφορά αυτών μέσω διαδικτύου και την διασφάλιση πρόσβασης στο σύστημα μόνο από εξουσιοδοτημένους χρήστες. Καθώς το σύστημα μπορεί να χρησιμοποιηθεί για την επεξεργασία εικόνων που περιέχουν ευαίσθητα δεδομένα (π.χ. ακτινολογική εξέταση από ηλεκτρονικό ιατρικό φάκελο ασθενούς) η σχεδίαση του συστήματος θα γίνει με τέτοιο τρόπο ώστε ο χρήστης να λαμβάνει μια συνεχόμενη ροή δεδομένων χωρίς να λαμβάνει στο σύνολο όλη τη πληροφορία. Επιπλέον απαιτούνται τεχνικές κρυπτογράφησης του συνθηματικού αλλά και του περιεχομένου της ιστοσελίδας που αποτελεί τη διεπαφή χρήστη.

Παραπάνω περιγράφηκαν οι βασικές ανάγκες που καλείται να καλύψει το σύστημα. Στην συνέχεια του ίδιου κεφαλαίου θα περιγραφούν ο σκοπός, ο στόχος του και γενικά σενάρια χρήσης του συστήματος.

## **1.2 Σκοπός**

Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη ενός συστήματος διαμοιρασμού και επεξεργασίας εικόνας μέσω διαδικτύου. Η δημιουργία αυτού του συστήματος έγινε κυρίως για να προσφέρει στους χρήστες των φορητών συσκευών τη δυνατότητα να επεξεργαστούν γρήγορα και με ασφάλεια τις εικόνες τους, ανεξαρτήτως του υλικού και λογισμικού της συσκευής τους.

Η κύρια λειτουργία του συστήματος θα είναι η επεξεργασία των εικόνων, ωστόσο θα αναπτυχθούν και άλλες υπηρεσίες προκειμένου να δίνουν τη δυνατότητα στο χρήστη να έχει γρήγορη πρόσβαση στις εικόνες του, με απλό τρόπο και από

παντού. Το σύστημα θα προσφέρει τη δυνατότητα αποθήκευσης, προβολής και επεξεργασίας όλων των ευρέως διαδεδομένων τύπων εικόνας και θα μπορεί να χρησιμοποιηθεί για διάφορες εφαρμογές και σε συνεργασία με άλλα συστήματα π.χ. για την επεξεργασία απεικονιστικών εξετάσεων από κάποιο νοσοκομείο ή γιατρό.

Οι λειτουργίες που θα εκτελούνται στη συσκευή του χρήστη θα είναι απλές και ανέξοδες σε ενεργειακούς και υπολογιστικούς πόρους ενώ τον κύριο υπολογιστικό φόρτο θα το αναλαμβάνει ο εξυπηρετητής. Το σύστημα θα αξιοποιεί με το βέλτιστο τρόπο τα χαρακτηριστικά της οθόνης της κάθε συσκευής για τη προβολή των εικόνων. Η ιστοσελίδα η οποία θα αποτελεί τη διεπαφή του χρήστη με το σύστημα, θα σχεδιαστεί και θα υλοποιηθεί με τέτοιο τρόπο, ώστε το περιεχόμενο να είναι διακριτό με μεγάλου μεγέθους επιλογές τις οποίες θα μπορεί ο χρήστης να επιλέξει-αγγίξει χωρίς δυσκολία από την οθόνη αφής της συσκευής του.

Αν και το σύστημα θα αναπτυχθεί με τέτοιο τρόπο ώστε να λαμβάνει υπόψη του και να αξιοποιεί τα χαρακτηριστικά των φορητών συσκευών, θα είναι πλήρως προσβάσιμο και λειτουργικό και από άλλες συσκευές που συνδέονται στο διαδίκτυο και διαθέτουν φυλλομετρητή, όπως οι σταθεροί και φορητοί υπολογιστές, ανεξαρτήτως κατασκευαστή και λειτουργικού συστήματος.

### **1.3 Στόχος**

Το σύστημα που αναπτύχθηκε, θα δίνει τη δυνατότητα στους χρήστες φορητών συσκευών αλλά και άλλων υπολογιστικών συστημάτων που συνδέονται στο διαδίκτυο και διαθέτουν φυλλομετρητή να επεξεργάζονται εικόνες. Το σύστημα όπως προαναφέρθηκε είναι βασισμένο στη δομή πελάτη-εξυπηρετητή. Ο εξυπηρετητής ο οποίος θα περιλαμβάνει προγραμματιζόμενο σύστημα πολλών πυρήνων ως τμήμα της αρχιτεκτονικής του, θα αναλαμβάνει την επεξεργασία των εικόνων. Η συσκευή του χρήστη θα εκτελεί απλές λειτουργίες προβολής των εικόνων και επικοινωνίας με τον εξυπηρετητή. Η δομή του συστήματος και η χρήση τεχνικών προεπεξεργασίας θα συμβάλουν στην προβολή των εικόνων υψηλής ανάλυσης σε σχεδόν πραγματικό χρόνο, με μόνη καθυστέρηση αυτής της επικοινωνίας εξυπηρετητή-πελάτη.

Απώτερος στόχος είναι η αξιοποίηση των χαρακτηριστικών των χρησιμοποιούμενων συσκευών ώστε να βελτιστοποιείται η ποιότητα της εικόνας

ώστε να ικανοποιεί τις προδιαγραφές των συσκευών-στόχων. Με αυτό τον τρόπο αξιοποιούνται κατάλληλα οι συσκευές, ενώ δεν απαιτούνται γενικού σκοπού ενεργοβόρα και απαιτητικά σε υπολογιστική ισχύ προγράμματα να εγκατασταθούν στις φορητές συσκευές. Αυτή η προσέγγιση αυξάνει σημαντικά τη συμβατότητα της προτεινόμενης λύσης και σε παλιότερες συσκευές, λιγότερο ικανές για επεξεργασία εικόνων υψηλής ποιότητας.

Ειδικός στόχος, είναι η αξιολόγηση της προσέγγισης σε ένα ιδιαίτερα κοινό πρόβλημα στο χώρο της ιατρικής, που αφορά το διαμοιρασμό πληροφορίας από τον ιατρικό φάκελο ενός ασθενή. Η προσέγγιση δίνει τη δυνατότητα διαμοιρασμού πληροφορίας που αφορά τις ιατρικές εικόνες δίχως όμως να διαμοιράσει το αρχείο καθεαυτό, ενώ η απεικόνιση δεν περιορίζεται από πρότυπα που μπορεί να διαφοροποιούνται από χώρα σε χώρα ή από οργανισμό σε οργανισμό.

# Κεφάλαιο 2

## Ανάλυση απαιτήσεων

---

2.1 Εισαγωγή

2.2 Απαιτήσεις συστήματος

2.3 Απαιτήσεις χρηστών

2.4 Πιθανά σενάρια χρήσης

---

### 2.1 Εισαγωγή

Πρωταρχικό βήμα στην ανάπτυξη λογισμικού είναι η ανάλυση των απαιτήσεων [1], [2]. Σε αυτό το στάδιο καταγράφονται και αναλύονται οι ανάγκες των χρηστών, τις οποίες καλείται να καλύψει το σύστημα. Οι απαιτήσεις διαχωρίζονται στις απαιτήσεις συστήματος και απαιτήσεις χρήστη.

### 2.2 Απαιτήσεις συστήματος

Οι απαιτήσεις του συστήματος αφορούν τις υπηρεσίες που πρέπει να παρέχει το σύστημα. Αυτές οι απαιτήσεις μπορούν να διαχωριστούν σε λειτουργικές και μη λειτουργικές. Οι λειτουργικές απαιτήσεις περιγράφουν τις εργασίες (λειτουργίες) που θα πρέπει να εκτελεί το σύστημα και καθορίζουν πως θα αντιδρά το σύστημα σε συγκεκριμένες εισόδους και τη συμπεριφορά του συστήματος σε συγκεκριμένες καταστάσεις. Οι μη λειτουργικές απαιτήσεις περιγράφουν τους περιορισμούς στις υπηρεσίες και λειτουργίες του συστήματος. Καθορίζουν την επίδοση του συστήματος, τα πρότυπα κτλ.



## **Λειτουργικές απαιτήσεις**

- Προσωπικός λογαριασμός για κάθε χρήστη.
- Αποστολή και αποθήκευση εικόνων υψηλής ανάλυσης στη βάση του συστήματος.
- Προβολή των εικόνων στον χρήστη ο οποίος τις έχει αποστείλει.
- Διαγραφή εικόνας από το λογαριασμό του χρήστη.
- Επεξεργασία των εικόνων.
- Επικοινωνία χρηστών με τους διαχειριστές του συστήματος.
- Διαθεσιμότητα του διαδικτυακού χώρου σε πολλούς χρήστες ταυτόχρονα.
- Άντληση στατιστικών στοιχείων για τον αριθμό των επισκεπτών της διαδικτυακής εφαρμογής.
- Δημιουργία αντιγράφων ασφαλείας (back-up) των αρχείων.
- Πρόσβαση μέσω διαδικτύου. Το μόνο λογισμικό που χρειάζεται από την πλευρά του χρήστη, είτε διαχειριστή είτε επισκέπτη, είναι ένα πρόγραμμα περιήγησης.
- Υψηλή μεταφερσιμότητα του συστήματος, ώστε το σύστημα να λειτουργεί ομαλά ανεξάρτητα από την πλατφόρμα.

## **Μη λειτουργικές απαιτήσεις**

- Χρόνος απόκρισης σε περίπτωση επεξεργασίας λιγότερος από 5 δευτερόλεπτα.
- Μικρός όγκος κάθε ιστοσελίδας, ώστε να γίνεται γρήγορα η φόρτωσή της.
- Έλεγχος της ασφάλειας με διαφύλαξη της ακεραιότητας, για τη διαφύλαξη και πληρότητα των πληροφοριών της εφαρμογής από πιθανές διαδικτυακές επιθέσεις.
- Οι διεπαφές να είναι απλές και να αξιοποιούν τις οθόνες αφής.
- Το περιβάλλον της εφαρμογής να είναι ελκυστικό και εύχρηστο, ώστε να κεντρίζει το ενδιαφέρον του χρήστη, αλλά να μην τον αποπροσανατολίζει.
- Περιοχές της οθόνης να διατηρούν σταθερή σχεδίαση, ώστε να μην υπάρχει σύγχυση από την πλευρά του χρήστη.

- Σε κάθε βήμα της πλοήγησης, να υπάρχουν τα απαραίτητα μενού για πρόσβαση στην αρχική σελίδα.
- Ο χρήστης να γνωρίζει κάθε στιγμή σε ποια σελίδα βρίσκεται.

## 2.3 Απαιτήσεις χρηστών

Αυτό το στάδιο περιλαμβάνει την κατανόηση και καταγραφή των στόχων των χρηστών και των εργασιών που χρειάζεται να εκτελέσουν. Ακολουθεί ανάλυση απαιτήσεων για κάθε κατηγορία χρηστών.

### Απαιτήσεις Διαχειριστή

- Δυνατότητα συντήρησης και αναβάθμισης του συστήματος.
- Διαχείριση της βάσης δεδομένων του συστήματος που είναι αποθηκευμένα όλα τα δεδομένα.
- Διαχείριση συστήματος μέσω προγράμματος πλοήγησης χωρίς την ανάγκη εγκατάστασης επιπρόσθετου λογισμικού.
- Εύκολος τρόπος επιλογής και επεξεργασίας δεδομένων.
- Διαχείριση και κατηγοριοποίηση περιεχομένου, ώστε να διαμορφώνουν την εμφάνιση του ιστοχώρου (π.χ. μενού) και να τον καθιστούν πιο εύχρηστο.
- Δημιουργία λογαριασμών για νέους χρήστες.
- Διατήρηση των στοιχείων επικοινωνίας για παροχή βοήθειας.

### Απαιτήσεις Επισκέπτη

- Προσπέλαση του συστήματος μέσω προγράμματος πλοήγησης χωρίς την ανάγκη εγκατάστασης επιπρόσθετου λογισμικού.
- Αποστολή και αποθήκευση εικόνων στη βάση του συστήματος
- Προβολή των αποθηκευμένων εικόνων και διαγραφή των εικόνων που ο κάθε χρήστης έχει αποστείλει στη βάση.

- Επεξεργασία των εικόνων που περιλαμβάνει μεγέθυνση και σμίκρυνση της εικόνας, μετακίνηση της εικόνας, αλλαγή της φωτεινότητας, αλλαγή της αντίθεσης, εύρεση ακμών, χρωματική αντιστροφή της εικόνας.
- Δυνατότητα επικοινωνίας με το διαχειριστή (πιθανόν για παροχή βοήθειας ή αναφορά προβλήματος).

## 2.4 Πιθανά σενάρια χρήσης

Το σύστημα θα προσφέρει υπηρεσίες αποθήκευσης, προβολής και επεξεργασίας εικόνων υψηλής ανάλυσης. Η πρόσβαση στο σύστημα γίνεται μέσω διαδικτύου, για αυτό η συσκευή του χρήστη πρέπει να διαθέτει σύνδεση στο διαδίκτυο και να έχει εγκατεστημένο κάποιο πρόγραμμα φυλλομετρητή.

Ο χρήστης για να αποκτήσει πρόσβαση στο σύστημα αρχικά θα πρέπει να εγγραφεί στην υπηρεσία. Αφού εγκριθεί η αίτησή του από τους διαχειριστές του συστήματος θα μπορεί να επισκεφτεί την ιστοσελίδα του συστήματος και αφού περάσει το στάδιο της αυθεντικοποίησης θα εισέρχεται στο κύριο μενού, από το οποίο θα μπορεί να επιλέξει μία από τις λειτουργίες. Αυτές οι λειτουργίες θα περιλαμβάνουν την αποστολή μιας εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος, την εύρεση μιας εικόνας από τη βάση, την διαγραφή εικόνας από τη βάση, την επιλογή και επεξεργασία μιας από τις εικόνες της βάσης και τέλος την επικοινωνία με τους διαχειριστές του συστήματος.

Επιλέγοντας την αποστολή μιας εικόνας στη βάση, ο χρήστης θα επιλέγει την εικόνα που θέλει από τη συσκευή του και στη συνέχεια η εικόνα θα μεταφέρεται στη βάση του συστήματος. Ο κάθε χρήστης θα έχει πρόσβαση στις εικόνες τις οποίες έχει αποστείλει μόνο αυτός. Στην συνέχεια θα μπορεί να δει τις εικόνες που έχει στο λογαριασμό του και θα μπορεί να διαγράψει όποια από αυτές επιθυμεί.

Η κύρια λειτουργία του συστήματος είναι η επεξεργασία εικόνων. Ο χρήστης επιλέγοντας αυτή τη λειτουργία θα καλείται να επιλέξει ένα αρχείο εικόνας για μετέπειτα επεξεργασία. Αφού επιλέξει την εικόνα που θέλει τότε θα επιλέγει το είδος της επεξεργασίας που θέλει να εφαρμόσει. Οι επεξεργασίες που θα μπορεί να εκτελέσει είναι η μεγέθυνση και σμίκρυνση της εικόνας, μετακίνηση της εικόνας, αλλαγή της φωτεινότητας, αλλαγή της αντίθεσης, εύρεση ακμών, χρωματική

αντιστροφή της εικόνας ενώ σε αυτό το μενού θα έχει επίσης την επιλογή για προβολή της αρχικής εικόνας χωρίς επεξεργασία και την επιλογή εισαγωγής νέας εικόνας προς επεξεργασία.

Οι τελευταίες λειτουργίες που θα μπορεί να εκτελέσει είναι η επικοινωνία με τους διαχειριστές του συστήματος όπου ο χρήστης συμπληρώνοντας την ειδική φόρμα θα μπορεί να αποστείλει μήνυμα στους διαχειριστές του συστήματος για οποιαδήποτε παρατήρηση ή οτιδήποτε πρόβλημα αντιμετώπισε. Τέλος επιλέγοντας να αποσυνδεθεί από το σύστημα θα διαγράφονται όλα τα στοιχεία που προέκυψαν από την τελευταία αλληλεπίδρασή του με το σύστημα και θα του εμφανίζεται η αρχική σελίδα σύνδεσης στο σύστημα.

## Κεφάλαιο 3

### Αρχιτεκτονική

---

3.1 Κεντριοποιημένη δομή πελάτη-εξυπηρετητή

3.2 Χαρακτηριστικά επικοινωνίας συσκευής-συστήματος

3.3 Μετάδοση των δεδομένων

3.4 Χρήση μονάδας επεξεργασίας γραφικών για παράλληλη επεξεργασία

---

### 3.1 Κεντριοποιημένη δομή πελάτη-εξυπηρετητή

Στο μοντέλο πελάτη-εξυπηρετητή, ο πελάτης θέτει μια αίτηση και ο εξυπηρετητής επιστρέφει κάποια δεδομένα ή εκτελεί μια σειρά από ενέργειες και επιστρέφει τα παραγόμενα αποτελέσματα [3], [4], [5]. Ο εξυπηρετητής μπορεί να ανταποκρίνεται άμεσα στην αίτηση αυτή ή να την εκτελεί σε μεταγενέστερο χρόνο εάν προηγούνται άλλες αιτήσεις προς εκτέλεση.

Η δομή πελάτη-εξυπηρετητή είναι πολύ σημαντική, διότι προσφέρει στους χρήστες πρόσβαση σε πληροφορίες μέσω σταθερών και εύκολων στην χρήση διασυνδέσεων και αυξάνει την ευελιξία και τη δυνατότητα δημιουργίας συστημάτων που υποστηρίζουν πολλά περιβάλλοντα.

#### 3.1.1 Τα στοιχεία του συστήματος

##### Πελάτης

Ο πελάτης είναι ο αιτών των υπηρεσιών και είναι αυτός που ξεκινάει πάντα την επικοινωνία με τον εξυπηρετητή. Μια μηχανή πελάτη πρέπει να μπορεί να τρέχει το λογισμικό για τη δημιουργία των γραφικών διεπαφής χρήστη (GUIs). Να

προσφέρει μια εύκολη στη χρήση διασύνδεση χρηστών, να στέλνει αιτήσεις, να δέχεται αποκρίσεις του εξυπηρετητή. Επιπλέον πρέπει να επιτρέπει στον χρήστη να βλέπει και να χειρίζεται τις πληροφορίες.

### **Εξυπηρετητής**

Ο εξυπηρετητής απαντάει στις αιτήσεις που αποστέλλονται από τους πελάτες. Αν και ο πελάτης κατέχει το μεγαλύτερο μέρος της προσοχής του χρήστη, καθώς με τον πελάτη αλληλεπιδρούν οι χρήστες, ο εξυπηρετητής είναι η καρδιά του συστήματος πελάτη-εξυπηρετητή. Οι εξυπηρετητές είναι οι μηχανές όπου αποθηκεύονται οι πληροφορίες και εκτελούνται οι διεργασίες.

Τα συστατικά του εξυπηρετητή είναι απλά. Μια μηχανή εξυπηρετητή πρέπει να μπορεί να αποθηκεύει, να ανακτά και να προστατεύει πληροφορίες. Να επιθεωρεί τις αιτήσεις των πελατών, να διαχειρίζεται πληροφορίες, να δημιουργεί εφαρμογές διαχείρισης πληροφοριών, όπως δημιουργία αντιγράφων ασφαλείας.

## **3.1.2 Τύποι των εξυπηρετητών**

Οι τύποι των εξυπηρετητών μπορούν να διαχωριστούν ανάλογα με τις υπηρεσίες που προσφέρουν αλλά και με το πώς η εφαρμογή κατανέμεται μεταξύ του πελάτη και του εξυπηρετητή.

### **3.1.2.1 Με βάση τις υπηρεσίες που προσφέρουν**

Με βάση τις υπηρεσίες που προσφέρουν μπορούν να διαχωριστούν σε πέντε κατηγορίες

- Εξυπηρετητής Εφαρμογών (Application server).
- Εξυπηρετητής Πληροφοριών (Data server).
- Εξυπηρετητής Υπολογισμών (Compute server).
- Εξυπηρετητής Βάσεων Δεδομένων (Database servers).

- Εξυπηρετητής Πόρων ή Επικοινωνιών (Resource or Communications server).

### **Εξυπηρετητής Εφαρμογών**

Οι εξυπηρετητές εφαρμογών (application servers) τρέχουν λογισμικό εφαρμογών, χρησιμοποιούνται όταν διανέμονται λογικές εφαρμογών μεταξύ του πελάτη και του εξυπηρετητή. Η εγκατάσταση εφαρμογών στον εξυπηρετητή δηλώνει ότι αυτές οι εφαρμογές θα είναι διαθέσιμες σε πολλούς πελάτες. Πολλοί πελάτες μπορούν να χρησιμοποιήσουν κλήσεις απομακρυσμένων διεργασιών (RPCs - Remote Procedure Calls) για να θέσουν σε λειτουργία μια επεξεργασία στον εξυπηρετητή. Πολλοί εξυπηρετητές εφαρμογών μπορούν ακόμα και να εργαστούν μαζί για να απαντήσουν στην αίτηση του πελάτη. Κάθε εξυπηρετητής μπορεί να τρέχει ένα διαφορετικό λειτουργικό σύστημα σε μια διαφορετική πλατφόρμα υλικού, ωστόσο αυτές οι λεπτομέρειες δεν αφορούν τον πελάτη ο οποίος μπορεί να κάνει αιτήσεις χωρίς να λαμβάνει υπόψη του τον τύπο της μηχανής που θα ανταποκριθεί.

### **Εξυπηρετητής Πληροφοριών**

Οι Εξυπηρετητές πληροφοριών (data servers) χρησιμοποιούνται για την αποθήκευση και διαχείριση πληροφοριών και χρησιμοποιούνται σε συνδυασμό με εξυπηρετητές υπολογισμών. Αυτοί οι εξυπηρετητές ερευνούν και ελέγχουν την αξιοπιστία των πληροφοριών, αλλά γενικά δεν μεταβιβάζουν μεγάλη ποσότητα πληροφοριών στο δίκτυο.

### **Εξυπηρετητής Υπολογισμών**

Οι εξυπηρετητές υπολογισμών (compute servers) εκτελούν τις αιτήσεις των πελατών για πληροφορίες στον εξυπηρετητή πληροφοριών και έπειτα προωθούν τα αποτελέσματα των αιτήσεων πίσω στον πελάτη.

### **Εξυπηρετητής Βάσεων Δεδομένων**

Οι εξυπηρετητές βάσεων δεδομένων (database servers) είναι τυπικά συστήματα πελάτη-εξυπηρετητή, και αναλαμβάνουν να εκτελέσουν την ίδια

εργασία με αυτή που κάνουν οι εξυπηρετητές πληροφοριών και υπολογισμών μαζί. Οι εξυπηρετητές βάσεων δεδομένων τρέχουν λογισμικό διαχείρισης βάσεων δεδομένων (DBMS - Database Management System). Το σύστημα διαχείρισης βάσεων δεδομένων προσφέρει υπηρεσίες ανάκτησης και διαχείρισης πληροφοριών. Οι εξυπηρετητές που συνδυάζουν τις λειτουργίες του εξυπηρετητή βάσεων δεδομένων και του εξυπηρετητή εφαρμογών είναι επίσης γνωστοί ως εξυπηρετητές συναλλαγών (transaction servers).

### **Εξυπηρετητής Πόρων ή Επικοινωνιών (Resource or Communication servers)**

Οι εξυπηρετητές πόρων (resource servers), περικλείουν τους εξυπηρετητές επικοινωνιών (communications servers) επιτρέπουν σε πολλούς πελάτες την προσπέλαση συγκεκριμένων πόρων, που είναι πολύ ακριβοί για να βρίσκονται σε έναν πελάτη. Για παράδειγμα, οι εξυπηρετητές εκτυπώσεων (print servers) συνδέουν πολλούς πελάτες με πολλούς εκτυπωτές. Οι εξυπηρετητές επικοινωνιών συνδέουν τα απομακρυσμένα συστήματα. Άλλοι εξυπηρετητές πόρων μπορούν να συνδέσουν πελάτες με άλλες συσκευές, όπως συσκευές πολυμέσων.

Ένας εύκολος τρόπος για να διαχωριστούν οι εξυπηρετητές εφαρμογών, βάσεων δεδομένων και συναλλαγών είναι το πώς ο πελάτης κάνει αιτήσεις στον εξυπηρετητή. Οι εξυπηρετητές δέχονται τους παρακάτω τύπους αιτήσεων από τους πελάτες:

- Οι εξυπηρετητές εφαρμογών ενεργούν κάπως πιο αποκεντρωμένα από τη βάση δεδομένων σε απάντηση του πελάτη.
- Οι εξυπηρετητές βάσεων δεδομένων επιστρέφουν πληροφορία σαν απάντηση σε μια αίτηση του πελάτη, που γίνεται σε SQL.
- Οι εξυπηρετητές συναλλαγών επιστρέφουν πληροφορία σαν απάντηση σε ένα μήνυμα που αποτελείται από ένα σύνολο ερωτημάτων SQL. Αυτό το σύνολο επιτυγχάνει ή αποτυγχάνει σαν μια μονάδα.



### **3.1.2.2 Με βάση τη κατανομή της εφαρμογής μεταξύ του πελάτη και του εξυπηρετητή**

#### **Ισχυροί εξυπηρετητές και ισχυροί πελάτες (fat servers and fat clients)**

Εκτός από τον διαχωρισμό ανάλογα με τις υπηρεσίες που προσφέρουν, οι εφαρμογές πελάτη-εξυπηρετητή μπορούν να διακριθούν και με βάση το πώς η εφαρμογή μεταξύ του πελάτη και του εξυπηρετητή. Το μοντέλο του ισχυρού εξυπηρετητή προσδίδει περισσότερες λειτουργίες στον εξυπηρετητή. Το μοντέλο του ισχυρού πελάτη προσδίδει περισσότερες λειτουργίες στον πελάτη. Οι εξυπηρετητές διαδικτύου (Web servers) είναι παραδείγματα ισχυρών εξυπηρετητών.

Ο ισχυρός πελάτης είναι ο πιο παραδοσιακός τύπος συστήματος πελάτη-εξυπηρετητή. Προσφέρουν ευλυγισία και ευκαιρίες για δημιουργία εργαλείων που επιτρέπουν στους τελικούς χρήστες να δημιουργήσουν τις δικές τους εφαρμογές.

Οι εφαρμογές των ισχυρών εξυπηρετητών είναι πιο εύκολο να διαχειρίζονται και να αναπτύσσονται στο δίκτυο διότι το μεγαλύτερο μέρος του κώδικα τρέχει στους εξυπηρετητές. Οι ισχυροί εξυπηρετητές προσπαθούν να ελαχιστοποιήσουν τις ανταλλαγές στο δίκτυο δημιουργώντας πιο ουσιώδη επίπεδα υπηρεσιών. Ο πελάτης στο μοντέλο του ισχυρού πελάτη προσφέρει το γραφικό περιβάλλον χρήστη και αλληλεπιδρά με τον εξυπηρετητή μέσω των κλήσεων απομακρυσμένων διεργασιών.

#### **Αρχιτεκτονικές δύο στρωμάτων (Two-tier) και τριών στρωμάτων (Three-tier) πελάτη-εξυπηρετητή**

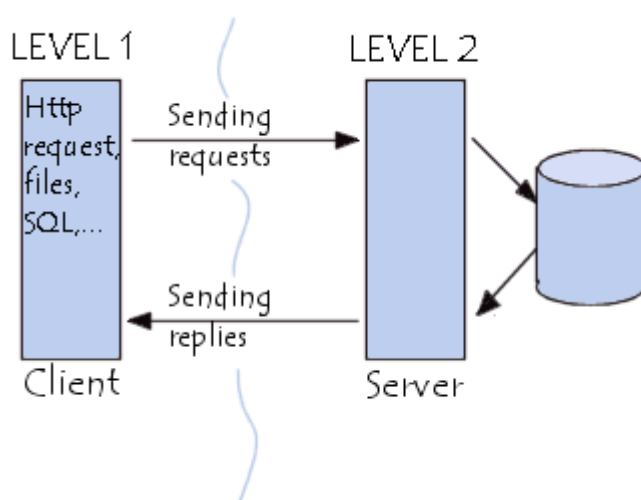
Πολλές φορές προτιμούνται να χρησιμοποιούνται όροι, όπως δύο στρωμάτων, τριών στρωμάτων αρχιτεκτονικές για τα συστήματα πελάτη-εξυπηρετητή αντί των όρων ισχυροί πελάτες και ισχυροί εξυπηρετητές. Ουσιαστικά αυτοί οι όροι βασίζονται στην ίδια βασική ιδέα. Έχουν να κάνουν με το πώς διαιρείται η εφαρμογή πελάτη-εξυπηρετητή σε λειτουργικές ενότητες, οι οποίες αναθέτονται είτε στον πελάτη, είτε σε έναν ή περισσότερους εξυπηρετητές.

### Αρχιτεκτονική δύο στρωμάτων

Οι εφαρμογές πελάτη-εξυπηρετητή πρώτης γενιάς εκτελούνταν γενικά με δυο λογικά στρώματα (Σχήμα 1). Αυτό το μοντέλο έχει συχνά δυο στρώματα υλικού. Το σύστημα πελάτη-εξυπηρετητή δύο στρωμάτων διαιρεί την εφαρμογή σε δυο συγκεκριμένα τμήματα (τα στρώματα), όπου ένα τμήμα τρέχει στον πελάτη και ένα ξεχωριστό τμήμα τρέχει στον εξυπηρετητή.

Η ποσότητα της λογικής εφαρμογής που λειτουργεί στον πελάτη ή στον εξυπηρετητή καθορίζει εάν αυτό είναι αδύνατο ή ισχυρό. Το αδύνατο υποδηλώνει ότι παρουσιάζεται μικρή ανάπτυξη της εφαρμογής και το ισχυρό ότι παρουσιάζεται ένα μεγάλο τμήμα της εφαρμογής. Υπάρχουν ποικίλες διαβαθμίσεις αδυνάτου και ισχυρού. Οι αδύνατοι πελάτες είναι ελκυστικοί όταν ο πελάτης υπολογιστής έχει περιορισμένη ισχύ, όπως όταν οι πελάτες είναι φορητές συσκευές.

Τα συστήματα πελάτη-εξυπηρετητή με δύο στρώματα εμφανίζονται να είναι πιο δύσκολο να αναπτυχθούν και να συντηρηθούν από το προσδοκώμενο. Οι εφαρμογές δύο στρωμάτων δεν κλιμακώνουν καλά. Επίσης, τα εργαλεία για την ανάπτυξη τέτοιων συστημάτων όπως η Visual Basic, η Delphi και το PowerBuilder πήραν χρόνο για να αναπτυχθούν και τα περισσότερα από αυτά τα συνεχίζουν να αναπτύσσονται για να υποστηρίζουν αρχιτεκτονική τριών στρωμάτων.



Σχήμα 1: Αρχιτεκτονική δύο στρωμάτων [25]

### Αρχιτεκτονική τριών στρωμάτων

Ο πιο πρόσφατος τύπος πελάτη-εξυπηρετητή που αναπτύσσεται είναι ο τύπος τριών στρωμάτων (Σχήμα 2). Μια πρωταρχική διαφορά μεταξύ των δυστρωματικών και τριστρωματικών εφαρμογών είναι το επιπλέον επίπεδο λογισμικού στον εξυπηρετητή. Οι εφαρμογές δύο στρωμάτων τείνουν να τοποθετούν την 'λογική' στον πελάτη και να περνούν εγγραφές στη βάση δεδομένων (ισχυρό μοντέλο πελάτη) ή να περνούν δεδομένα στη βάση δεδομένων, όπου αποθηκευμένες διαδικασίες εκτελούν την 'λογική' της εφαρμογής (αδύνατο μοντέλο πελάτη), οι εφαρμογές τριών στρωμάτων τείνουν να περνούν μηνύματα μεταξύ των τμημάτων της εφαρμογής του πελάτη και του εξυπηρετητή. Το τμήμα του εξυπηρετητή εφαρμόζει τη 'λογική' της εφαρμογής, κατόπιν την στέλνει στη βάση δεδομένων. Η 'λογική' της εφαρμογής συνήθως καλείται «Business Rules» στο χώρο της αρχιτεκτονικής πελάτη-εξυπηρετητή.

Η αρχιτεκτονική πελάτη-εξυπηρετητή τριών στρωμάτων προσθέτει κάποια πολυπλοκότητα, διότι πρόκειται για ένα επιπρόσθετο κομμάτι κώδικα που αναπτύσσεται. Τα εργαλεία και οι γλώσσες προγραμματισμού που χρησιμοποιούνται στον κώδικα του εξυπηρετητή εξαρτώνται από την πλατφόρμα του εξυπηρετητή.

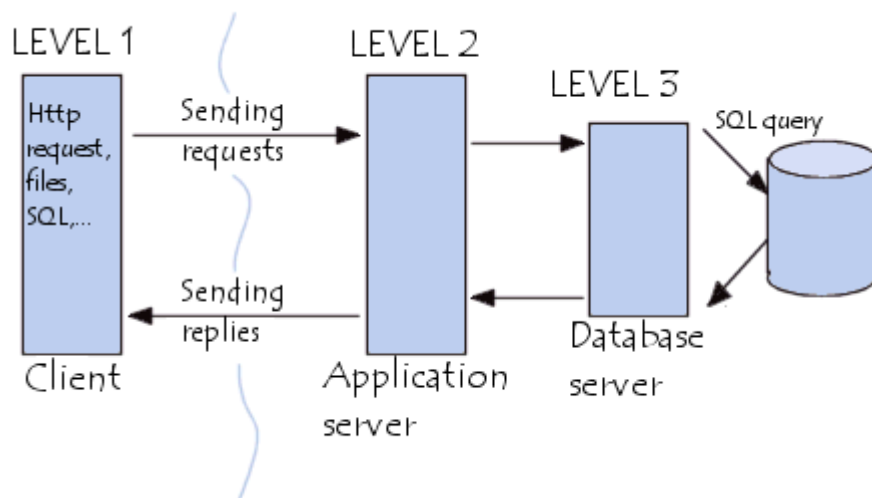
Το τμήμα του εξυπηρετητή της αρχιτεκτονικής τριών στρωμάτων αυξάνει την πολυπλοκότητα της εφαρμογής. Ωστόσο υπάρχουν ορισμένα πλεονεκτήματα σε αυτή τη προσέγγιση. Αυτά περιλαμβάνουν την κλιμάκωση, τα γενικότερα λιγότερα προβλήματα στα δίκτυα διανομής και την ευλυγισία.

Η κλιμάκωση βελτιώθηκε, διότι ο εξυπηρετητής και η βάση δεδομένων είναι διαχωρισμένα, μπορούν να ξεκινήσουν από ένα απλό «υπολογιστή-οικοδεσπότη» και αργότερα να χωριστούν. Πολλαπλές εφαρμογές εξυπηρετητή μπορούν να επικοινωνήσουν με μια κεντρική βάση δεδομένων ή μια εφαρμογή εξυπηρετητή μπορεί ακόμα να εξυπηρετήσει τους πελάτες, ενώ προσπελάζονται οι πολλαπλές βάσεις δεδομένων.

Η ευελιξία κερδίζεται, διότι ο πελάτης, ο εξυπηρετητής και τα συστήματα βάσεων δεδομένων μπορούν το καθένα να αντικατασταθούν χωρίς να επηρεάζουν τα άλλα κομμάτια, δεδομένου ότι η διασύνδεση επίσης δεν αλλάζει. Για παράδειγμα, μετατρέποντας τη βάση δεδομένων από Sybase σε Oracle επηρεάζεται μόνο το τμήμα του εξυπηρετητή της εφαρμογής, όχι ο πελάτης. Το να

ξαναδιατυπώνεις έναν πελάτη από Visual Basic σε Delphi δεν έχει επίδραση στο υπόλοιπο τμήμα της εφαρμογής, δεδομένου ότι υπάρχει αλληλεπίδραση ανάμεσα στον κώδικα του πελάτη και τον κώδικα του εξυπηρετητή.

Πολλές από τις σημερινές εμπορικές εφαρμογές, που βασίζονται σε βάσεις δεδομένων, συμπεριλαμβανόμενου και του SAP R/3 [6], χρησιμοποιούν το μοντέλο πελάτη-εξυπηρετητή τριών στρωμάτων για να κερδίσουν τη δυνατότητα αυξομείωσης και ευλυγισίας. Επίσης οι εφαρμογές διαδικτύου είναι γενικά τριών στρωμάτων πελάτη-εξυπηρετητή εφαρμογές.



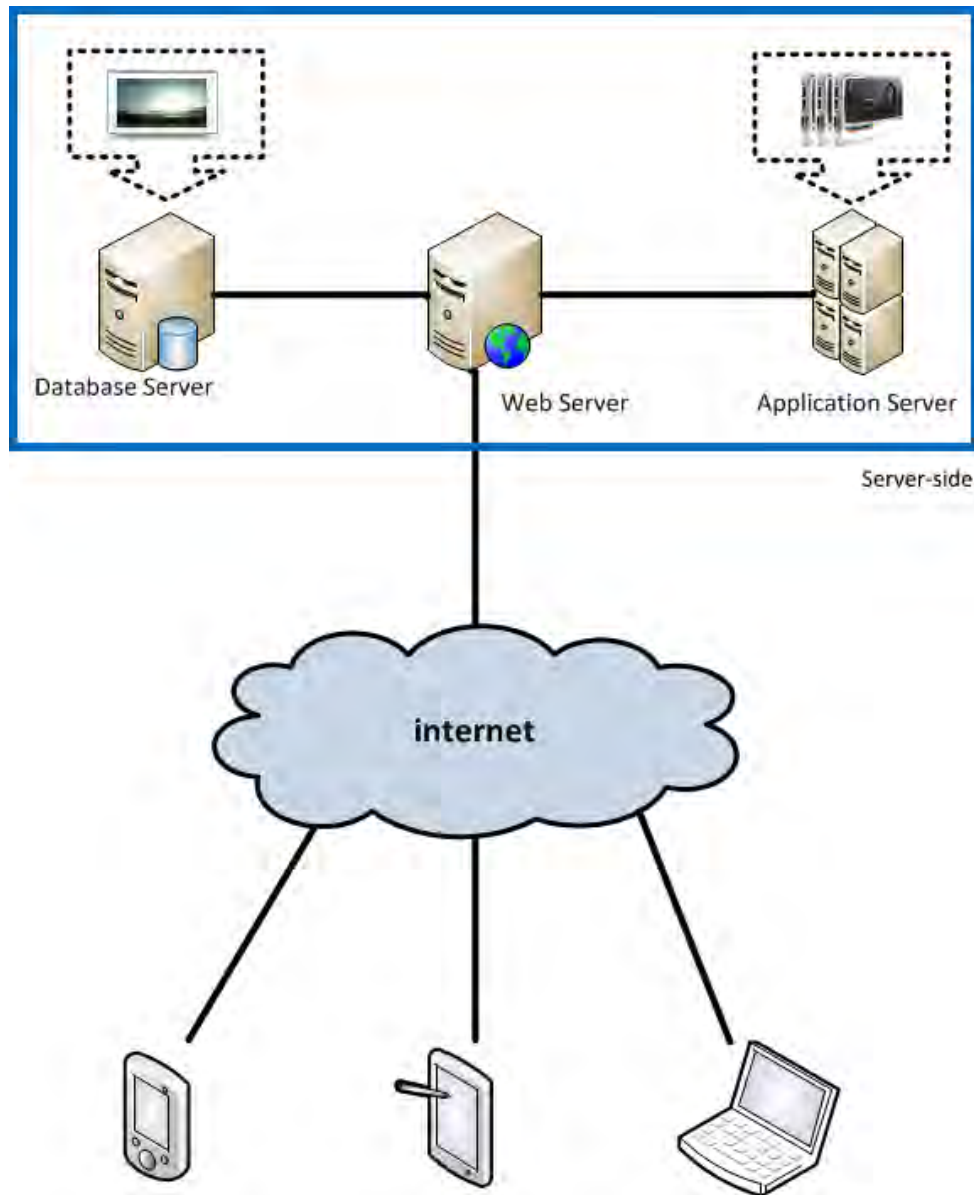
Σχήμα 2: Αρχιτεκτονική τριών στρωμάτων [25]

### 3.1.3 Αξιοποίηση της αρχιτεκτονικής πελάτη-εξυπηρετητή

Με βάση τους παραπάνω τύπους που περιγράφηκαν, το σύστημα που θα αναπτυχθεί θα αποτελείται από τρεις εξυπηρετητές, όπως φαίνεται και στο Σχήμα 3 ο πρώτος για τη διαμοίραση του περιεχομένου της σελίδας μέσω μιας σύνδεσης https (web server), ο δεύτερος για την εξυπηρέτηση των ερωτημάτων και απαντήσεων από και προς της βάση του συστήματος (database server) και ο τρίτος για την επεξεργασία των δεδομένων (application server).

Όσον αφορά τη κατανομή του φόρτου εργασίας μεταξύ του πελάτη και εξυπηρετητή στο σύστημα, τον κύριο όγκο επεξεργασίας τον αναλαμβάνει ο εξυπηρετητής επεξεργασίας δεδομένων, ενώ οι πελάτες- φορητές συσκευές θα εκτελούν απλές λειτουργίες προβολής των εικόνων και αποστολή ερωτημάτων στον

εξυπηρετητή, έτσι με βάση αυτή τη κατανομή το σύστημα μπορεί να χαρακτηριστεί ότι έχει ισχυρό εξυπηρετητή και αδύναμο πελάτη.



Σχήμα 3: Αρχιτεκτονική του συστήματος

### 3.2 Χαρακτηριστικά επικοινωνίας συσκευής συστήματος

Η επικοινωνία του χρήστη με το σύστημα, απαιτεί τη σύνδεση της συσκευής του χρήστη στο διαδίκτυο. Καθώς το σύστημα θα χρησιμοποιείται κυρίως από φορητές συσκευές, η σύνδεση στο διαδίκτυο θα γίνεται μέσω κάποιου ασύρματου τοπικού δικτύου (WLAN) ή μέσω δικτύου κινητής τηλεφωνίας. Η στοίβα

πρωτοκόλλων που θα χρησιμοποιηθεί είναι το tcp/ip. Το πρωτόκολλο επικοινωνίας θα είναι το Https [11] ενώ δεν δημιουργείται ένας εξειδικευμένος πελάτης ανά συσκευή, αλλά μια κοινή υπηρεσία, διαθέσιμη από το Internet προκειμένου να μπορεί να χρησιμοποιηθεί από οποιαδήποτε συσκευή (παλαιά και νεότερη ή ακόμα και από όσες θα εμφανιστούν μελλοντικά).

Κατά την επικοινωνία της συσκευής του χρήστη με το σύστημα ανταλλάσσονται δεδομένα των οποίων ο κύριος όγκος περιλαμβάνει δεδομένα εικόνων, που αποστέλλονται από τον εξυπηρετητή στη συσκευή του χρήστη. Μεταφορά εικόνων από τη συσκευή του χρήστη στο σύστημα γίνεται μόνο κατά την αποστολή εικόνας από τη συσκευή στη βάση του συστήματος, αντιθέτως μεταφορά εικόνων από το σύστημα στη συσκευή του χρήστη γίνεται σε όλη τη διάρκεια επεξεργασίας των εικόνων. Οι εικόνες που αποστέλλονται στη συσκευή του χρήστη, έχουν προκύψει μετά από μεθόδους προ-επεξεργασίας σε προγενέστερο χρόνο. Οι επεξεργασμένες εικόνες αποτελούν ένα στιγμιότυπο της αρχικής εικόνας του χρήστη που είναι αποθηκευμένη τη βάση του συστήματος, προκύπτουν με βάση τα χαρακτηριστικά της συσκευής του χρήστη και τη πλοήγηση του χρήστη στο σύστημα.

Ο χρήστης επισκεπτόμενος την ιστοσελίδα του συστήματος επικοινωνεί με τους τρεις εξυπηρετητές του συστήματος για τις διάφορες λειτουργίες, για την προβολή της ιστοσελίδας, την αποστολή και λήψη δεδομένων από βάση του συστήματος και την εκτέλεση μεθόδων προ-επεξεργασίας των εικόνων.

### **3.3 Μετάδοση των δεδομένων**

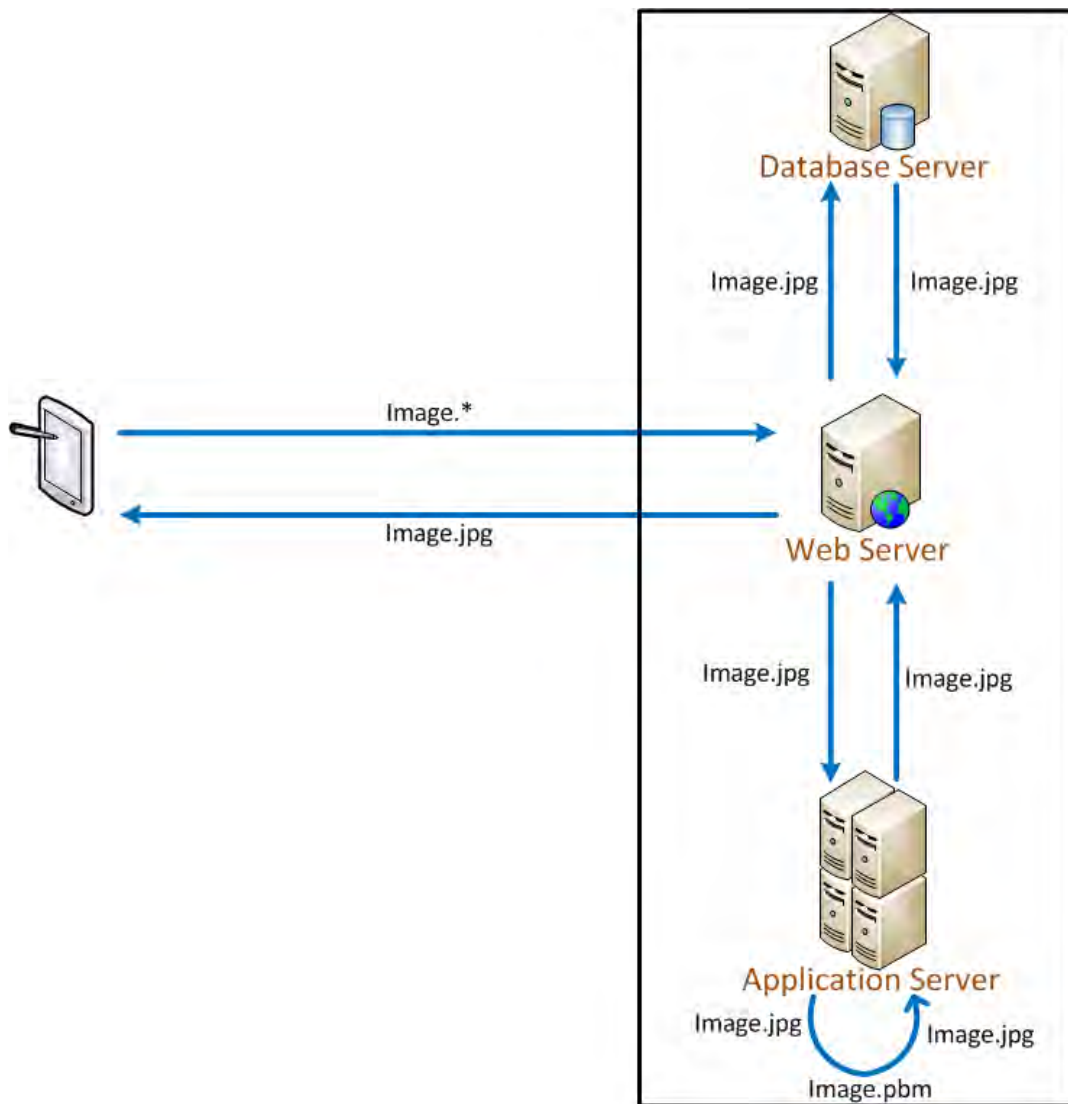
Κατά τη λειτουργία του συστήματος μεταφέρονται δεδομένα από το σύστημα στο χρήστη και το αντίστροφο, ενώ επίσης και στο εσωτερικό του συστήματος υπάρχει μεταφορά των δεδομένων μεταξύ του εξυπηρετητή που βρίσκεται η βάση δεδομένων, του εξυπηρετητή επεξεργασίας δεδομένων και του εξυπηρετητή που περιέχει την ιστοσελίδα. Αυτά τα δεδομένα είναι κυρίως αρχεία εικόνων και μηνύματα για την επίτευξη της επικοινωνίας.

Το σύστημα επιτρέπει στον χρήστη να εισάγει και να επεξεργαστεί περισσότερους από 140 τύπους εικόνων. Οι εικόνες αυτές προκειμένου να είναι διαχειρίσιμες και να μπορούν να προβληθούν από κάθε φυλλομετρητή, πρέπει να

μετατραπούν σε ενδιάμεσο τύπο εικόνας που να υποστηρίζεται από το φυλλομετρητή. Μετά από έρευνα διαπιστώθηκε ότι ο τύπος εικόνας που υποστηρίζει το σύνολο των φυλλομετρητών είναι το JPEG (.jpg), έτσι οι εικόνες που θα λαμβάνει ο χρήστης και θα προβάλλονται στη συσκευή του θα πρέπει να είναι αυτού του τύπου. Χρήσιμο κρίθηκε οι εικόνες να αποθηκεύονται και στη βάση με αυτή τη μορφή προκειμένου να είναι γρήγορα και εύκολα προσπελάσιμες από τον χρήστη, για αυτό το λόγο κατά την αποστολή μιας εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος γίνεται αυτόματα η μετατροπή και αποθήκευση των εικόνων σε jpg.

Ο εξυπηρετητής επεξεργασίας εικόνων δέχεται εικόνες jpg, παράγει εικόνες τύπου pbm (Portable Bit Map) και επιστρέφει εικόνες jpg όπως φαίνεται και στο Σχήμα 4. Ο τύπος εικόνας pbm δεν είναι μια μορφή που χρησιμοποιείται για την αποθήκευση ή την μετάδοση μιας εικόνας καθώς είναι πολύ δαπανηρός και δεν είναι αρκετά εκφραστικός για κάτι τέτοιο. Είναι απλώς ένας ενδιάμεσος τύπος εύκολα διαχειρίσιμος από το πρόγραμμα επεξεργασίας εικόνας και τις διαθέσιμες βιβλιοθήκες που παρέχει η Nvidia (θα γίνει αναφορά στην Nvidia σε άλλο σημείο της Πτυχιακής εργασίας) και χρησιμοποιούνται στη παράλληλη επεξεργασία. Η χρησιμοποίηση αυτού του ενδιάμεσου τύπου καθιστά εύκολη την διαχείριση των εικόνων από το πρόγραμμα αλλά λόγω των περιορισμών του τύπου pbm, οι εικόνες κατά την επεξεργασία μετατρέπονται σε ασπρόμαυρες και οι επεξεργασίες εφαρμόζονται πάνω στην ασπρόμαυρη εκδοχή της αρχικής εικόνας. Ωστόσο η εικόνα στη βάση του συστήματος παραμένει όπως ήταν αρχικά και ο χρήστης μπορεί να δει τις εικόνες έτσι όπως τις έχει αποστείλει.

Κατά την επεξεργασία μιας εικόνας όπως φαίνεται στο παρακάτω σχήμα γίνεται μεταφορά της εικόνας από τη βάση του συστήματος στον εξυπηρετητή επεξεργασίας και μετατροπή από jpg σε pbm. Έπειτα εκτελούνται διαδικασίες προ-επεξεργασίας για προϋπολογισμό των εικόνων που πιθανών θα ζητήσει ο χρήστης στο επόμενο βήμα. Αφού γίνει η επεξεργασία των εικόνων ακολουθεί μετατροπή των εικόνων από pbm σε jpg προκειμένου οι εικόνες να μπορούν να προβληθούν στο φυλλομετρητή του χρήστη. Στη συνέχεια μεταφέρονται από τον εξυπηρετητή στη συσκευή μέσω του διαδικτύου και προβάλλονται στη συσκευή του.



Σχήμα 4: Διαφορετικοί τύποι εικόνας κατά τη μεταφορά και επεξεργασία εικόνων

### 3.4 Χρήση μονάδας επεξεργασίας γραφικών για παράλληλη επεξεργασία

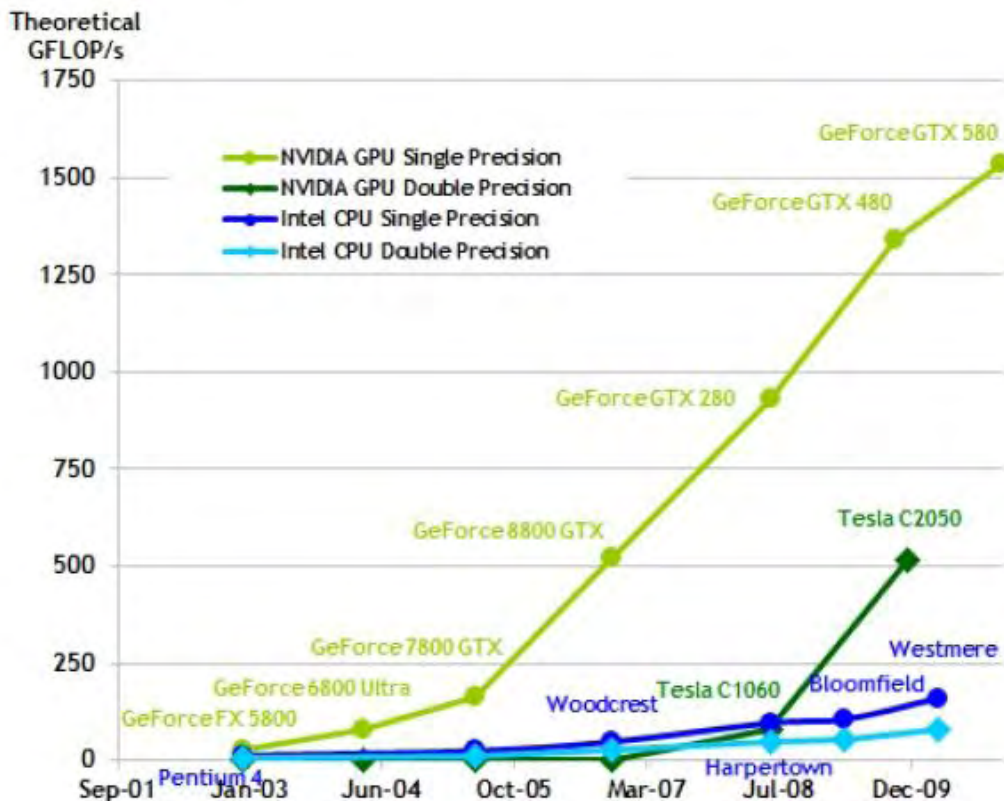
Προκειμένου το σύστημα να μπορεί να αντεπεξέρχεται στους υπολογιστικά απαιτητικούς αλγορίθμους επεξεργασίας εικόνας και στο μεγάλο όγκο των δεδομένων που πρόκειται να επεξεργαστεί πρέπει να χρησιμοποιηθεί μια ισχυρή και αποδοτική υπολογιστική μηχανή. Όπως προαναφέρθηκε ο εξυπηρετητής επεξεργασίας εικόνων διαθέτει και χρησιμοποιεί μονάδα επεξεργασίας γραφικών, για την εκτέλεση των αλγορίθμων σε παραλληλία.



Από το 2003, η βιομηχανία ημιαγωγών έχει επικεντρωθεί σε δύο κύριους δρόμους για το σχεδιασμό των μικροεπεξεργαστών [7]. Ο δρόμος των πολλαπλών πυρήνων (multicores) επιδιώκει να διατηρήσει την ταχύτητα εκτέλεσης των σειριακών προγραμμάτων ενώ μεταφέρεται σε πολλαπλούς πυρήνες. Οι επεξεργαστές πολλαπλών πυρήνων ξεκίνησαν ως διπύρηνοι, με τον αριθμό των επεξεργαστών να διπλασιάζεται σχεδόν σε κάθε νέα γενιά επεξεργαστών. Ένα τωρινό παράδειγμα είναι ο μικροεπεξεργαστής Intel Core i7, ο οποίος έχει τέσσερις πυρήνες, που είναι συμβατοί με τις εντολές της αρχιτεκτονικής x86, επιτρέποντας την εκτέλεση πολλαπλών εντολών εκτός σειράς. Ο μικροεπεξεργαστής υποστηρίζει υπερνημάτωση (hyperthreading) με δύο νήματα υλικού (hardware threads) και είναι σχεδιασμένος για τη μεγιστοποίηση της ταχύτητας επεξεργασίας των σειριακών προγραμμάτων.

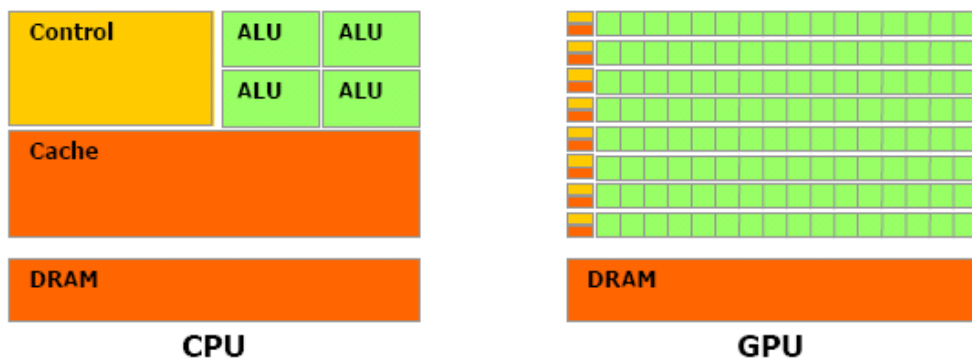
Σε αντίθεση ο δρόμος των πολυπύρηνων μικροεπεξεργαστών (many-core) επικεντρώνεται περισσότερο στην αποδοτική εκτέλεση των παράλληλων προγραμμάτων. Οι πολυπύρηνοι μικροεπεξεργαστές ξεκίνησαν ως ένας μεγάλος αριθμός πυρήνων που και πάλι με κάθε γενιά διπλασιάζεται ο αριθμός τους. Ένα τωρινό παράδειγμα αποτελεί η μονάδα επεξεργασίας γραφικών NVIDIA GeForce GTX 280 με διακόσιους σαράντα πυρήνες, ο κάθε ένας από τους οποίους είναι υπερνηματομένος και βρίσκεται σε σειρά με άλλους επτά πυρήνες, διαμοιραζόμενος μαζί τους τον έλεγχο και τη κρυφή μνήμη εντολής. Οι πολυπύρηνοι επεξεργαστές, ειδικά οι μονάδες επεξεργασίας γραφικών είναι πρωτοποριακοί στην κούρσα των αποδόσεων της κινητής υποδιαστολής από το 2003 (Σχήμα 5). Ενώ η βελτίωση της απόδοσης των γενικού σκοπού μικροεπεξεργαστών έχει καθυστερήσει σημαντικά, οι μονάδες επεξεργασίας γραφικών συνεχίζουν να βελτιώνονται ακατάπαυστα.

Από το 2009, η αναλογία μεταξύ πολυπύρηνων μονάδων επεξεργασίας γραφικών και επεξεργαστών πολλαπλών πυρήνων για τη μέγιστη απόδοση στις πράξεις κινητής υποδιαστολής είναι περίπου 10 προς 1 (Σχήμα 5). Αυτά τα αποτελέσματα προκύπτουν από τη ταχύτητα που μπορούν να υποστηρίξουν αυτοί οι μικροεπεξεργαστές 1 teraflop (1000 gigaflops) έναντι 100 gigaflops το 2009.



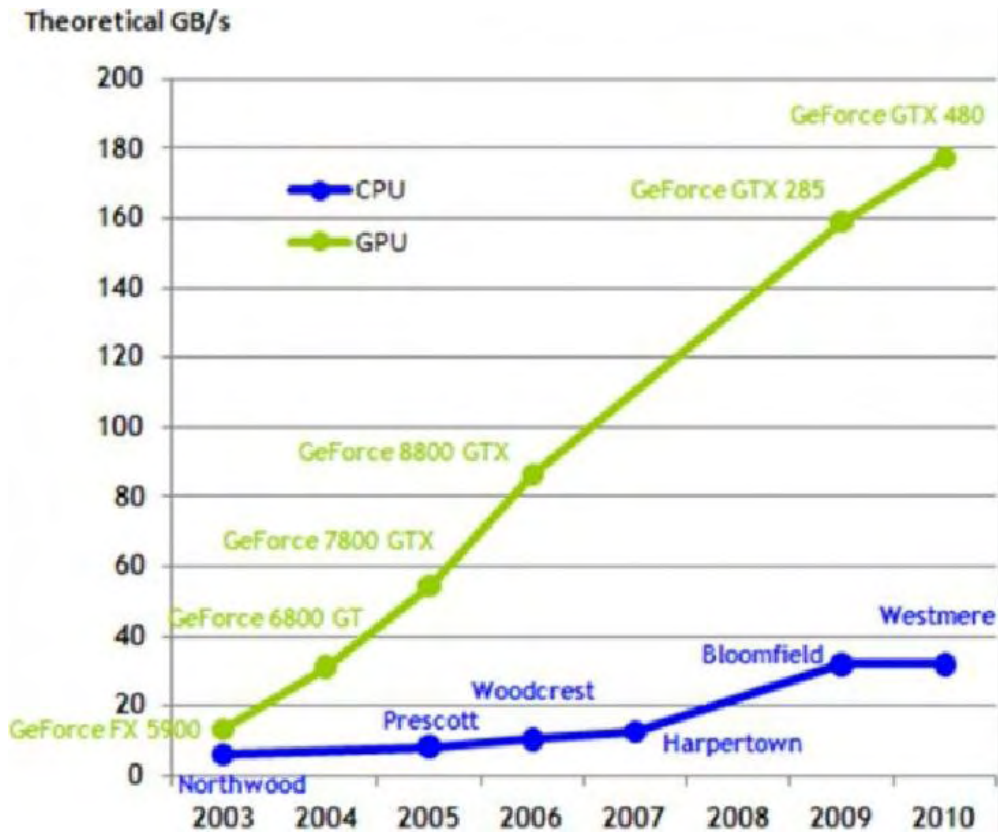
Σχήμα 5: Εξέλιξη των υπολογιστικών δυνατοτήτων, κεντρικής μονάδας επεξεργασίας και μονάδας επεξεργασία γραφικών [26]

Θα μπορούσε κανείς να αναρωτηθεί γιατί υπάρχει μια τέτοια μεγάλη διαφορά επιδόσεων μεταξύ πολυπύρηνων μονάδων επεξεργασίας γραφικών και κεντρικών μονάδων επεξεργασίας πολλαπλών πυρήνων. Η απάντηση βρίσκεται στις διαφορές ως προς το σχεδιασμό μεταξύ των δύο τύπων επεξεργαστών, όπως φαίνεται στο Σχήμα 6. Ο σχεδιασμός μιας κεντρική μονάδα επεξεργασίας έχει βελτιστοποιηθεί για την εκτέλεση σειριακού κώδικα. Χρησιμοποιεί εξελιγμένη λογική ελέγχου για να επιτρέψει τις οδηγίες εκτέλεσης ενός νήματος να εκτελεστούν παράλληλα, ή ακόμη και με διαφορετική προτεραιότητα από τη διαδοχική σειρά τους, διατηρώντας την εντύπωση της διαδοχικής εκτέλεσης. Σημαντικότερη είναι η χρήση μεγάλων προσωρινών μνημών (cache memory) για να μειωθεί η καθυστέρηση των μεγάλων πολύπλοκων εφαρμογών. Ούτε η λογική του ελέγχου ούτε οι προσωρινές μνήμες συνέβαλαν στην μεγιστοποίηση της ταχύτητας υπολογισμού. Από το 2009, οι νέοι μικροεπεξεργαστές γενικού σκοπού πολλαπλών πυρήνων έχουν συνήθως τέσσερις μεγάλους πυρήνες επεξεργασίας σχεδιασμένοι να προσφέρουν μεγάλη απόδοση εκτέλεσης για σειριακό κώδικα.



Σχήμα 6: Αρχιτεκτονική κεντρική μονάδα επεξεργασίας (CPU) και μονάδας επεξεργασίας γραφικών (GPU) [19]

Το εύρος ζώνης της μνήμης είναι ένα ακόμη σημαντικό ζήτημα. Οι μικροεπεξεργαστές γραφικών λειτουργούν με περίπου δέκα φορές μεγαλύτερο εύρος ζώνης σε σχέση με κεντρικές μονάδες επεξεργασίας όπως φαίνεται και στο Σχήμα 7. Στα τέλη του 2006, η GeForce 8800 GTX, ήταν σε θέση να μεταφέρει δεδομένα με περίπου 85gigabytes ανά δευτερόλεπτο (Gb / s), εκτός και εντός της κύριας δυναμικής μνήμης τυχαίας προσπέλασης (DRAM). Εξαιτίας των απαιτήσεων της μνήμης πλαισίου και του μη περιοριστικού μοντέλου μνήμης –δηλαδή ο τρόπος με τον οποίο διάφορα συστήματα λογισμικού, εφαρμογών και συσκευών εισόδου/εξόδου προσδοκούν να προσπελαίνουν τη μνήμη- οι επεξεργαστές γενικού σκοπού θα πρέπει να ικανοποιούν τις απαιτήσεις παλαιότερων λειτουργικών συστημάτων, εφαρμογών και συσκευών εισόδου/εξόδου, γεγονός που περιορίζει σημαντικά την αύξηση του εύρους ζώνης της μνήμης. Σε αντίθεση, με απλούστερα πρότυπα μνήμης, οι σχεδιαστές των γραφικών μονάδων επεξεργασίας μπορούν πιο εύκολα να επιτύχουν μεγαλύτερο εύρος ζώνης της μνήμης. Ο πιο πρόσφατος μικροεπεξεργαστής NVIDIA GT200 υποστηρίζει περίπου 150 GB/s. Το εύρος ζώνης μνήμης των μικροεπεξεργαστών πιθανότατα δεν θα αυξηθεί πέραν του 50 GB/s για περίπου 3 χρόνια, οπότε η κεντρική μονάδα επεξεργασίας θα συνεχίσει να είναι σε μειονεκτική θέση όσον αφορά το εύρος ζώνης μνήμης για κάποιο χρονικό διάστημα.



Σχήμα 7: Εύρος ζώνης μεταφοράς δεδομένων στη μνήμη (memory bandwidth) [26]

Με βάση τα παραπάνω και λαμβάνοντας υπόψη τη φύση των υπολογισμών που θα πραγματοποιούνται στον εξυπηρετητή η χρησιμοποίηση μονάδας επεξεργασίας γραφικών στην αρχιτεκτονική του συστήματος εγγυάται την υψηλή ταχύτητα επεξεργασίας των δεδομένων. Επιπλέον με αυτό τον τρόπο μειώνεται σε μεγάλο βαθμό η αναλογία κόστους προς απόδοσης, η οποία εξαρτάται από τον τύπο της μονάδας επεξεργασίας γραφικών που θα χρησιμοποιηθεί. Η αποτελεσματική αξιοποίηση των δυνατοτήτων των καρτών γραφικών απαιτεί την δημιουργία προγραμμάτων τα οποία εκτελούνται με παράλληλο τρόπο και αξιοποιούν τις δυνατότητες της κάρτα γραφικών. Σε αυτή την εργασία μελετήθηκε η χρήση της οικονομικής κάρτας nvidia 9500 gt 512 MB, ωστόσο τα προγράμματα που θα περιέχει το σύστημα θα λαμβάνουν υπόψη τις δυνατότητες της κάθε κάρτας και θα προσαρμόζονται ώστε να την αξιοποιούν βέλτιστα.

# Κεφάλαιο 4

## Σχεδίαση

---

4.1 Εισαγωγή

4.2 Αρχές σχεδιασμού διεπαφής χρήστη

4.3 Σχεδιασμός σελίδας

4.4 Διαγράμματα ροής δεδομένων

4.5 Διάγραμμα χρήσης συστήματος

4.6 Διαμοίραση λειτουργιών πελάτη-εξυπηρετητή

---

### 4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστεί η σχεδίαση του συστήματος. Σε αυτή τη φάση θα γίνει λεπτομερή σχεδίαση του κάθε τμήματος που καθορίστηκε στην αρχιτεκτονική του συστήματος.

### 4.2 Αρχές σχεδιασμού διεπαφής χρήστη

Για τη σχεδίαση διεπαφής χρήστη και ειδικότερα μιας ιστοσελίδας η οποία θα αποτελεί τη διεπαφή χρήστη, υπάρχουν κάποιες αρχές οι οποίες θα πρέπει να τηρούνται. Αυτές οι αρχές δεν πρέπει να υποκαθιστούν την ανάλυση απαιτήσεων, αλλά πρέπει να συνδυάζονται με αυτήν, προκειμένου να προχωρήσουμε στο σχεδιασμό της διεπαφής. Υπάρχουν γενικές και ειδικές αρχές σχεδιασμού οι οποίες θα αναλυθούν παρακάτω[20].

## 4.2.1 Γενικές αρχές σχεδιασμού χρηστικότητας

Στην ενότητα αυτή περιγράφονται οι σημαντικότερες και οι πιο συχνά χρησιμοποιούμενες αρχές [20]:

1. **Αρχή της σκιαγράφησης των εν δυνάμει χρηστών:** Πρέπει να είναι γνωστό ποιοι είναι οι εν δυνάμει χρήστες, τους στόχους τους, τις δυνατότητες, την εμπειρία τους, τις ανάγκες τους προκειμένου η εφαρμογή να σχεδιαστεί με βάση αυτά.
2. **Ομοιομορφία και Συνέπεια:** Είναι η ομοιότητα στη συμπεριφορά του συστήματος προς τον χρήστη. Δηλαδή αν η ίδια ενέργεια συντελείται σε δύο ή περισσότερα διαφορετικά μέρη της εφαρμογής, θα πρέπει να παρουσιάζεται και να λειτουργεί ακριβώς με τον ίδιο τρόπο σε όλα τα μέρη.
3. **Οικειότητα:** Η χρήση προηγούμενης γνώσεις του χρήστη από άλλες πραγματικές απαντήσεις ή υπολογιστικά συστήματα κατά τη διάδραση με ένα νέο σύστημα.
4. **Μινιμαλισμός:** Η χρήση όσο το δυνατόν λιγότερων και απλών στοιχείων ώστε η διεπαφή να μην προκαλεί σύγχυση στο χρήστη. Το σύστημα πρέπει να διατηρεί τη μέγιστη δυνατή αποδοτικότητα προβάλλοντας στον χρήστη τη λιγότερη δυνατή πληροφορία.
5. **Χρήση κατανοητής προς τους χρήστες γλώσσας:** Η χρήση ευκολονόητης γλώσσας, ορολογίας και συμβόλων ώστε να κάνει ευκολότερη τη χρήση του συστήματος από τους χρήστες.
6. **Αρχή της έκθεσης των λειτουργιών (feature exposure):** Η άμεση πληροφόρηση το χρήστη για όλες τις δυνατές λειτουργίες, του συστήματος.
7. **Αρχή εστίασης προσοχής:** Ο χρήστης εστιάζει περισσότερο σε κάποια στοιχεία της διεπαφής από κάποια άλλα. Για αυτό το λόγο τα σημαντικά στοιχεία της διεπαφής θα πρέπει να ξεχωρίζουν και να τραβούν τη προσοχή του χρήστη. Αυτό μπορεί να γίνει με διάφορους τρόπους όπως η μεγάλη αντίθεση μεταξύ του αντικειμένου και του φόντου της σελίδας, ή το διαφορετικό μέγεθος του αντικειμένου σχετικά με τα άλλα αντικείμενα της σελίδας.
8. **Αρχή της αισθητικής και της λεπτομέρειας:** Σύμφωνα με αυτή την αρχή, μια διαδικτυακή εφαρμογή προσελκύει τον χρήστη όταν είναι καλαίσθητη και καλοφτιαγμένη. Οι λεπτομέρειες, πολλές φορές, μπορούν να οδηγήσουν το

χρήστη να προβλέψει τη συμπεριφορά ενός στοιχείου ελέγχου. Ωστόσο υπάρχουν κάποιοι περιορισμοί, όπως η υπερβολική χρήση πολυμέσων μπορεί να προκαλέσει σύγχυση στους νέους χρήστες και να τους δώσει την εντύπωση ότι πρόκειται για μια δύσχρηστη εφαρμογή, ακόμα και αν δεν είναι. Επίσης τα πολλά πολυμέσα μπορεί να επιβραδύνουν την απόκριση του προγράμματος και να προκαλέσουν ενόχληση στους χρήστες.

9. **Αρχή της ασφάλειας:** Η διεπαφή χρήστη θα πρέπει να αναπτυχθεί ώστε να μειώνει την ανασφάλεια των χρηστών, προβλέποντας, προλαμβάνοντας και διορθώνοντας τα πιθανά λάθη των χρηστών.
10. **Αναίρεση ενεργειών:** Ο χρήστης θα πρέπει να έχει τη δυνατότητα να επιστρέψει στη προηγούμενη κατάσταση.

#### 4.2.2 Ειδικές αρχές σχεδιασμού

Αυτές οι ειδικές αρχές που θα περιγραφούν στη συνέχεια, κατατάσσονται στις ακόλουθες ανάγκες σχεδιασμού: ευκολία μάθησης, απλότητα, ευκολία χρήσης, ταχύτητα, χρήση εικονιδίων, ιδιαίτερη ανάγκη για την αρχή της ασφάλειας, συμμόρφωση με χρήση προτύπων [2], [20].

1. Το πάνω μέρος της οθόνης είναι το ιδανικό σημείο εμφάνισης του τίτλου της σελίδας.
2. Η ιεραρχία ενός ιστότοπου είναι σκόπιμο να είναι πιο «πλατιά» παρά «βαθιά» και αυτό για να είναι ευκολότερο στο χρήστη να περιηγηθεί στη σελίδα.
3. Με την τεχνική του eye-tracking, που δίνει στοιχεία και για το που κοιτάζει ο χρήστης στην οθόνη και άλλα σχετικά, έχει προκύψει το συμπέρασμα ότι οι χρήστες, συχνά, ενώ βρίσκονται σε ένα παράθυρο, πηγαίνουν σε ένα άλλο και μετά επιστρέφουν για να συνεχίσουν την ανάγνωση και πλοήγηση στο πρώτο. Αυτό σημαίνει ότι υπάρχει ιδιαίτερη ανάγκη βοήθειας προς τους χρήστες να ξαναβρίσκουν το δρόμο τους στον ιστοχώρο. Αυτό μπορεί να επιτευχθεί με περιγραφικούς και άμεσους τίτλους στις υποσελίδες, υπόδειξη της θέσης τους εκείνη τη στιγμή κλπ.

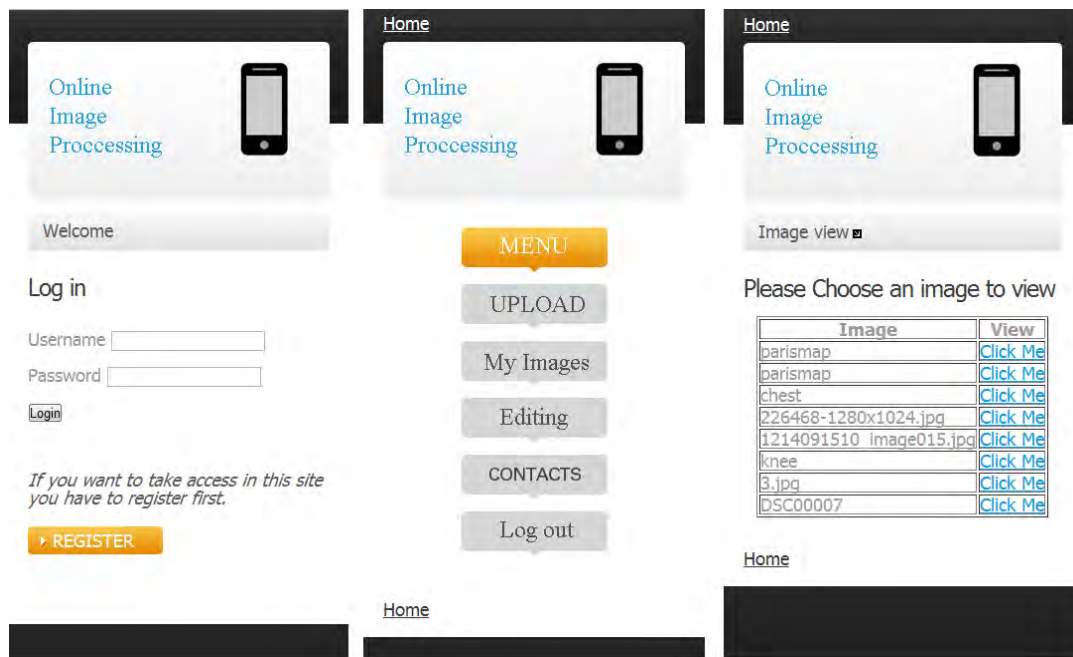
4. Οι χρήστες στο διαδίκτυο, συνήθως, μένουν για πολύ λίγο χρόνο σε μια σελίδα, και μεταπηδούν από το ένα παράθυρο του φυλλομετρητή στο άλλο. Άρα πρέπει να τους δίνουμε αυτό που θέλουν άμεσα, εύκολα και κατανοητά.
5. Η διεύθυνση της σελίδας (url) πρέπει να είναι ευκολομνημόνευτη.
6. Κάθε υποσελίδα πρέπει να έχει το δικό της τίτλο, οποίος να προδίδει και το περιεχόμενος της.
7. Πρέπει να διατηρούνται τα προκαθορισμένα χρώματα του φυλλομετρητή, όπου υπάρχουν, και να μη γίνεται χρήση των χρωμάτων αυτών με τρόπο που να παραπλανεί τους χρήστες (π.χ. ένα μπλε κείμενο θα προιδεάσει τους επισκέπτες της σελίδας ότι πρόκειται για σύνδεσμο).

### 4.3 Σχεδιασμός σελίδας

Για τη σχεδίαση της ιστοσελίδας του συστήματος χρησιμοποιήθηκαν δύο έτοιμα πρότυπα τα οποία αποτέλεσαν τη βάση της σχεδίασης. Αυτά τα πρότυπα τροποποιήθηκαν και προσαρμόστηκαν στην συγκεκριμένη εφαρμογή, όσον αφορά το περιεχόμενο και τα ιδιαίτερα χαρακτηριστικά της εφαρμογής. Η επιλογή αυτών των προτύπων έγινε γιατί η ανάπτυξη ενός προτύπου από την αρχή θα απαιτούσε ειδικό λογισμικό που συνεπάγεται πρόσθετο κόστος, επιπλέον τα πρότυπα που χρησιμοποιήθηκαν έχουν σχεδιαστεί ειδικά για φορητές συσκευές οπότε είναι προσαρμοσμένα στα ειδικά χαρακτηριστικά αυτών και επιλέχθηκαν δύο πρότυπα, προκειμένου ο συνδυασμός αυτών να καλύψει πλήρως τις ανάγκες του συστήματος για τη διεπαφή χρήστη.

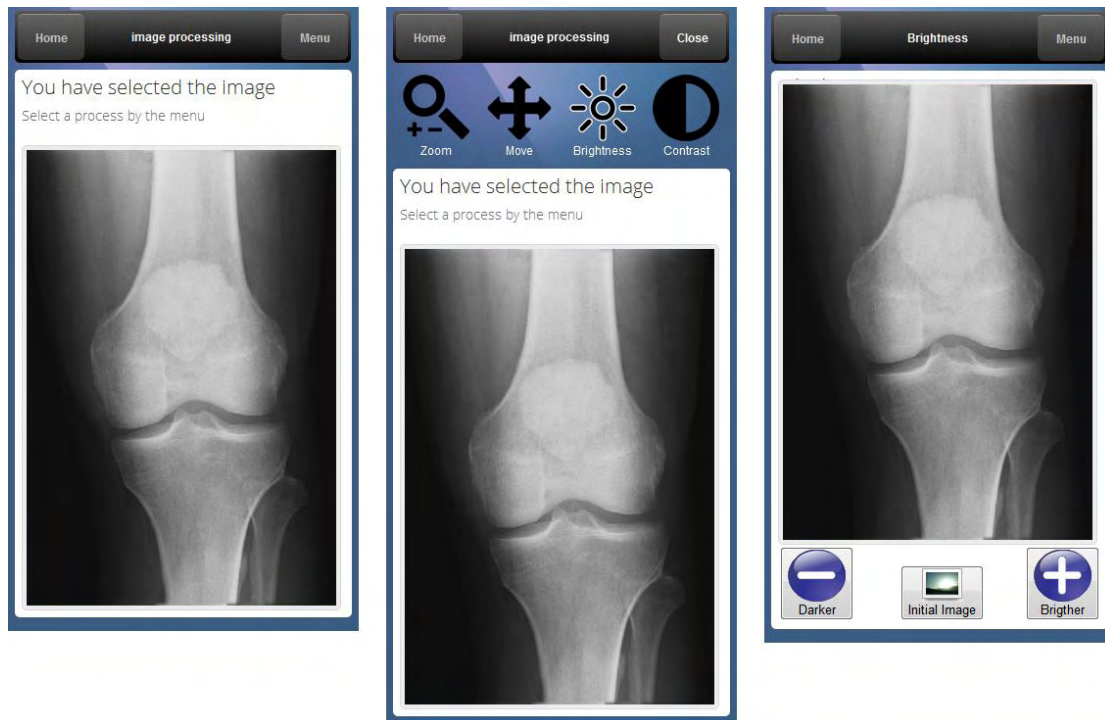
Το πρώτο πρότυπο [9] είναι δωρεάν και χρησιμοποιήθηκε κατά κόρων στη δημιουργία της ιστοσελίδας. Αυτό το πρότυπο διαθέτει απλό και εύχρηστο σχεδιασμό ιδανικό για φορητές συσκευές και χρησιμοποιήθηκε σε όλες τις σελίδες του συστήματος με εξαίρεση τις σελίδες όπου αφορούν άμεσα την επεξεργασία εικόνων. Όπως φαίνεται στο Σχήμα 8 οι σελίδες που ακολουθούν αυτό το πρότυπο έχουν χρωματικές αποχρώσεις άσπρου μαύρου και στο πάνω μέρος τον τίτλο της σελίδας, ενώ στο πάνω και κάτω μέρος υπάρχει η επιλογή για την επιστροφή στην αρχική σελίδα, προκειμένου να μπορεί ο χρήστης να επιστρέψει εύκολα στην αρχή.





Σχήμα 8: Σχεδίαση του πρώτου προτύπου ιστοσελίδας

Το δεύτερο πρότυπο [10] που χρησιμοποιήθηκε είχε κάποιο μικρό κόστος της τάξης των δέκα δολαρίων και υιοθετήθηκε από τις σελίδες που αναφέρονται στο κομμάτι της επεξεργασίας των εικόνων. Το πρότυπο αυτό παρέχει αναδιπλούμενα μενού που εξοικονομούν χώρο στην οθόνη για προβολή των εικόνων και έχει λειτουργίες φυσικής ανάδρασης του χρήστη με την οθόνη αφής όπως η αντίδραση του μενού με το σύρσιμο του δακτύλου του χρήστη στην οθόνη. Στο Σχήμα 9 απεικονίζονται μερικές ενδεικτικές σελίδες που σχεδιάστηκαν με βάση αυτό το πρότυπο και όπως φαίνεται χαρακτηριστικό αυτό του προτύπου είναι η μπάρα στο πάνω μέρος της σελίδας με τις δύο βασικές επιλογές για επιστροφή στην αρχική σελίδα και την επιλογή προβολής του μενού που περιέχει τις λειτουργίες επεξεργασίας εικόνας.



**Σχήμα 9: Εικόνα του δεύτερου προτύπου ιστοσελίδας**

Εκτός από το γραφικό σχεδιασμό της σελίδας, η σελίδα περιλαμβάνει κώδικα ο οποίος αναλαμβάνει να εκτελέσει τις διάφορες λειτουργίες όπως η επικοινωνία με τον εξυπηρετητή και διάφορες άλλες διαδικασίες απαραίτητες για τη λειτουργία του συστήματος. Αξίζει να σημειωθεί πως αν και το γραφικό κομμάτι της διεπαφής είχε κάποιο κόστος, η ανάπτυξη του κώδικα που διεκπεραιώνει τις λειτουργίες δεν κόστισε τίποτα, δημιουργήθηκε από την αρχή με τη βοήθεια ενός απλού επεξεργαστή κειμένου.





#### **4.4 Διαγράμματα ροής δεδομένων**

Τα διαγράμματα ροής δεδομένων (ΔΡΔ) είναι εργαλεία της δομημένης ανάλυσης και σχεδίασης του συστήματος και απεικονίζουν γραφικά τη σχέση μεταξύ διεργασιών και δεδομένων. [2], [1].

Τα διαγράμματα ροής δεδομένων έχουν τέσσερα κύρια στοιχεία: τη ροή δεδομένων (data flow), τη διεργασία (process), την εξωτερική οντότητα (interface)

και την αποθήκευση δεδομένων (data store), όπως παριστάνονται στον ακόλουθο Πίνακα 1.

**Πίνακας 1: Τα τέσσερα κύρια στοιχεία των διαγραμμάτων ροής δεδομένων**

α/α	Τύπος	Περιγραφή	Συμβολισμός
1.	<b>Ροή δεδομένων (data flow)</b>	Δίαυλοι κυκλοφορίας πληροφορίας γνωστού περιεχομένου.	
2.	<b>Διεργασία (process)</b>	Εργασίες που γίνονται από ανθρώπους, μηχανές ή Η/Υ σε εισερχόμενες ροές δεδομένων με σκοπό την παραγωγή εξερχόμενων ροών δεδομένων.	
3.	<b>Εξωτερική οντότητα (interface)</b>	Οντότητες που βρίσκονται εκτός του συστήματος αλλά αποτελούν πηγή ή προορισμό δεδομένων του.	
4.	<b>Αποθήκευση δεδομένων (data store)</b>	Αποθήκες δεδομένων. Μπορούν να επικοινωνούν μόνο με διεργασίες.	

Οι κυριότερες δυνατότητες των ΔΡΔ είναι ότι παρέχουν στοιχεία για:

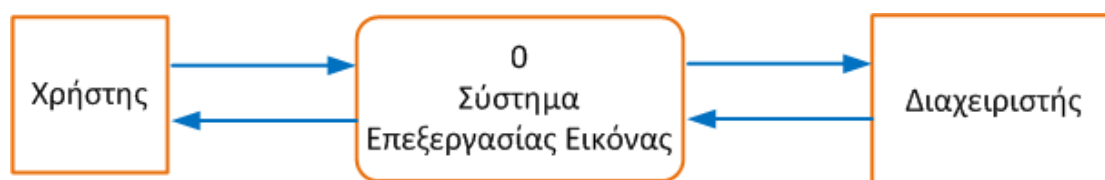
- Τη διάσπαση ενός συστήματος σε υποσυστήματα.
- Τις ροές δεδομένων στο σύστημα.
- Τα εισερχόμενα και εξερχόμενα δεδομένα καθώς και τις αποθηκεύσεις τους.
- Τις πηγές και τους προορισμούς του συστήματος.

Η κατασκευή ενός ΔΡΔ είναι μια διαδικασία από πάνω προς τα κάτω (top-down) για να υπάρχει πληρέστερη και λεπτομερέστερη αναπαράσταση του συστήματος. Έτσι, προκύπτει η ακόλουθη ιεραρχία:

- Γενικό διάγραμμα ή Διάγραμμα επιπέδου μηδέν
- Διάγραμμα πρώτου επιπέδου
- Διάγραμμα κατωτέρων επιπέδων

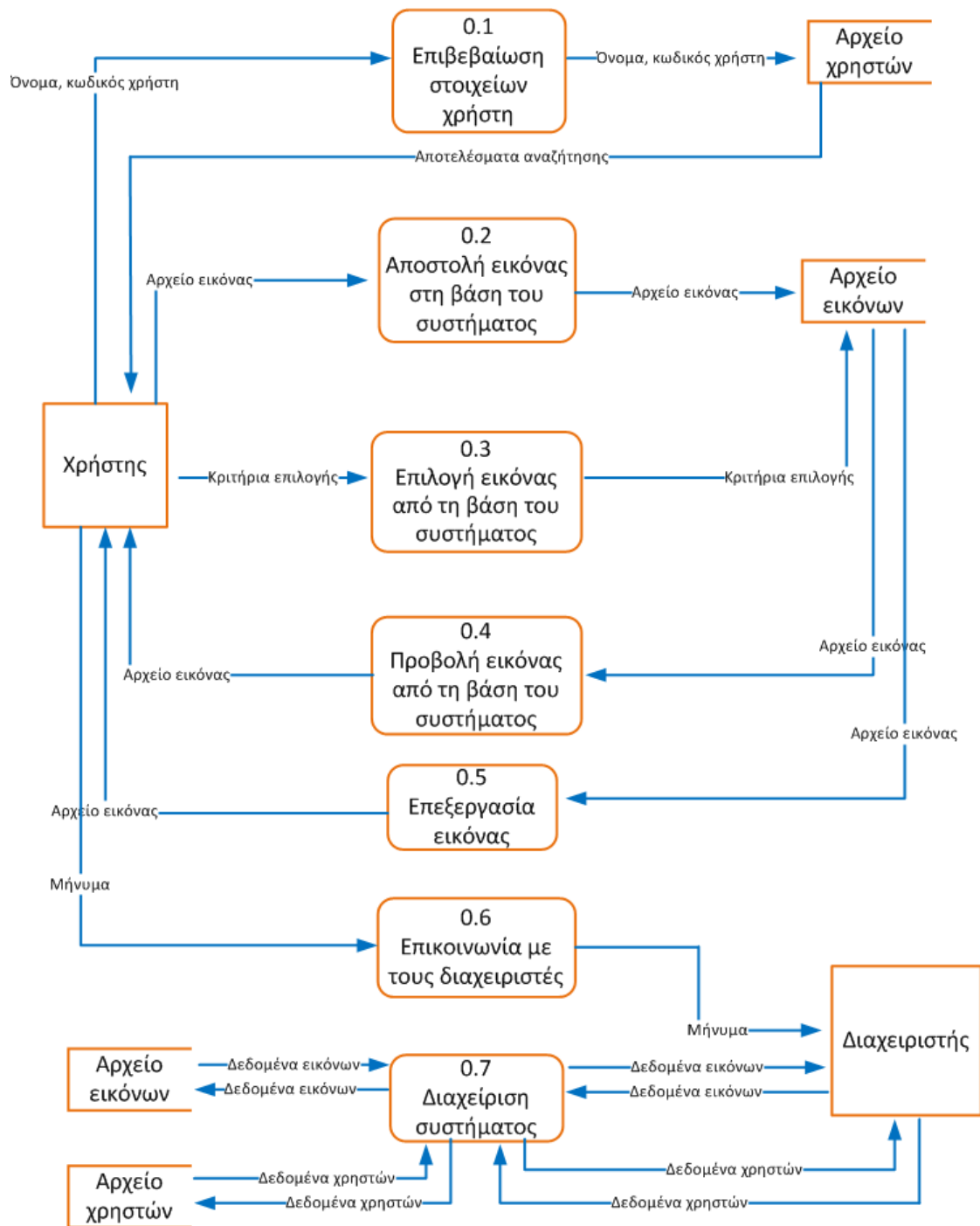
## Εφαρμογή στο σύστημα

Στο παρακάτω διάγραμμα επιπέδου μηδέν (Σχήμα 10) βλέπουμε ότι με το σύστημα αλληλεπιδρούν οι χρήστες και οι διαχειριστές του συστήματος, στέλλοντας και λαμβάνοντας δεδομένα. Ουσιαστικά σε αυτό το διάγραμμα διαφαίνονται τα όρια του συστήματος.



Σχήμα 10: Διάγραμμα ροής δεδομένων Επιπέδου 0

Στο διάγραμμα πρώτου επιπέδου που ακολουθεί (Σχήμα 11) περιγράφονται οι βασικές διεργασίες, ροές και αρχεία δεδομένων. Αρχικά θα πρέπει οι χρήστες του συστήματος να εισάγουν το όνομα και το κωδικό χρήστη προκειμένου να εισέλθουν στο σύστημα. Αφού τα στοιχεία που εισήγαγαν αντιστοιχούν σε κάποιο εγγεγραμμένο χρήστη, μπορούν εισέλθουν στο σύστημα και να επιλέξουν μια από τις λειτουργίες που αφορούν την αποστολή εικόνας στη βάση του συστήματος, προβολή εικόνας που έχει ήδη αποσταλεί στη βάση, επεξεργασία εικόνας και επικοινωνία με τους διαχειριστές του συστήματος. Επίσης από αυτό το διάγραμμα φαίνεται ότι οι διαχειριστές του συστήματος εκτελούν λειτουργίες διαχείρισης του συστήματος και έχουν πρόσβαση στα αρχεία που περιέχουν τα στοιχεία των χρηστών και των απεσταλμένων εικόνων.

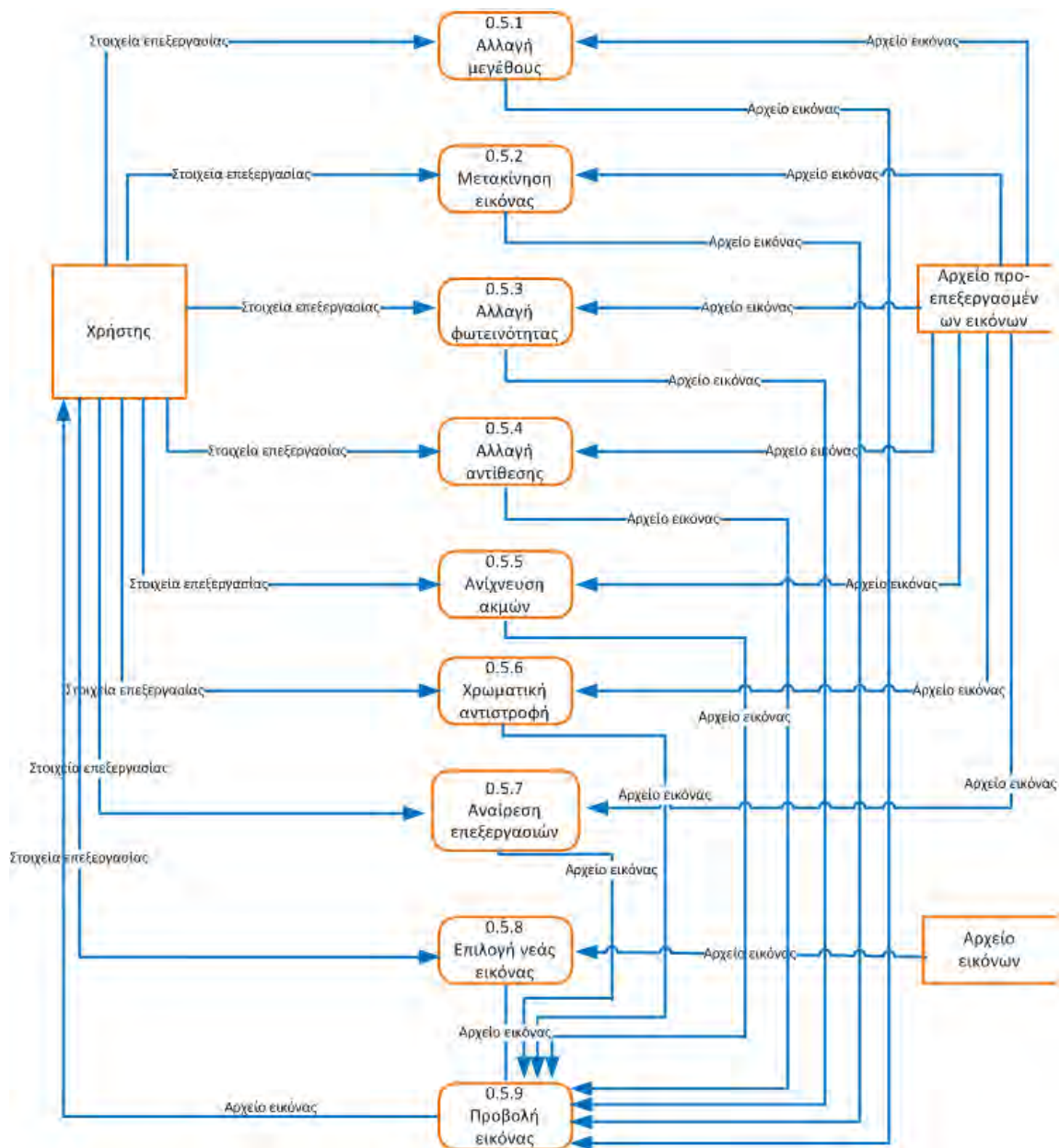


Σχήμα 11: Διάγραμμα ροής δεδομένων Επιπέδου 1

Λόγω της ανάγκης για βαθύτερη κατανόηση, μια διεργασία από το πρώτο επίπεδο και οι σχετικές ροές δεδομένων αναπαριστώνται πληρέστερα στα διαγράμματα κατώτερου επιπέδου. Οπότε κρίθηκε καλό να διασπαστεί η διεργασία 0.5 και 0.7 του ανώτερου επιπέδου, που αφορούν διεργασίες επεξεργασίας εικόνας και διαχείρισης συστήματος αντίστοιχα και να περιγραφούν αναλυτικότερα.

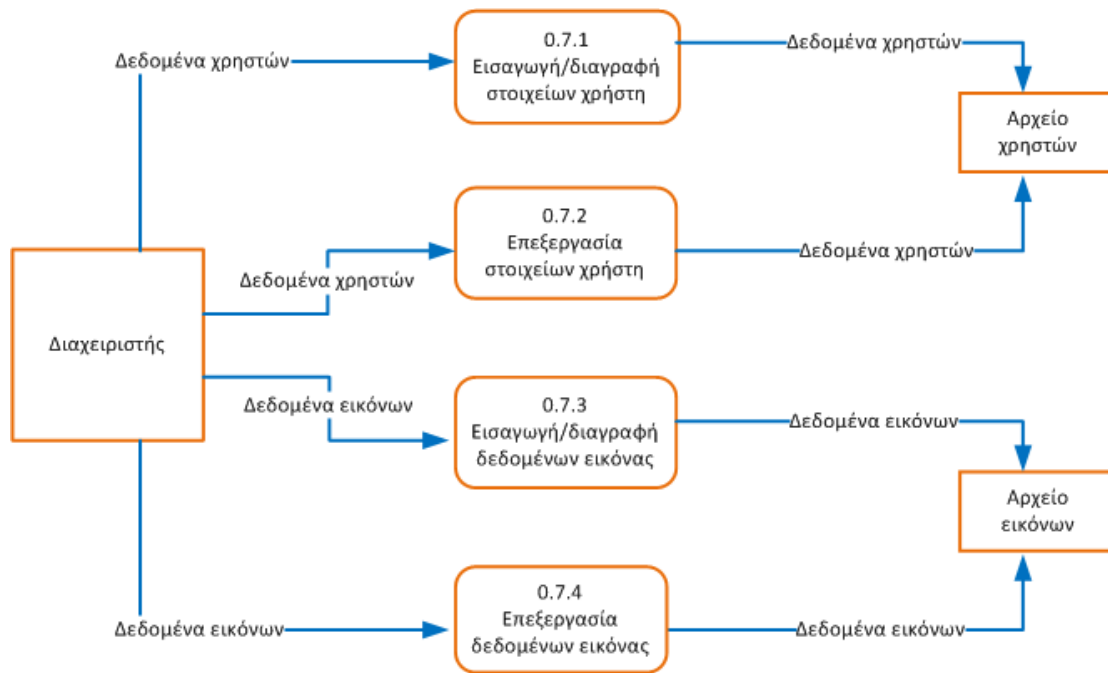
Στη διεργασία επεξεργασία εικόνας (Σχήμα 12) συμμετέχει ο χρήστης και μπορεί να επεξεργαστεί μια εικόνα μεταβάλλοντας το μέγεθός της, μετακινώντας την, αλλάζοντας τη φωτεινότητα της, την αντίθεσή της, να εφαρμόσει μέθοδο ανίχνευσης ακμών ή χρωματική αντιστροφή και επιπλέον έχει τη δυνατότητα να αναιρέσει τις επεξεργασίες που έχουν εφαρμοστεί στην εικόνα επαναφέροντάς την στην αρχική μορφή ή να επιλέξει νέα. Αυτές οι επεξεργασίες εφαρμόζονται στην εικόνα που έχει επιλέξει ο χρήστης από τη βάση και δεν αποθηκεύονται στην εικόνα που υπάρχει στη βάση.

Όπως έχει προαναφερθεί το σύστημα χρησιμοποιεί μεθόδους προ-επεξεργασίας δεδομένων και παράγει τις επεξεργασμένες εικόνες πριν τις ζητήσει ο χρήστης, έτσι στο παρακάτω σχήμα η αποθήκη δεδομένων Αρχείο προ-επεξεργασμένων εικόνων περιέχει αυτές τις εικόνες οι οποίες είναι αποτέλεσμα προ-επεξεργασίας. Όταν ο χρήστης επιλέξει την εικόνα και την επεξεργασία και το βαθμό που θέλει να την εφαρμόσει, ανακτάται η εικόνα από το Αρχείο προ-επεξεργασμένων εικόνων και προβάλλεται στον χρήστη.



Σχήμα 12: Διάγραμμα ροής δεδομένων Επιπέδου 2 για τη διεργασία 0. 5 του Επιπέδου 1

Στο παρακάτω διάγραμμα (Σχήμα 13) αναλύεται η διεργασία της διαχείρισης του συστήματος. Όπως είναι λογικό τη διαχείριση του συστήματος την εκτελούν οι διαχειριστές του συστήματος. Οι λειτουργίες διαχείρισης περιλαμβάνουν την εισαγωγή νέου χρήστη στο σύστημα, την επεξεργασία των στοιχείων του και την διαγραφή των στοιχείων ή και του χρήστη από το σύστημα. Επιπλέον οι διαχειριστές του συστήματος έχουν πρόσβαση στις εικόνες που έχει αποστείλει ο κάθε χρήστης στη βάση του συστήματος και μπορούν να επεξεργαστούν τα στοιχεία των εικόνων που έχουν αποσταλεί, να εισάγουν νέα ή να διαγράψουν κάποια υπάρχοντα.

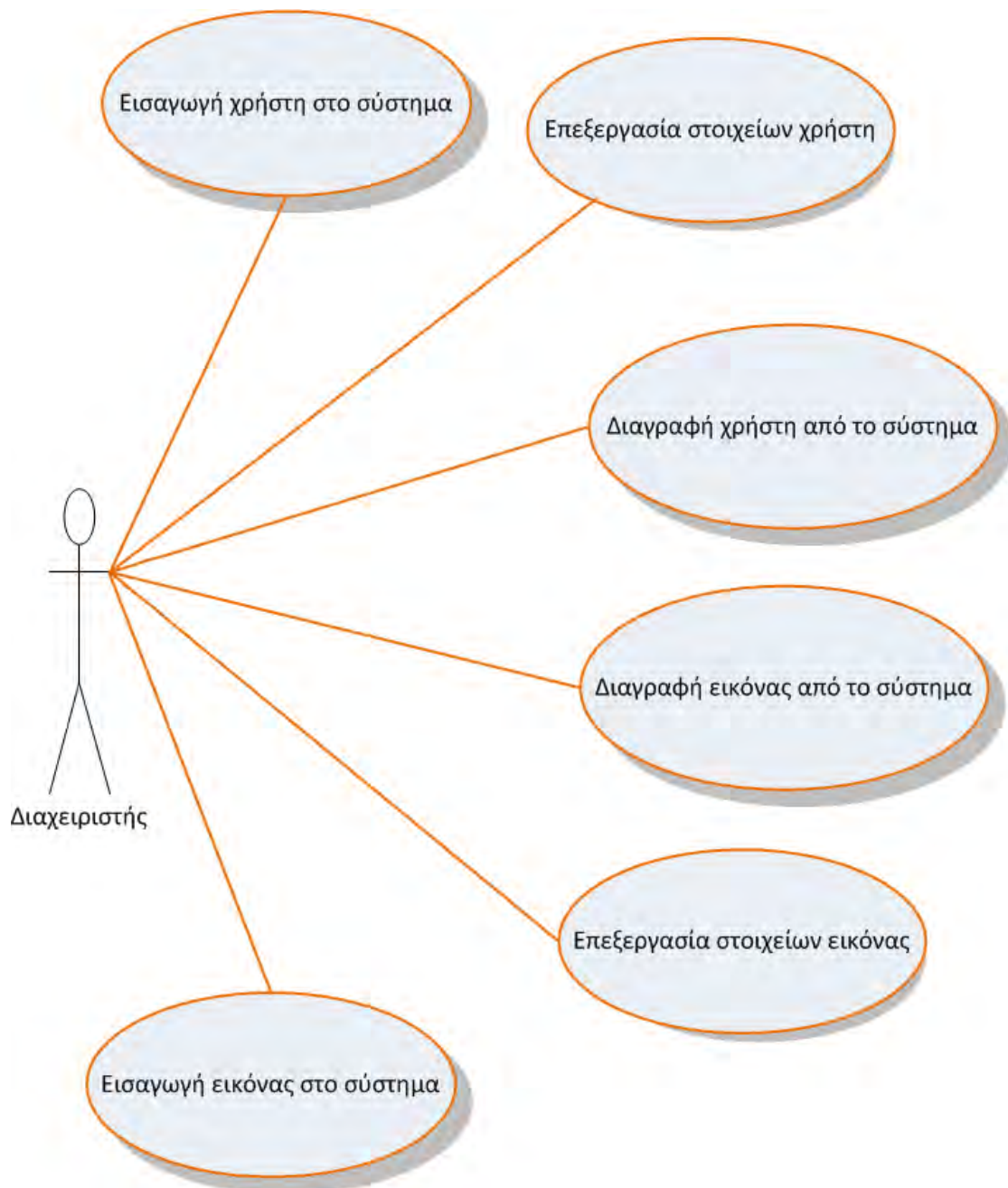


Σχήμα 13: Διάγραμμα ροής δεδομένων Επιπέδου 2 για τη διεργασία 0.7 του Επιπέδου 1

## 4.5 Διάγραμμα χρήσης συστήματος

Ο διαχειριστής του συστήματος όπως φαίνεται και στο παρακάτω Σχήμα 14, μπορεί να εισάγει νέο χρήστη και εικόνες στο σύστημα, να επεξεργαστεί τα στοιχεία των ήδη εγγεγραμμένων χρηστών και των εικόνων που έχουν αποστείλει στη βάση και τέλος να διαγράψει κάποιον χρήστη ή κάποια εικόνα χρήστη.

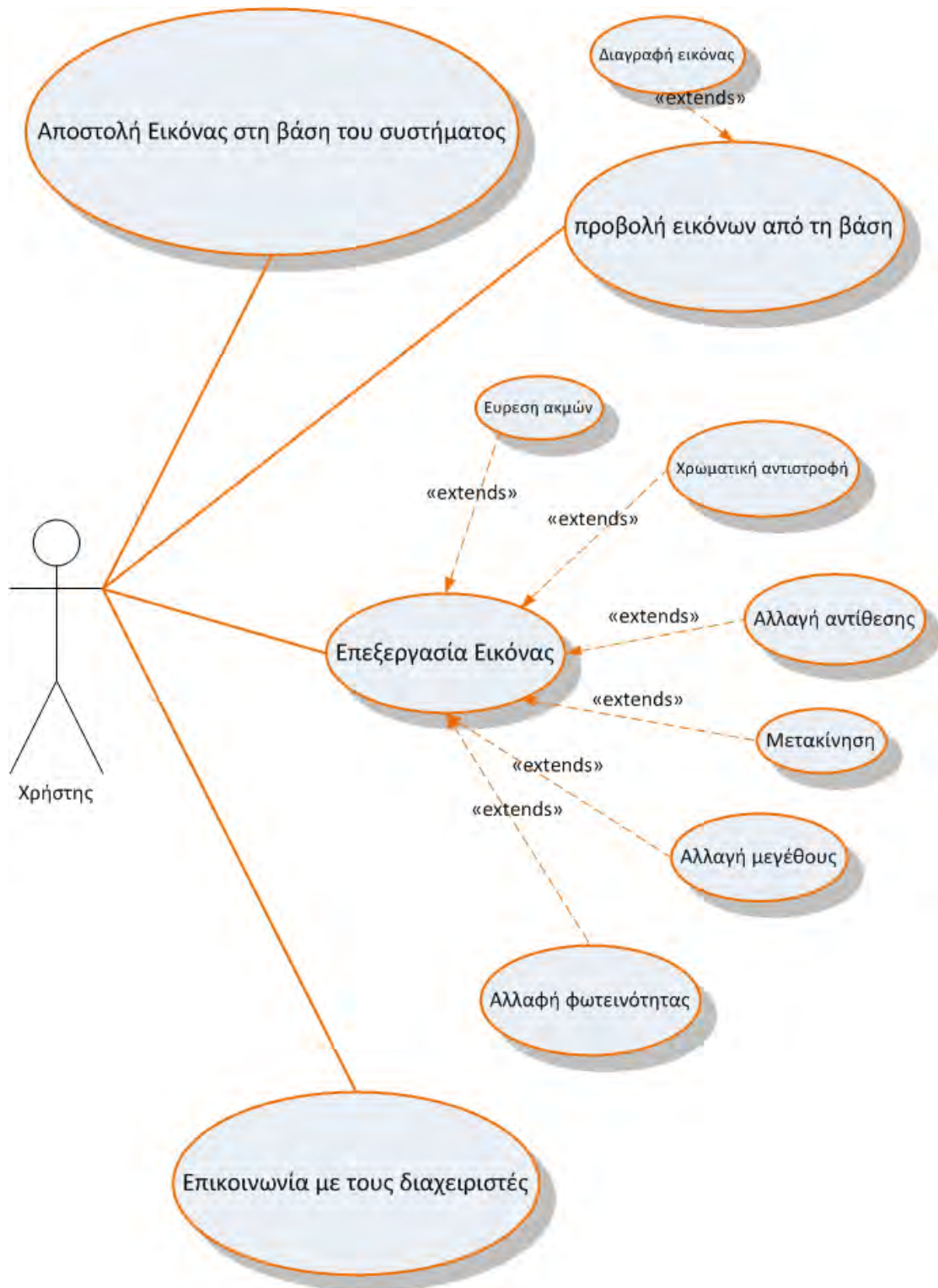




**Σχήμα 14: Διάγραμμα περίπτωσης χρήσης διαχειριστή**

Στο παρακάτω Σχήμα 15 απεικονίζονται οι λειτουργίες που μπορεί να εκτελέσει ο χρήστης στο σύστημα. Ερμηνεύοντας το σχήμα από πάνω προς τα κάτω, διαπιστώνεται ότι οι λειτουργίες περιλαμβάνουν την αποστολή εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος, την προβολή των ήδη απεσταλμένων εικόνων και τη δυνατότητα διαγραφής αυτών από τον χρήστη, την επεξεργασία των εικόνων του χρήστη, την επικοινωνία με τους διαχειριστές του συστήματος και την αποσύνδεση από το σύστημα. Η επεξεργασία των εικόνων περιλαμβάνει τη

μεγέθυνση, μετακίνηση, αλλαγή φωτεινότητας, αλλαγή αντίθεσης, εύρεση ακμών και χρωματική αντιστροφή.



Σχήμα 15: Διάγραμμα περίπτωσης χρήσης χρήστη

## 4.6 Διαμοίραση λειτουργιών πελάτη-εξυπηρετητή

Το σύστημα που αναπτύχθηκε όντας διαδικτυακή εφαρμογή, περιέχει λειτουργίες οι οποίες εκτελούνται από το φυλλομετρητή στη συσκευή του χρήστη και λειτουργίες οι οποίες εκτελούνται στον εξυπηρετητή. Όπως έχει προαναφερθεί στη συσκευή του χρήστη εκτελούνται απλές διεργασίες προβολής εικόνων, αποστολής και λήψης δεδομένων μέσω της ιστοσελίδας. Για τη κατασκευή αυτής της ιστοσελίδας προκειμένου να δημιουργεί το γραφικό περιβάλλον χρήστη, να επικοινωνεί με τη βάση δεδομένων του συστήματος για ανάκτηση και εισαγωγή δεδομένων και να αναθέτει αιτήματα επεξεργασίας στον εξυπηρετητή έχει χρησιμοποιηθεί ένα σύνολο γλωσσών ανάπτυξης διαδικτυακών εφαρμογών. Κάθε μια από αυτές τις γλώσσες, εξυπηρετεί ένα συγκεκριμένο σκοπό και χρησιμοποιείται καταλλήλως. Η καρδιά του συστήματος, εκεί όπου εκτελούνται οι λειτουργίες επεξεργασίας εικόνας, είναι ο εξυπηρετητής. Ο εξυπηρετητής αναλαμβάνει το κύριο όγκο επεξεργασίας και δημιουργίας εικόνων, είναι εφοδιασμένος με κάρτα γραφικών γενικού σκοπού όπου εκτελούνται παράλληλα αλγόριθμοι επεξεργασίας εικόνας.

### 4.6.1 Λειτουργίες που θα εκτελεί η συσκευή του χρήστη

Η συσκευή του χρήστη μέσω του φυλλομετρητή που διαθέτει εκτελεί τον κώδικα που βρίσκεται στη σελίδα του συστήματος. Οι γλώσσες που χρησιμοποιήθηκαν για τη δημιουργία της ιστοσελίδας του συστήματος είναι η html, javascript, php, jquery ενώ οι ερωτήσεις στη βάση γίνονται με SQL. Η html και javascript δημιουργούν το γραφικό περιβάλλον ενώ οι υπόλοιπες χρησιμοποιούνται για την επικοινωνία του χρήστη με τον εξυπηρετητή και τη βάση του συστήματος.

Όταν ο χρήστης επισκεφτεί την ιστοσελίδα του συστήματος, ο φυλλομετρητής της συσκευής του αναλαμβάνει να δημιουργήσει το γραφικό περιβάλλον που προκύπτει από τον κώδικα της σελίδας. Το γραφικό περιβάλλον έχει αναπτυχθεί ώστε να εμφανίζει με το βέλτιστο τρόπο το περιεχόμενο σε οθόνες μικρού μεγέθους, το περιεχόμενο να είναι ευδιάκριτο και εύκολα επιλέξιμο από οθόνες αφής. Επίσης εκμεταλλεύεται τις οθόνες αφής και προσφέρει λειτουργίες όπως το εικονικό σύρσιμο του μενού με το δάκτυλο.

Μόλις ο χρήστης συνδεθεί στο σύστημα αποκτά πρόσβαση στη βάση του συστήματος και δημιουργείται ένα session [21] (τα sessions είναι μια δυνατότητα που προσφέρει η rhp και είναι ένας τρόπος για τη διατήρηση δεδομένων μεταξύ διαδοχικών επισκέψεων σε μια ιστοσελίδα) καθ' όλη την επικοινωνία με το σύστημα. Οι λειτουργίες που εκτελούνται μέσω του φυλλομετρητή της συσκευής είναι η επικοινωνία με τη βάση του συστήματος και η αποστολή και λήψη δεδομένων, η αποστολή αιτήσεων για προ-επεξεργασία εικόνων και προβολή των παραγόμενων νέων εικόνων.

Η σελίδα προκειμένου να αποσταλεί στον εξυπηρετητή επεξεργασίας εικόνων αίτηση για προ-επεξεργασία αντιλαμβάνεται σε κάθε στάδιο που βρίσκεται ο χρήστης και προβλέπει τις αμέσως επόμενες κινήσεις του χρήστη. Τα στοιχεία αυτά αποστέλλονται στον εξυπηρετητή και παράγονται οι επεξεργασμένες εικόνες οι οποίες έχουν κωδικό όνομα. Όταν ο χρήστης επιλέξει να εφαρμόσει μια επεξεργασία σε μια εικόνα, υπολογίζεται από τη σελίδα το κωδικό όνομα της εικόνας με βάση την επιλογή του χρήστη και γίνεται μεταφορά και προβολή της εικόνας στην συσκευή του χρήστη. Όταν ο χρήστης επιλέγει να εφαρμόσει μια επεξεργασία σε μια εικόνα εκτελούνται εργασίες μεταφοράς της εικόνας και όχι υπολογισμού, καθώς η εικόνα έχει προ-υπολογιστεί σε προγενέστερο χρόνο.

Με την αποσύνδεση του χρήστη από το σύστημα, διαγράφονται όλες οι εικόνες που έχουν παραχθεί για τον συγκεκριμένο χρήστη κατά τη προ-επεξεργασία στον εξυπηρετητή, διαγράφεται το session και οι πληροφορίες που είχαν αποθηκευτεί στη συσκευή του χρήστη και εμφανίζεται η αρχική οθόνη του συστήματος. Ο χρήστης σε περίπτωση που δεν κάνει αποσύνδεση από το σύστημα και απλώς φύγει από την ιστοσελίδα, τα δεδομένα παραμένουν στον εξυπηρετητή και το session παραμένει ανοιχτό. Αν δεν κάνει αποσύνδεση μπορεί να επισκεφτεί την ιστοσελίδα του συστήματος μετά από κάποιο χρόνο με την ίδια συσκευή που χρησιμοποίησε και να συνεχίσει από το σημείο που είχε μείνει.

#### **4.6.2 Λειτουργίες που θα εκτελεί ο εξυπηρετητής**

Ο εξυπηρετητής επεξεργασίας αποτελεί τον πυρήνα του συστήματος καθώς εκεί εκτελείται η επεξεργασία των εικόνων. Τα προγράμματα τα οποία εκτελούνται

δημιουργήθηκαν με τις γλώσσες προγραμματισμού c/cuda. Τα προγράμματα αυτά καλούνται από τη γραμμή εντολών με τα κατάλληλα ορίσματα και δεν έχουν γραφικό περιβάλλον. Η χρήση cuda επιτρέπει την αξιοποίηση της κάρτα γραφικών γενικού σκοπού που διαθέτει ο εξυπηρετητής, καθώς και οποιασδήποτε άλλης κάρτας γραφικών Nvidia που υποστηρίζει cuda.

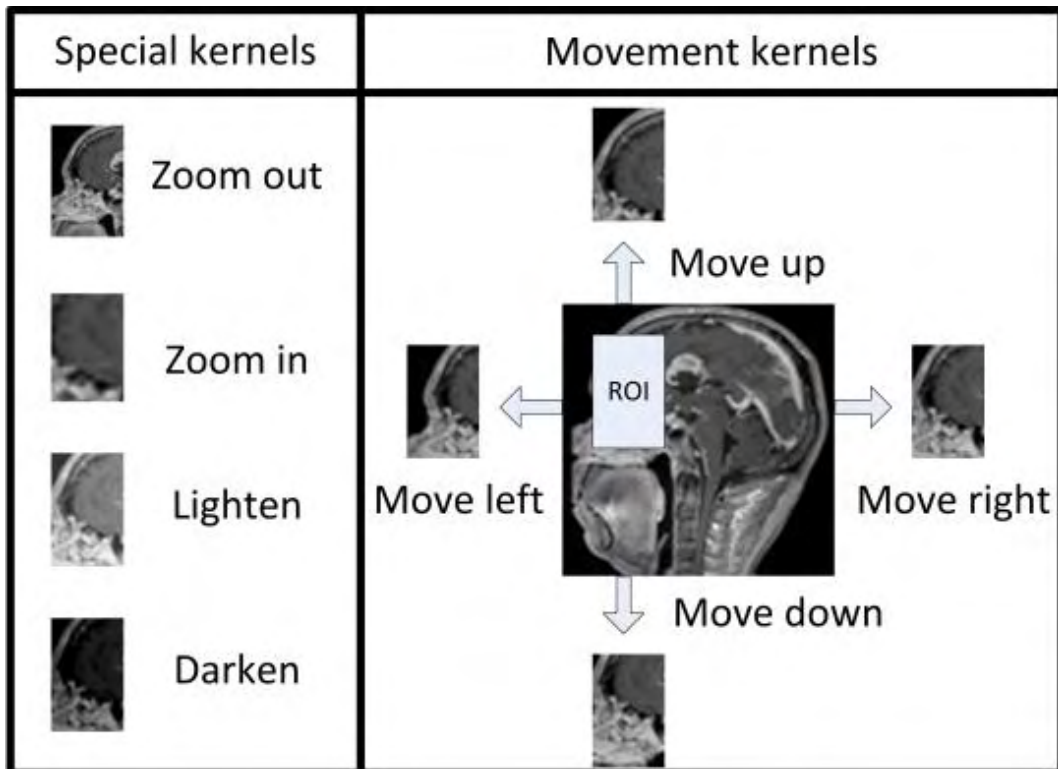
Ο εξυπηρετητής λαμβάνει αιτήματα από τον φυλλομετρητή του χρήστη για προ-επεξεργασία και προχωρά στην κατάλληλη επεξεργασία. Τα αιτήματα αυτά αναφέρουν ποιο πρόγραμμα του εξυπηρετητή θα χρησιμοποιηθεί, ποια εικόνα θα επεξεργαστεί και με ποιόν τρόπο. Οι λειτουργίες που εκτελεί είναι τυπικές επεξεργασίες εικόνων που ωστόσο εκτελούνται με παράλληλο τρόπο. Οι επεξεργασίες που μπορεί να εκτελέσει είναι η μετακίνηση (δεξιά, αριστερά, πάνω και κάτω), μεγέθυνση και σμίκρυνση προβολής, προσαρμογή σε πλήρη οθόνη, αύξηση/μείωση φωτεινότητας, αύξηση/μείωση αντίθεσης, εύρεση ακμών, χρωματική αντιστροφή.

Όταν ο χρήστης επιλέξει μια εικόνα προς επεξεργασία η εικόνα αυτή αποστέλλεται από τη βάση του συστήματος στον εξυπηρετητή εξεργασίας με κωδικό όνομα που προσδιορίζει τον χρήστη που ανήκει και δείκτες που φανερώνουν ότι αυτή η εικόνα είναι η αρχική, δηλαδή είναι ανεπεξεργαστη εικόνα. Έπειτα αφού η εικόνα έχει μεταφερθεί στον εξυπηρετητή η ιστοσελίδα διαβάζει και αποστέλλει τα χαρακτηριστικά της οθόνης της συσκευής του χρήστη στον εξυπηρετητή ο οποίος επεξεργάζεται και αλλάζει το μέγεθος της αρχικής εικόνας στο μέγεθος της οθόνης της συσκευής. Στη συνέχεια γίνεται μετατροπή της εικόνας από jpg σε pbm προκειμένου να μπορεί να τη διαβάσει το πρόγραμμα επεξεργασίας.

Για κάθε επεξεργασία που εφαρμόζεται σε μια εικόνα παράγεται νέα στην οποία αναγράφεται ο βαθμός που έχει εφαρμοστεί η κάθε επεξεργασία στο κατάλληλο πεδίο του κωδικού ονόματος της εικόνας. Αφού παραχθεί η νέα εικόνα αποτέλεσμα της επεξεργασίας, γίνεται η αντίστροφη μετατροπή του τύπου της από pbm σε jpg, προκειμένου να είναι προσπελάσιμη από το φυλλομετρητή της συσκευής του χρήστη.

Μεταξύ του χρήστη και του συστήματος, υπάρχει μια συνεχή ροή εικόνων, ο χρήστης λαμβάνει εικόνες οι οποίες έχουν προκύψει από την αρχική εικόνα που είναι αποθηκευμένη στη βάση και έχουν προσαρμοστεί για τη προβολή στη συσκευή του. Με αυτόν τον τρόπο ο χρήστης δεν λαμβάνει ποτέ την αρχική εικόνα, αλλά ένα στιγμιότυπο αυτής, που προκύπτει από μια περιοχή ενδιαφέροντος (ROI- Region Of

Interest) αυτής της εικόνας, έχει διαστάσεις σύμφωνες με τη συσκευή του και έχει υποστεί τις επεξεργασίες που έχει επιλέξει ο χρήστης (Σχήμα 16). Με αυτό το χαρακτηριστικό ενισχύεται η ασφάλεια του συστήματος καθώς οι εικόνες που είναι αποθηκευμένες στη βάση του συστήματος δεν εξέρχονται από αυτή ποτέ, πράγμα ιδιαίτερα σημαντικό για εφαρμογές που περιέχουν ευαίσθητα δεδομένα, όπως ιατρικές απεικονιστικές εξετάσεις.



Σχήμα 16: Επιλογή περιοχής ενδιαφέροντος από εικόνα προς επεξεργασία [12]

# Κεφάλαιο 5

## Υλοποίηση

---

5.1 Εισαγωγή

5.2 Αλγόριθμοι επεξεργασίας εικόνας

5.3 Φόρμες Τελικού Συστήματος

5.4 Βάση Δεδομένων

5.5 Το υπολογιστικό σύστημα που χρησιμοποιήθηκε

---

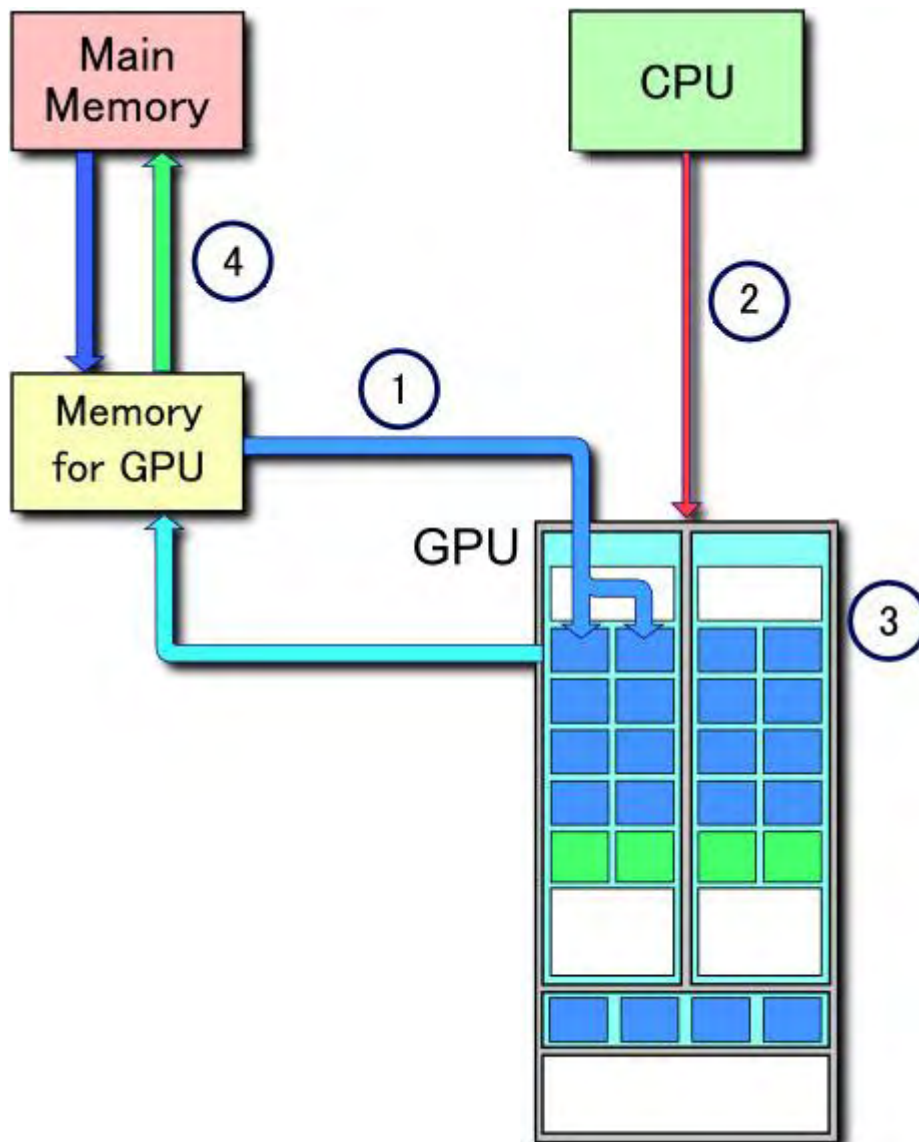
### 5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα περιγραφούν οι κύριες λειτουργίες του συστήματος και θα παρουσιαστεί μέρος του κώδικα των προγραμμάτων επεξεργασίας καθώς και οι αλγόριθμοι επεξεργασίας. Επίσης θα παρουσιαστεί το γραφικό περιβάλλον που αποτελεί τη διεπαφή του χρήστη και του χειριστή, η βάση του συστήματος και το υπολογιστικό σύστημα που χρησιμοποιήθηκε.

### 5.2 Υλοποίηση αλγορίθμων επεξεργασίας εικόνας

Τα διάφορα προγράμματα επεξεργασίας εικόνας που εκτελούνται στον εξυπηρετητή επεξεργασίας, ακολουθούν μια κοινή διαδικασία, στο Σχήμα 11 που ακολουθεί αναπαριστάται περιληπτικά αυτή η διαδικασία. Αρχικά ① γίνεται αντιγραφή των δεδομένων από τη μνήμη του συστήματος στη μνήμη της κάρτας γραφικών, έπειτα ② δίνεται η εντολή από τη κεντρική μονάδα επεξεργασίας (CPU) για εκτέλεση στην μονάδα επεξεργασίας γραφικών (GPU), στη συνέχεια ③ γίνεται

παράλληλη εκτέλεση σε κάθε πυρήνα της κάρτας και τέλος ④ αντιγραφή των αποτελεσμάτων από τους πυρήνες της κάρτας στη μνήμη της κάρτας και από εκεί στη μνήμη του συστήματος.



Σχήμα 17: Διαδικασία εκτέλεσης προγραμμάτων [27]

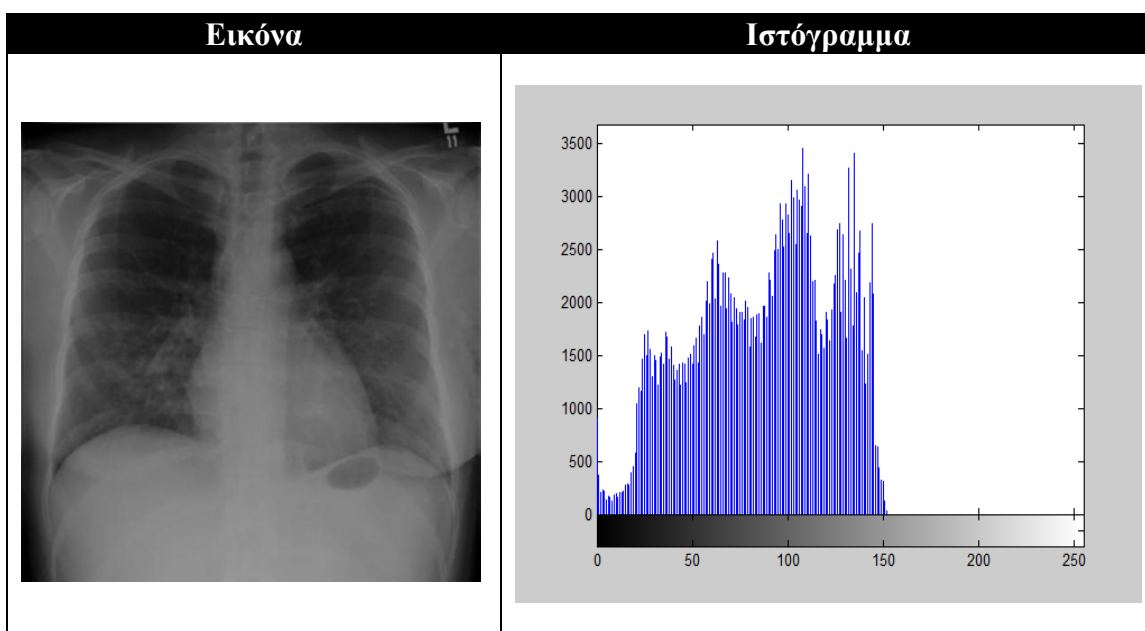
Οι λειτουργίες που περιέχει το σύστημα όσον αφορά την επεξεργασία εικόνων περιλαμβάνουν την αλλαγή φωτεινότητας, την αλλαγή της αντίθεσης, την αλλαγή μεγέθους, τη μετακίνηση, την εύρεση ακμών και τη χρωματική αντιστροφή της εικόνας. Παρακάτω θα παρουσιαστεί μέρος του κώδικα των προγραμμάτων που υλοποιούν αυτές τις λειτουργίες και θα περιγραφεί η λογική στην οποία βασίστηκαν. Η υλοποίηση αυτών των αλγορίθμων έγινε ώστε να αξιοποιούν τις δυνατότητες της κάρτας γραφικών γενικού σκοπού και να εκτελούνται παράλληλα.

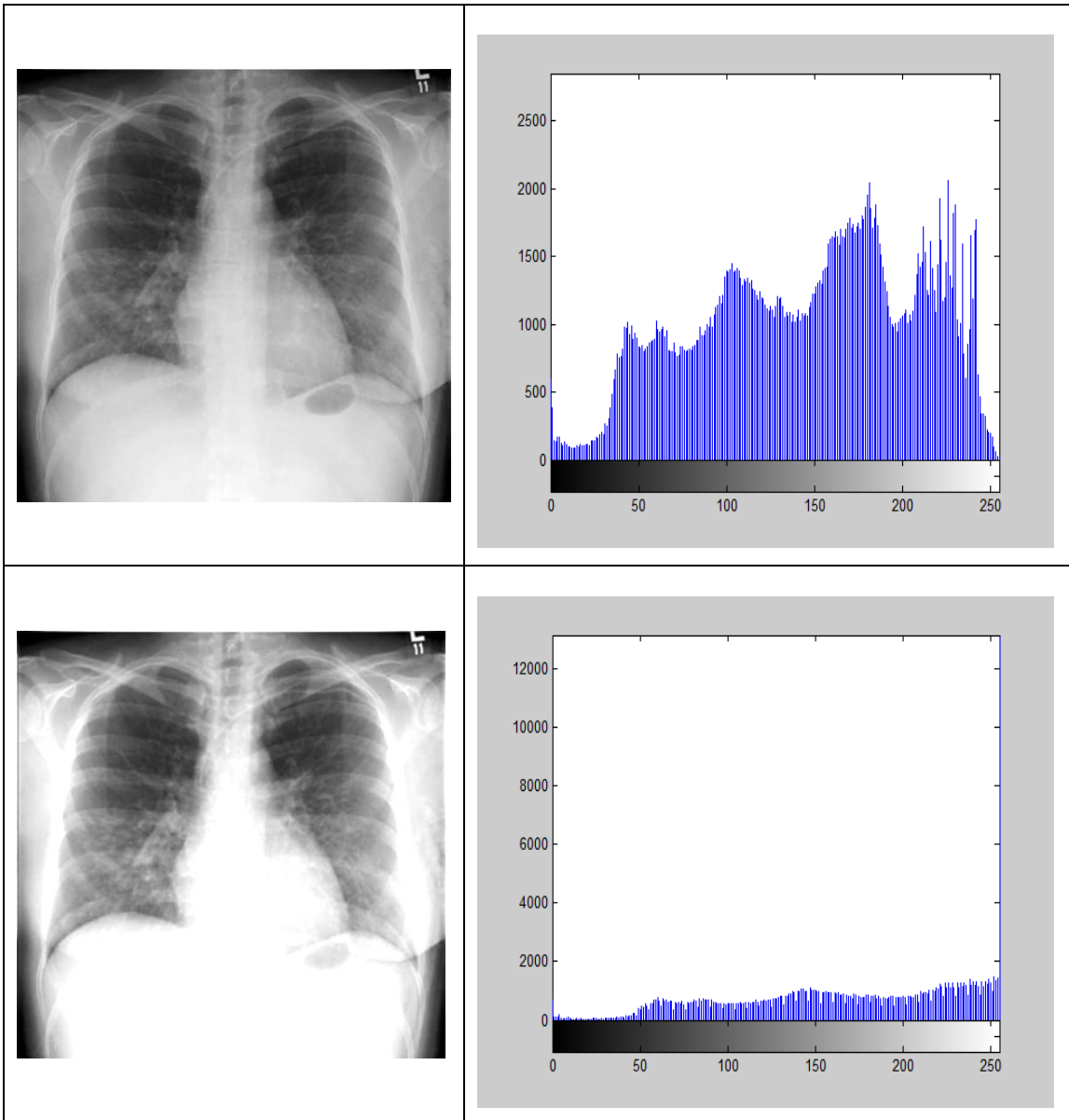


## 5.2.1 Αλγόριθμος αλλαγής φωτεινότητας

Η κάθε εικόνα αποτελείται από ένα σύνολο εικονοστοιχείων τα οποία έχουν κάποια τιμή, η οποία αντιπροσωπεύει τη φωτεινότητα του συγκεκριμένου εικονοστοιχείου. Για τις εικόνες που χρησιμοποιούν 8 bit για να αναπαραστήσουν τη τιμή του κάθε εικονοστοιχείου, οι δυνατές τιμές που μπορεί να πάρει το κάθε εικονοστοιχείο είναι 256, από το 0 έως το 255. Αυτές οι τιμές αναπαριστούν αποχρώσεις του γκρι ξεκινώντας από το απόλυτο μαύρο (0) και καταλήγοντας στο απόλυτο λευκό (255). Για την αλλαγή της φωτεινότητας της εικόνας αυτές οι τιμές των εικονοστοιχείων μεταβάλλονται, όπως φαίνεται στον Πίνακα 2 από το ιστόγραμμα που αντιστοιχεί σε κάθε εικόνα, αυξάνονται όταν η εικόνα είναι πιο φωτεινή και μειώνονται όταν η εικόνα είναι πιο σκούρα.

Πίνακας 2: Αντιστοιχία ιστογραμμάτων με εικόνες διαφορετικής φωτεινότητας





Η υλοποίηση της επεξεργασίας αλλαγής φωτεινότητας είναι απλή και περιλαμβάνει τη πρόσθεση ή αφαίρεση ενός αριθμού σε κάθε εικονοστοιχείο της εικόνας. Επίσης πρέπει να ελεγχθεί εάν η νέα τιμή βρίσκεται στο πεδίο των επιτρεπόμενων τιμών  $[0-255]$ . Παρακάτω στον Πίνακα 3 αναγράφεται ο ψευδοκώδικας για τη μεταβολή της φωτεινότητας της εικόνας και η αντίστοιχη υλοποίηση της συνάρτησης σε Cuda. Η συνάρτηση αυτή παίρνει ως εισόδους ένα δείκτη προς τα δεδομένα της εικόνας, το ύψος, το πλάτος της εικόνας και ένα δείκτη που φανερώνει κατά πόσο και πως θα μεταβληθεί η φωτεινότητα της εικόνας.

Πίνακας 3: Ψευδοκώδικας και η υλοποίηση σε cuda της συνάρτησης αλλαγής φωτεινότητας

Ψευδοκώδικας	Κώδικας Cuda
<pre> newValue= value + brightness If newValue &lt; 0 Then newValue = 0 If newValue &gt; 255 Then newValue = 255                     </pre>	<pre> __global__ void transformKernel( float* g_odata, int width, int height, float br) {      // calculate normalized texture coordinates     unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;     unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;  br=br-1.0;      float u = x / (float) width;     float v = y / (float) height;      // transform coordinates     u -= 0.5f;     v -= 0.5f;     float tu = u + 0.5f;     float tv = v+ 0.5f;  float maxv=0.f;      // read from texture and write to global memory     g_odata[y*width + x] = max(( min(tex2D(tex, tu, tv)+br*tex2D(tex3, maxv, maxv) ,tex2D(tex2, maxv, maxv) ) ),tex2D(tex4, maxv, maxv)) ; ;  }                     </pre>

## 5.2.2 Αλγόριθμος αλλαγής αντίθεσης

Για την αύξηση της αντίθεσης τα φωτεινά σημεία της εικόνας πρέπει να γίνουν περισσότερο φωτεινά και τα σκοτεινά σημεία περισσότερα σκοτεινά. Αυτό φαίνεται και από το παρακάτω Πίνακα 4 όπου με την αύξηση της αντίθεσης της εικόνας μειώθηκε ο αριθμός των εικονοστοιχείων που είχαν τιμή φωτεινότητας κοντά στο 128 και αυξήθηκε ο αριθμός των εικονοστοιχείων που έχουν τιμές κοντά στα

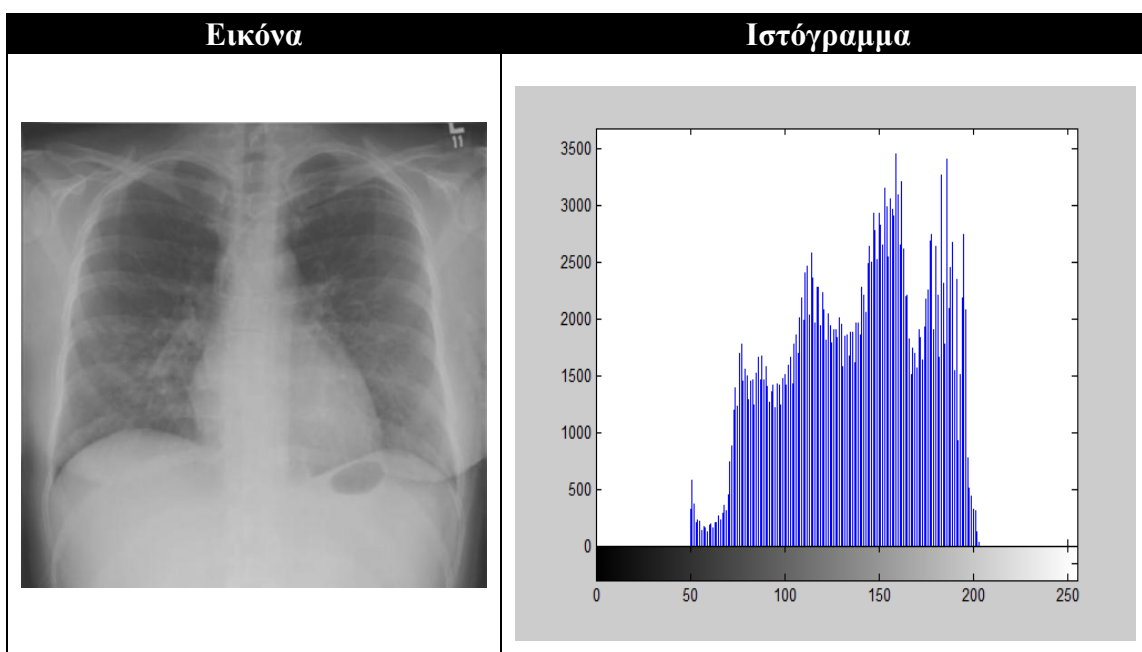
άκρα του πεδίου ορισμού 0 και 255. Ενώ με τη μείωση της αντίθεσης της εικόνας παρατηρείται ότι το εύρος των τιμών φωτεινότητας των εικονοστοιχείων μειώνεται και παρατηρείται συρρίκνωση του ιστογράμματος.

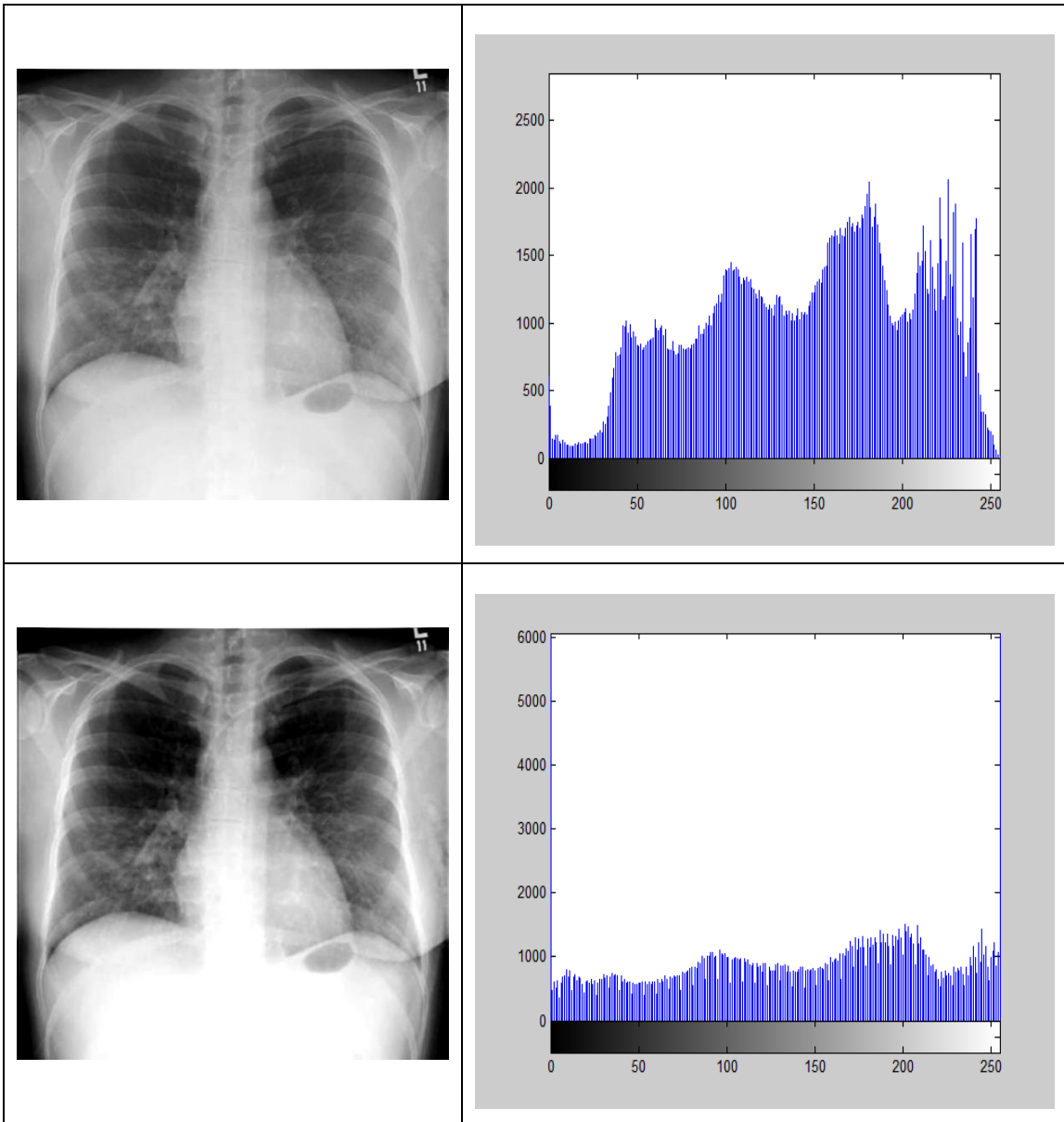
Η συνάρτηση που χρησιμοποιήθηκε για την αλλαγή της αντίθεσης είναι η παρακάτω [23]:

$$\begin{aligned}(\text{output Level} - 127.5) &= \text{contrast} * (\text{input Level} - 127.5) \Rightarrow \\ \text{output Level} &= \text{contrast} * (\text{input Level} - 127.5) + 127.5 \quad (1)\end{aligned}$$

Όπου contrast είναι η μεταβλητή που δείχνει το βαθμό αλλαγής της φωτεινότητας και είναι θετική τιμή. Για  $\text{contrast} < 1$  προκύπτει εικόνα με μικρότερη αντίθεση της αρχικής και για  $\text{contrast} > 1$  προκύπτει εικόνα με μεγαλύτερη αντίθεση από την αρχική εικόνα.

**Πίνακας 4: Αντιστοιχία ιστογραμμάτων με εικόνες διαφορετικής αντίθεσης**





Η υλοποίηση της επεξεργασίας αλλαγής αντίθεσης περιλαμβάνει την εφαρμογή της παραπάνω συνάρτησης (1) σε κάθε εικονοστοιχείο της εικόνας. Επίσης και πάλι πρέπει να ελεγχθεί εάν η νέα τιμή βρίσκεται στο πεδίο των επιτρεπόμενων τιμών [0-255]. Παρακάτω στον Πίνακα 5 αναγράφεται ο ψευδοκώδικας για τη μεταβολή της αντίθεσης της εικόνας και η αντίστοιχη υλοποίηση της συνάρτησης σε Cuda. Η συνάρτηση αυτή παίρνει ως εισόδους ένα δείκτη προς τα δεδομένα της εικόνας, το ύψος, το πλάτος της εικόνας και ένα δείκτη που φανερώνει κατά πόσο και πως θα μεταβληθεί η αντίθεση της εικόνας.

Πίνακας 5: Ψευδοκώδικας και η υλοποίηση σε cuda της συνάρτησης αλλαγής αντίθεσης

Ψευδοκώδικας	Κώδικας Cuda
<pre> newValue= contrastAmount*(value – 127.5) + 127.5 If newValue &lt; 0 Then newValue = 0 If newValue &gt; 255 Then newValue = 255 </pre>	<pre> __global__ void transformKernel( float* g_odata, int width, int height, float clevel) {     float maxv=0.f;      // calculate normalized texture coordinates     unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;     unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;      float u = x / (float) width;     float v = y / (float) height;      // transform coordinates     u -= 0.5f;     v -= 0.5f;     float tu = u + 0.5f;     float tv = v+ 0.5f;      //contrastAmount * (inputLevel - 127.5) + 127.5;     g_odata[y*width + x]= max(min( clevel*(tex2D(tex, tu, tv)-tex2D(tex2, maxv, maxv)/2)+ tex2D(tex2, maxv, maxv)/2, tex2D(tex2, maxv, maxv)), tex2D(tex3, maxv, maxv)) ; } </pre>

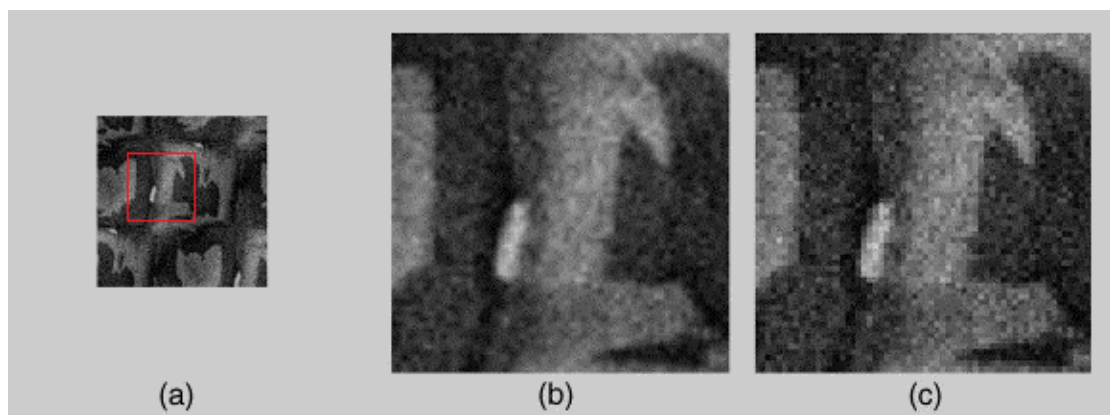
### 5.2.3 Αλγόριθμος αλλαγής μεγέθους

Για την δημιουργία του προγράμματος αλλαγής μεγέθους της εικόνας χρησιμοποιήθηκε η βιβλιοθήκη NVIDIA Performance Primitives [13]. Αυτή η βιβλιοθήκη περιέχει μια συλλογή από συναρτήσεις επεξεργασίας εικόνας, βίντεο και σήματος που σύμφωνα με την Nvidia πετυχαίνουν πέντε με δέκα φορές υψηλότερη ταχύτητα εκτέλεσης σε σχέση με την υλοποίηση των αλγορίθμων σε κεντρική μονάδα επεξεργασίας (cpu).

Από αυτή τη βιβλιοθήκη χρησιμοποιήθηκαν διάφορες συναρτήσεις, η συνάρτηση η οποία πραγματοποιεί την αλλαγή μεγέθους είναι η `ppriResize_8u_C1R`

ενώ οι υπόλοιπες έχουν βοηθητικό ρόλο, όπως η διαχείριση της μνήμης. Η συνάρτηση `ppriResize_du_C1R` δέχεται ως ορίσματα στοιχεία που αφορούν την αρχική εικόνα, την παραγόμενη εικόνα, τους συντελεστές μεγέθυνσης και τον τύπο παρεμβολής που θα εφαρμοστεί.

Κατά τη μεγέθυνση μιας εικόνας δημιουργούνται κάποιες τιμές ώστε να καλύψουν τα εικονοστοιχεία που εμφανίζονται κενά, λόγω της μη επαρκούς ανάλυσης της αρχικής εικόνας. Αυτές οι τιμές δημιουργούνται με ένα αλγόριθμο παρεμβολής (interpolation) (Σχήμα 18). Η μέθοδος παρεμβολής που επιλέχτηκε είναι η γραμμική παρεμβολή λόγω της σχέσης αποτελέσματος, υπολογιστικής πολυπλοκότητας.



**Σχήμα 18:** Μεγέθυνση εικόνας (a). Με γραμμική παρεμβολή (b) και με παρεμβολή γειτονικού εικονοστοιχείου (c) [30]

#### 5.2.4 Αλγόριθμος μετακίνησης εικόνας

Για τη μετακίνηση της εικόνας χρησιμοποιείται η ίδια συνάρτηση που χρησιμοποιήθηκε και στο πρόγραμμα για τη μεγέθυνση της εικόνας. Καλώντας την ίδια συνάρτηση αλλάζοντας τα ορίσματα που αφορούν τη περιοχή ενδιαφέροντος της εικόνας, παράγονται εικόνες οι οποίες αποτελούν ένα μέρος της αρχικής εικόνας. Με αυτό τον τρόπο προκύπτει μια εικόνα που φαίνεται να είναι μετατοπισμένη η αρχική.

### 5.2.5 Αλγόριθμος εύρεσης ακμών εικόνας

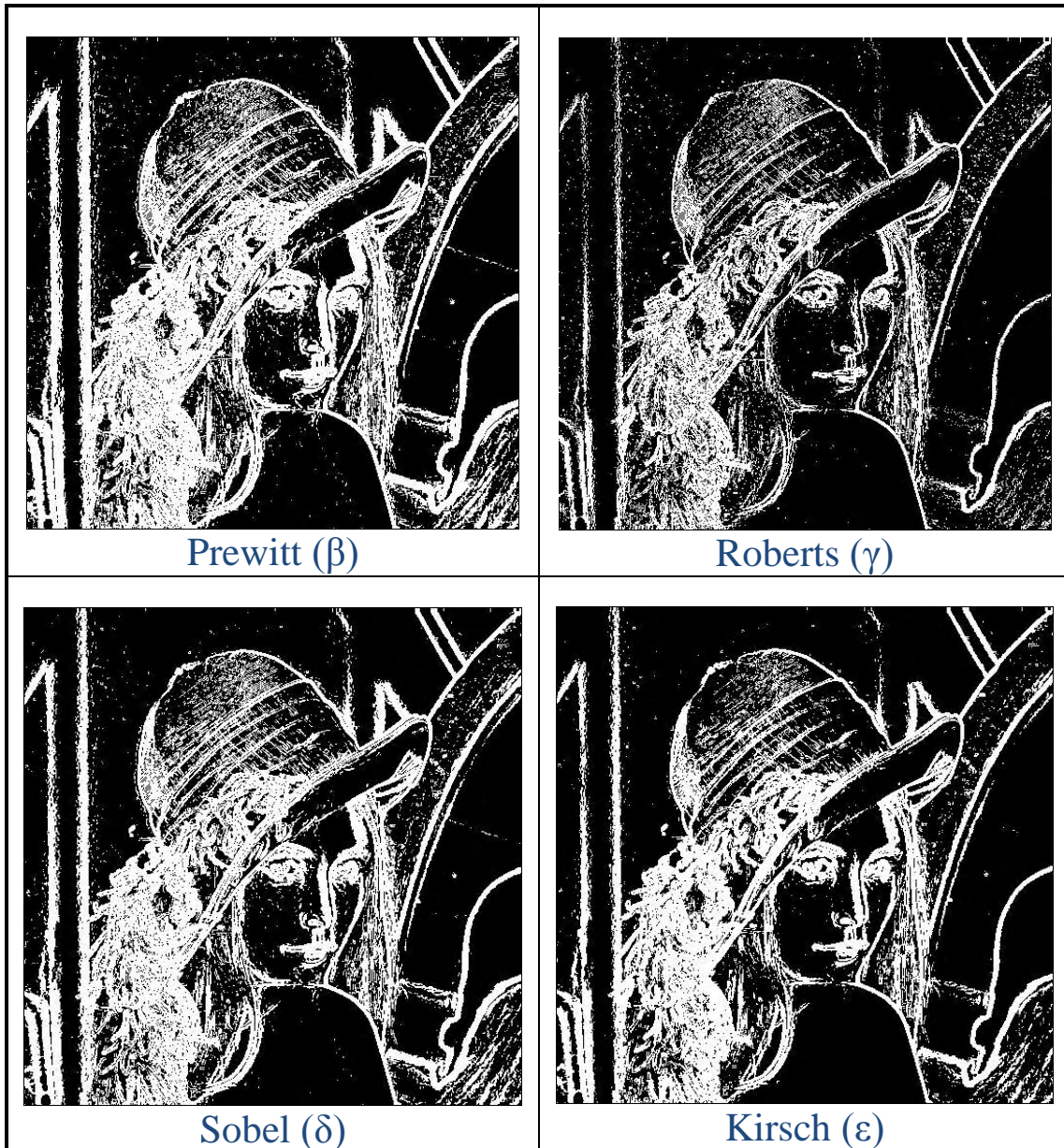
Για την εύρεση ακμών της εικόνας επιλέχθηκε η μέθοδος Sobel [14]. Επιλέχθηκε αυτή η μέθοδος γιατί αν και είναι ίδιας πολυπλοκότητας με τη Prewitt εμφανίζει λιγότερες ακμές, έτσι δεν εμφανίζονται οι μικρότερες ακμές που πιθανών θα προκαλούσαν σύγχυση στο χρήστη. Μελετήθηκε και η μέθοδος του Kirsch, η οποία είναι πιο ανεκτική στο θόρυβο αλλά χρησιμοποιεί οκτώ συνελκτικούς πυρήνες σε αντίθεση με τη μέθοδο Sobel που χρησιμοποιεί δύο στις συνελίξεις που πραγματοποιεί, έτσι αν και τα αποτελέσματα μπορούν να θεωρηθούν καλύτερα, προκαλούν πολλαπλάσιο υπολογιστικό φόρτο. Επίσης, ακόμη μια μέθοδος που μελετήθηκε είναι η Roberts η οποία έχει και αυτή δύο συνελκτικούς πυρήνες, ωστόσο με αυτή τη μέθοδο δεν τονίζονται τόσο οι ακμές όσο με τις μεθόδους Sobel και Prewitt που έχουν και αυτές δύο συνελκτικούς πυρήνες. Οπότε με βάση τις ανάγκες σε υπολογιστικούς πόρους και το αποτέλεσμα της εφαρμογής της κάθε μεθόδου επιλέχθηκε η μέθοδος Sobel.

Πίνακας 6: Εικόνες που προέκυψαν από τις μεθόδους ανίχνευσης ακμών (β)Prewitt, (γ)Roberts, (δ)Sobel, (ε)Kirsch [15]



Αρχική εικόνα (α)





Παρακάτω στον Πίνακα 7 αναγράφεται ο ψευδοκώδικας που υλοποιήθηκε για τη μέθοδο Sobel και η αντίστοιχη η υλοποίηση της συνάρτησης σε Cuda. Η συνάρτηση αυτή παίρνει ως εισόδους ένα δείκτη προς τα δεδομένα της εικόνας, το ύψος, το πλάτος της εικόνας και ένα δείκτη που φανερώνει κατά πόσο και πως θα μεταβληθεί η αντίθεση της εικόνας. Όπου στον ψευδοκώδικα οι πίνακες  $G_x$  και  $G_y$  είναι οι :

$G_x$

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

$G_y$

+1	+2	+1
0	0	0
+1	+2	+1

Πίνακας 7: Ψευδοκώδικας και η υλοποίηση σε cuda της μεθόδου Sobel

Ψευδοκώδικας	Κώδικας Cuda
<pre> for row in 1 to ImageLength loop   for col in 1 to ImageWidth loop     sumx := 0; sumy := 0;     for i in -1 to +1 loop       for j in -1 to +1 loop         sumx := sumx+ o(row+i, col+j) * Gx(i, j);         sumy := sumy+ o(row+i, col+j) * Gy(i, j);       end loop     end loop     D(row, col) := abs(sumx) + abs(sumy)   end loop end loop </pre>	<pre> __global__ void SobelTex( float *pSobelOriginal, unsigned int Pitch,           int w, int h ) {     unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;     unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;      if (x &lt; w &amp;&amp; y &lt; h) {          float pix00 = tex2D( tex, (float) x-1, (float) y-1 );         float pix01 = tex2D( tex, (float) x+0, (float) y-1 );         float pix02 = tex2D( tex, (float) x+1, (float) y-1 );         float pix10 = tex2D( tex, (float) x-1, (float) y+0 );         float pix11 = tex2D( tex, (float) x+0, (float) y+0 );         float pix12 = tex2D( tex, (float) x+1, (float) y+0 );         float pix20 = tex2D( tex, (float) x-1, (float) y+1 );         float pix21 = tex2D( tex, (float) x+0, (float) y+1 );         float pix22 = tex2D( tex, (float) x+1, (float) y+1 );  float maxv=0.f;  pSobelOriginal[y*w+x]=min( abs(pix02+2*pix12+pix22-pix00- 2*pix10- pix20)+abs(pix00+2*pix01+pix02- pix20-2*pix21-pix22), tex2D(tex2, maxv, maxv) );      } } </pre>

## 5.2.6 Αλγόριθμος χρωματικής αντιστροφής εικόνας

Η χρωματική αντιστροφή της εικόνας γίνεται με την αφαίρεση της τιμής κάθε εικονοστοιχείου της εικόνας από τη τιμή 255, η οποία είναι η μέγιστη τιμή που μπορεί να πάρει το εικονοστοιχείο και αντιπροσωπεύει το απόλυτο λευκό [24]. Με αυτό τον τρόπο τα εικονοστοιχεία με τιμή 0 (απόλυτο μαύρο) παίρνουν τη τιμή 255 (απόλυτο λευκό), και τα εικονοστοιχεία με τιμή 255 παίρνουν τη τιμή 0, το ίδιο ισχύει και για όλες τις ενδιάμεσες τιμές.

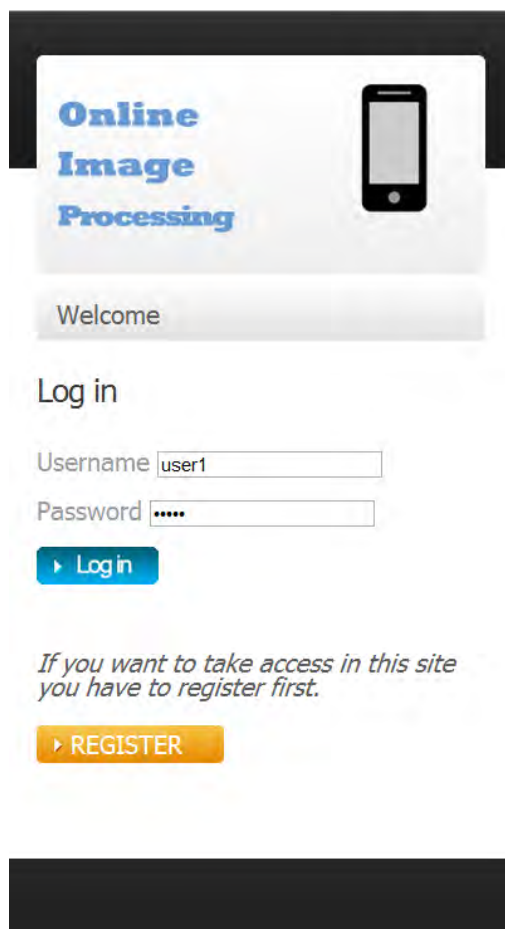
Πίνακας 8: Ψευδοκώδικας και η υλοποίηση σε cuda της συνάρτησης χρωματικής αντιστροφής

Ψευδοκώδικας	Κώδικας Cuda
<pre>newValue = value - 255 If newValue &lt; 0 Then newValue = 0 If newValue &gt; 255 Then newValue = 255</pre>	<pre>__global__ void transformKernel( float* g_odata, int width, int height, float br) {     // calculate normalized texture     coordinates     unsigned int x = blockIdx.x*blockDim.x + threadIdx.x;     unsigned int y = blockIdx.y*blockDim.y + threadIdx.y;      float u = x / (float) width;     float v = y / (float) height;      // transform coordinates     u -= 0.5f;     v -= 0.5f;     float tu = u + 0.5f;     float tv = v + 0.5f;      float maxv=0.f;      // read from texture and write to global     memory     g_odata[y*width + x] = tex2D(tex2, maxv, maxv)-tex2D(tex, tu, tv); }</pre>

## 5.3 Φόρμες Τελικού Συστήματος

### Χρήστης

Η αρχική σελίδα του συστήματος (Σχήμα 19) αποτελείται από βασικά στοιχεία, για να μην προκαλέσει σύγχυση και αποθαρρύνει τον χρήστη να χρησιμοποιήσει την υπηρεσία, αλλά παράλληλα τον πληροφορεί για τον σκοπό του συστήματος. Ο επισκέπτης πληροφορείται από την αρχική σελίδα ότι η ιστοσελίδα αυτή αποτελεί ένα σύστημα επεξεργασίας εικόνας για φορητές συσκευές, ενώ η ύπαρξη φόρμας συμπλήρωσης ονόματος χρήστη, κωδικού, η επιλογή εγγραφής (Register) και η σχετική φράση πληροφορούν τον επισκέπτη ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να εισέλθουν στο σύστημα.



Σχήμα 19: Αρχική σελίδα εισαγωγής στο σύστημα

Εάν ο επισκέπτης της σελίδας επιθυμεί να χρησιμοποιήσει το σύστημα πρέπει να επιλέξει από τη αρχική οθόνη να εγγραφεί στο σύστημα. Στην φόρμα που θα του εμφανιστεί (Σχήμα 20) θα πρέπει να συμπληρώσει κάποια βασικά στοιχεία για αυτόν και τον σκοπό που θα χρησιμοποιήσει το σύστημα. Αφού συμπληρώσει τη φόρμα, θα πρέπει να αποστείλει την αίτησή του στους διαχειριστές του συστήματος, οι οποίοι θα επεξεργαστούν τα στοιχεία και θα εισάγουν το χρήστη στο σύστημα ή θα απορρίψουν την αίτηση.

Back

**Online  
Image  
Processing**

Register

Register information

*Please fill the form  
We will reply to you soon*

Registration form

Name:

Username:

E-mail:

Use:

▶ CLEAR    ▶ SEND

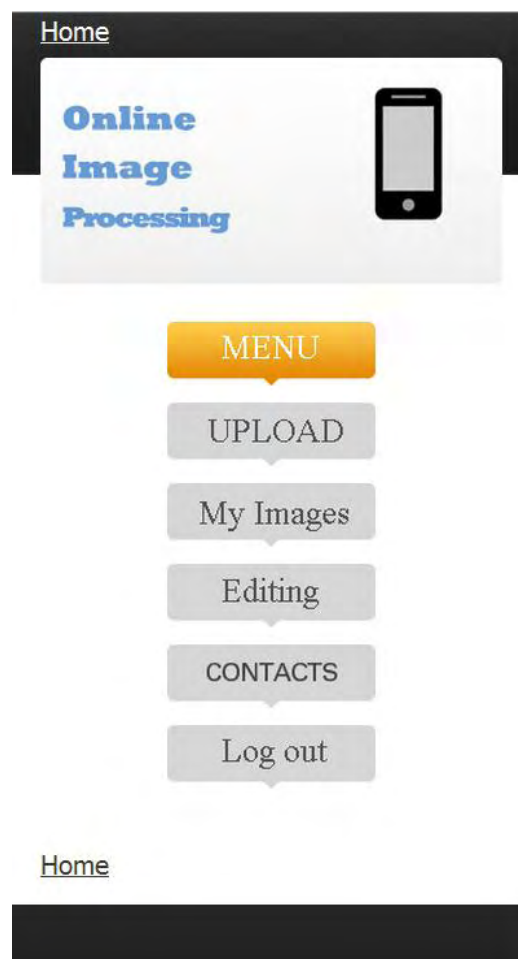
Back

**Σχήμα 20: Σελίδα εγγραφής χρήστη στο σύστημα**

Έχοντας γίνει δεκτή η αίτηση του χρήστη από τους διαχειριστές, ο χρήστης θα δεχθεί ένα μήνυμα ηλεκτρονικού ταχυδρομείου όπου θα τον πληροφορεί ότι η αίτησή

του έγινε δεκτή και μπορεί να χρησιμοποιήσει το σύστημα. Σε αυτό το μήνυμα θα περιέχεται και το όνομα χρήστη και ο κωδικός πρόσβασης στο σύστημα. Έτσι ο χρήστης έχοντας αυτά τα στοιχεία και ενεργό λογαριασμό στην υπηρεσία, μπορεί να επισκευτεί την ιστοσελίδα που είχε επισκεφτεί και νωρίτερα (Σχήμα 19) και συμπληρώνοντας τη φόρμα να εισέλθει στο σύστημα.

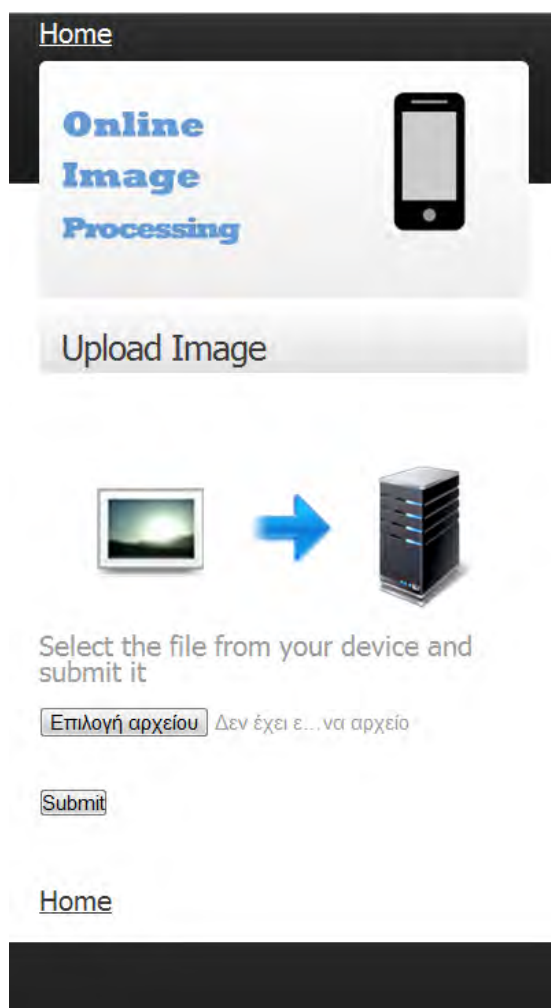
Εισάγοντας τα σωστά στοιχεία ο χρήστης εισέρχεται στη σελίδα που περιέχει τις κύριες επιλογές του συστήματος. Σε αυτή τη σελίδα εμφανίζεται το βασικό μενού του συστήματος με έξι επιλογές. Το μενού αυτό (Σχήμα 20) είναι απλό και αποτελείται από μια στήλη από επιλογές. Επιλέχτηκε αυτή υλοποίηση του μενού προκειμένου οι χρήστες να μπορούν να επιλεγούν τις λειτουργίες εύκολα από μικρές οθόνες αφής, το περιεχόμενο να είναι ευανάγνωστο και να υπάρχουν σχετικά μεγάλα κενά μεταξύ των επιλογών προκειμένου να μην επιλέγονται παραπάνω από μια λειτουργίες με ένα πάτημα του δακτύλου του χρήστη στην οθόνη.



Σχήμα 21: Σελίδα με βασικό μενού

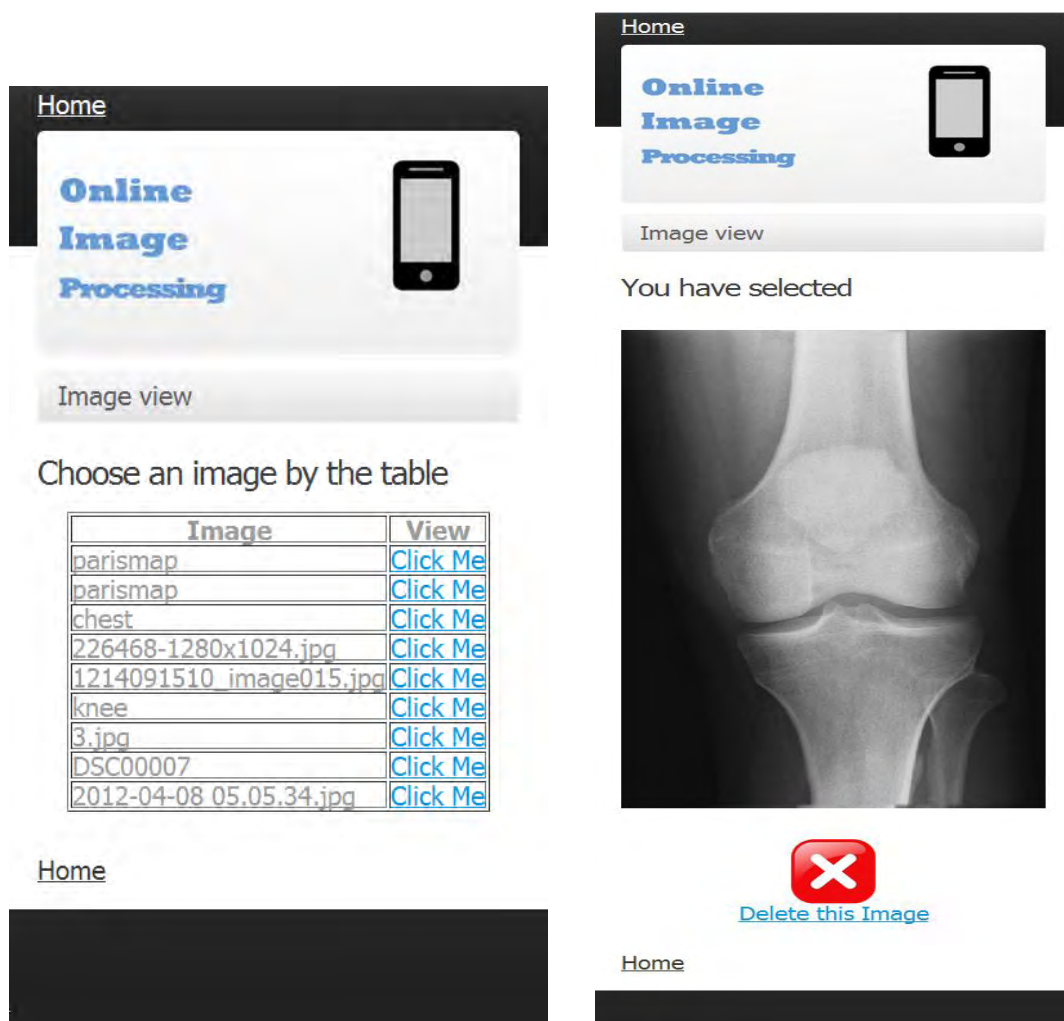
Οι λειτουργίες του μενού που εμφανίζεται στη παραπάνω εικόνα περιλαμβάνουν την αποστολή εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος, την προβολή των εικόνων που έχει ήδη αποστείλει στη βάση ο χρήστης, την επεξεργασία εικόνας, την επικοινωνία με τους διαχειριστές του συστήματος και την έξοδο του χρήστη από το σύστημα.

Επιλέγοντας την αποστολή εικόνας στη βάση (upload), εμφανίζεται μια νέα σελίδα (Σχήμα 22) όπου ο χρήστης επιλέγει την εικόνα που θέλει να αποστείλει από τη συσκευή του και έπειτα πατώντας την κατάλληλη επιλογή τη αποστέλλει στη βάση. Εάν η εικόνα έχει αποσταλθεί και αποθηκευτεί στη βάση με επιτυχία, εμφανίζεται μήνυμα επιβεβαίωσης αποστολής και αποθήκευσης. Ο χρόνος μεταφοράς του αρχείου από τη συσκευή του χρήστη στη βάση του συστήματος εξαρτάται από το μέγεθος του αρχείου και τη ταχύτητα του δικτύου.



Σχήμα 22: Σελίδα αποστολής αρχείου εικόνας από τη συσκευή του χρήστη στη βάση του συστήματος

Εάν ο χρήστης έχει ήδη αποστείλει εικόνες στη βάση του συστήματος, τότε μπορεί να προβάλει τις εικόνες που έχει αποστείλει επιλέγοντας τη λειτουργία My Images από το κεντρικό μενού. Από τη λίστα των εικόνων (Σχήμα 23 Αριστερή εικόνα) που θα του εμφανιστεί μπορεί να επιλέξει μια από αυτές για να τη δει. Αφού επιλέξει μια εικόνα από τη βάση τότε εμφανίζεται μια καινούργια σελίδα όπου προβάλλεται η εικόνα που επέλεξε (Σχήμα 23 Δεξιά εικόνα) και του δίνεται η δυνατότητα να διαγράψει την εικόνα που προβάλλεται από τη βάση του συστήματος. Ένα ο χρήστης επιλέξει να διαγράψει την εικόνα, εμφανίζεται μήνυμα επιβεβαίωσης προκειμένου να αποφθεχθούν οι κατά λάθος διαγραφές εικόνων.

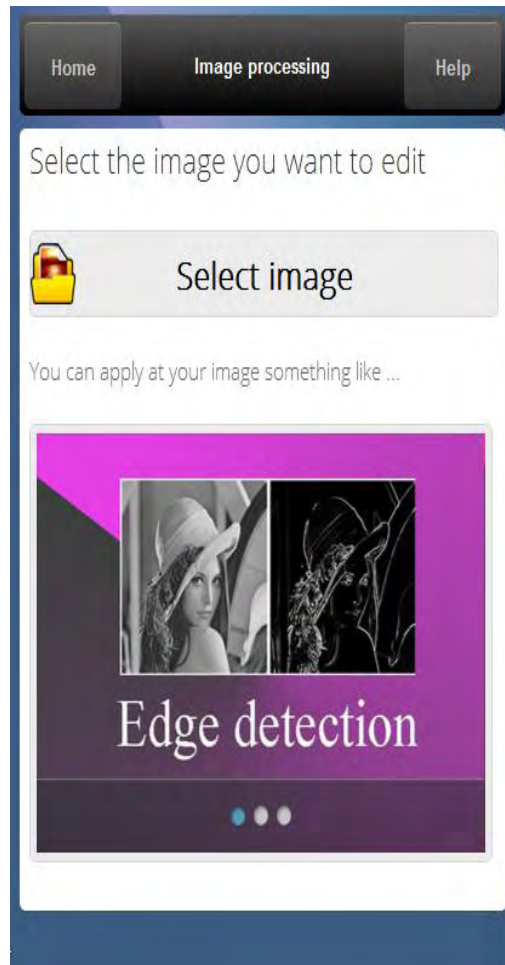


**Σχήμα 23:** Αριστερή εικόνα, σελίδα επιλογής εικόνας για προβολή. Δεξιά εικόνα, σελίδα προβολής εικόνας που επιλέχθηκε από τον χρήστη

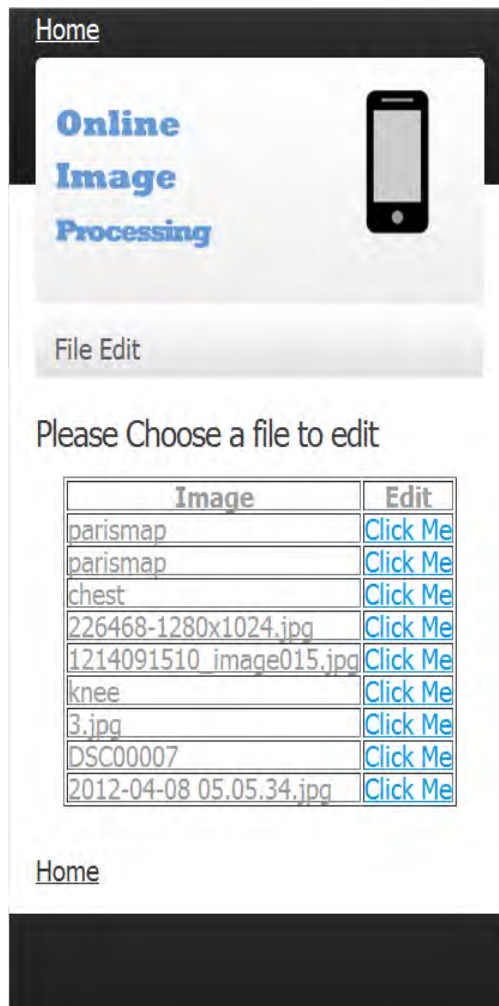
Η κύρια λειτουργία που εκτελεί το σύστημα είναι η επεξεργασία των εικόνων. Ο χρήστης για να επεξεργαστεί μια από τις εικόνες του, αρχικά θα πρέπει να επιλέξει



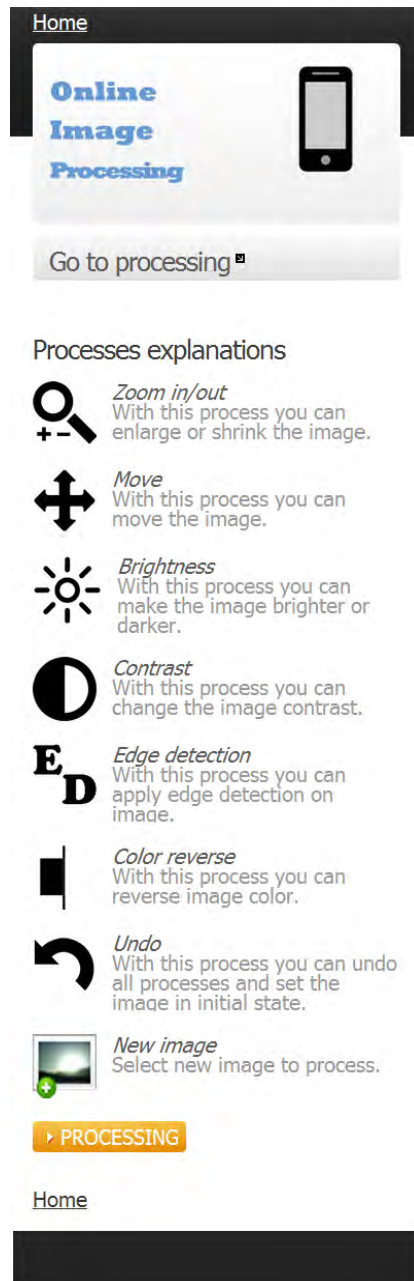
τη κατάλληλη λειτουργία από το κεντρικό μενού (Editing). Στη συνέχεια θα του εμφανιστεί μια σελίδα (Σχήμα 24) η οποία με περιεκτικό τρόπο και με τη χρήση εικόνων δείχνει στον χρήστη κάποιες από τις λειτουργίες του συστήματος. Από αυτή τη σελίδα μπορεί να ενημερωθεί για τις δυνατότητες επεξεργασίας του προγράμματος (Σχήμα 26) πατώντας την επιλογή Help και να επιλέξει μια εικόνα προς επεξεργασία, πατώντας την επιλογή Select image.



**Σχήμα 24:** Σελίδα επεξεργασίας εικόνας



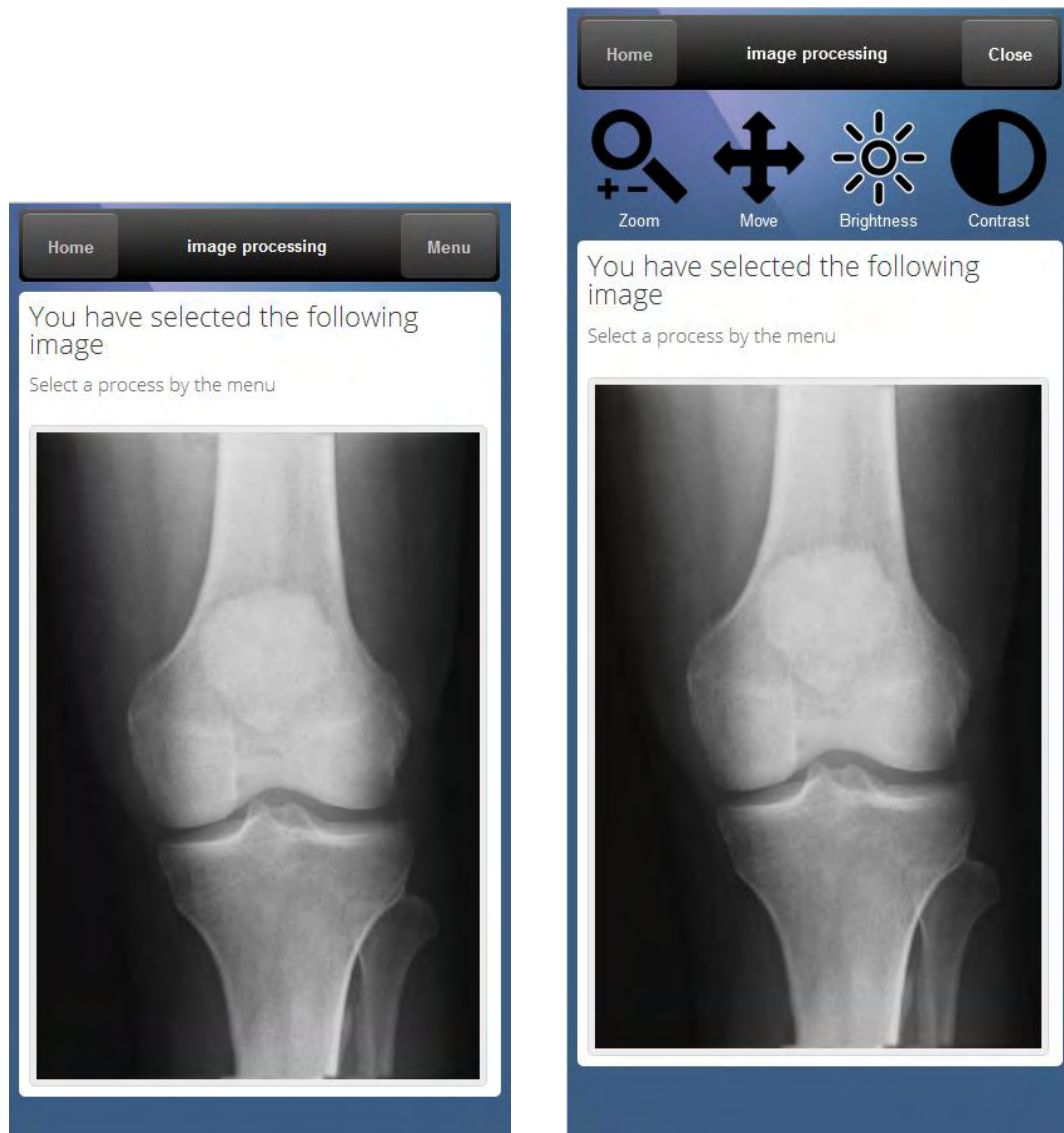
Σχήμα 25: Σελίδα επιλογής εικόνας προς επεξεργασία, από κατάλογο



Σχήμα 26: Σελίδα επεξήγησης των εικονιδίων επεξεργασίας εικόνας στον χρήστη

Μόλις ο χρήστης επιλέξει την εικόνα που θέλει, ανοίγει η σελίδα που φαίνεται στο Σχήμα 27 από την οποία θα μπορεί να επιλέξει την επεξεργασία που θα εφαρμόσει στην εικόνα. Η σχεδίαση αυτή της σελίδας όπως και ολόκληρου του συστήματος έγινε με σκοπό τη βέλτιστη αξιοποίηση των οθόνων αφής και των ιδιαίτερων χαρακτηριστικών τους. Ο χρήστης μπορεί να επιλέξει την επεξεργασία που θέλει από το αναδυόμενο μενού MENU. Πέραν από τις επιλογές για επεξεργασία, στο ίδιο μενού υπάρχουν και οι επιλογές αναιρέσεις όλως των επεξεργασιών που έχουν γίνει στην εικόνα, η προβολή της αρχικής ανεπεξεργαστης

εικόνας καθώς και η επιλογή για επεξεργασία νέας εικόνας. Τα περιεχόμενα αυτού του μενού εμφανίζονται στο πάνω μέρος της οθόνης και ο χρήστης μπορεί να περιηγηθεί σε αυτά σέρνοντας το δάκτυλό του πάνω από αυτά, μιμούμενος την κίνηση αλλαγής σελίδας βιβλίου.

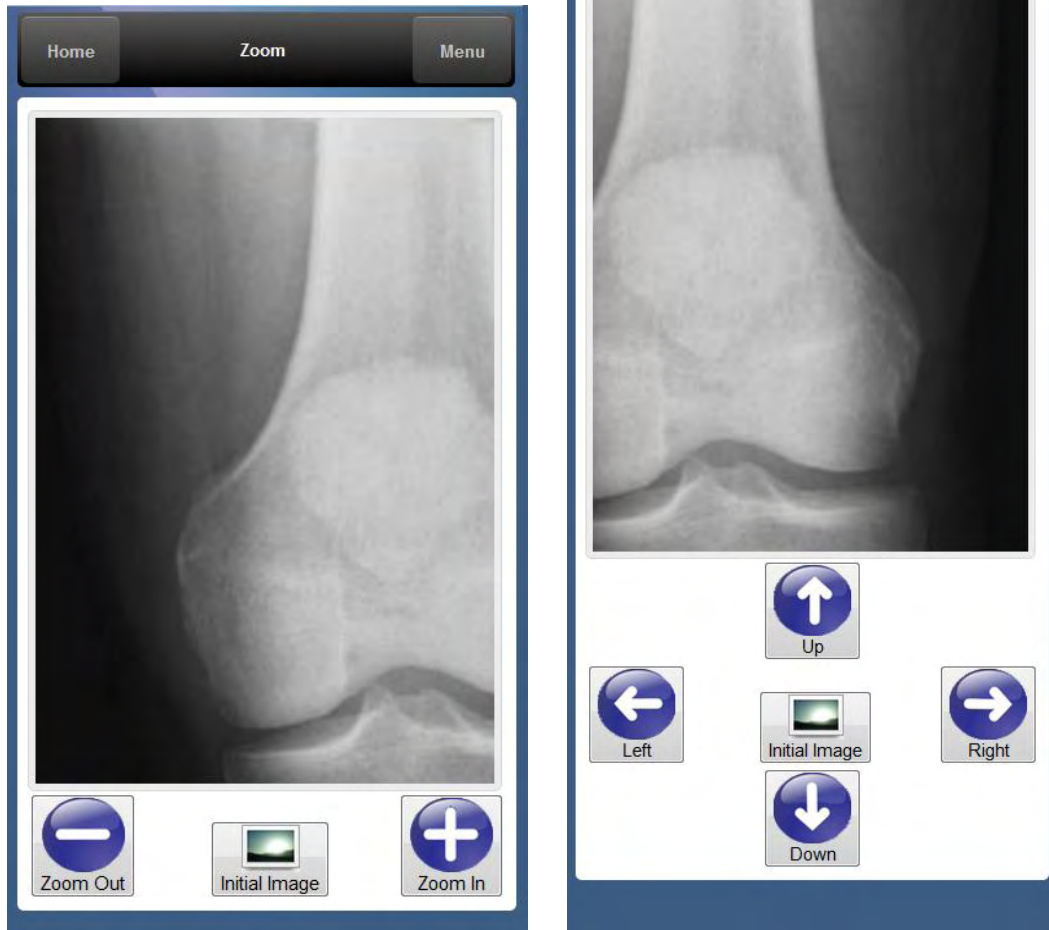


**Σχήμα 27:** Σελίδα προβολής και επεξεργασίας της επιλεγμένης εικόνας. Με προβολή του αναδυόμενου μενού (δεξιά εικόνα) και με απόκρυψη του αναδυόμενου μενού (αριστερή εικόνα)

Το παράθυρο από το οποίο μπορεί ο χρήστης να επιλέξει την επεξεργασία των εικόνων είναι απλό και το μενού είναι αναδυόμενο προκειμένου να μην δεσμεύει χώρο από τη οθόνη. Στο πάνω μέρος της σελίδας εκτός από την επιλογή για επιστροφή στην αρχική σελίδα, υπάρχει ένδειξη που πληροφορεί το χρήστη τι επεξεργασία έχει επιλέξει και εφαρμόζει εκείνη τη στιγμή, ενώ στο δεξιά μέρος της

πάνω μπάρας είναι το μενού που περιέχει τις άλλες δυνατές επεξεργασίες εικόνας. Η εικόνα που επεξεργάζεται ο χρήστης εκτείνεται στο μεγαλύτερο μέρος του χώρου της σελίδας και εκμεταλλεύεται πλήρως την οθόνη. Στο κάτω μέρος της οθόνης (Σχήμα 28) βρίσκονται τα εικονίδια τα οποία καθορίζουν τον βαθμό που εφαρμόζεται η επεξεργασία. Ο χρήστης έχει τη δυνατότητα να εφαρμόσει +- 5 στάθμες τη φορά, αλλά μπορεί να επιλέξει πάλι την ίδια επεξεργασία, και εφαρμόσει στην εικόνα την ίδια επεξεργασία όσες στάθμες θέλει.

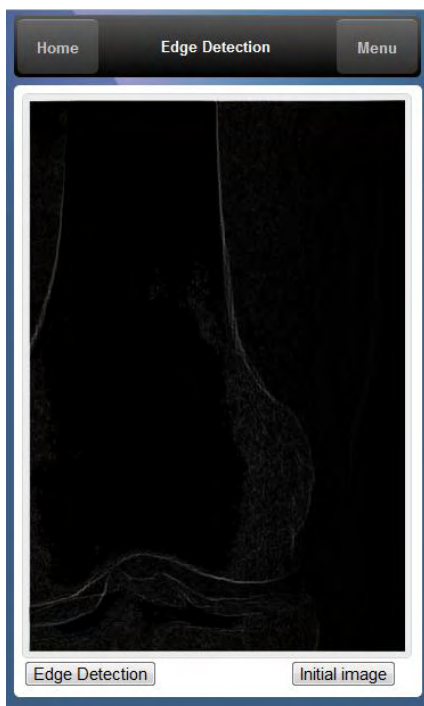
Οι σελίδες στις οποίες γίνεται επεξεργασία έχουν τον ίδιο σχεδιασμό, αλλά κάθε μια έχει διαφορετικά στοιχεία προκειμένου να ανταποκρίνεται στη κάθε επεξεργασία. Οι επεξεργασίες που μπορεί να εφαρμόσει ο χρήστης είναι η αλλαγή μεγέθους, μετακίνηση, αλλαγή φωτεινότητας, αλλαγή αντίθεσης, ανίχνευση ακμών και χρωματική αντιστροφή. Οι αλλαγές που κάνει ο χρήστης στην εικόνα δεν αποθηκεύονται στην εικόνα στη βάση του συστήματος και διαρκούν μέχρις ότου ο χρήστης αποσυνδεθεί από το σύστημα. Στις παρακάτω εικόνες φαίνεται πως υλοποιήθηκαν αυτές οι σελίδες και οι δυνατότητες που παρέχουν, έχοντας ως παράδειγμα μια ενδεικτική εικόνα που ο αρχικός της τύπος ήταν DICOM.



Σχήμα 28: Αριστερή εικόνα, σελίδα αλλαγής μεγέθους εικόνας. Δεξιά εικόνα, σελίδα μετακίνησης εικόνας



Σχήμα 29: Αριστερή εικόνα, σελίδα αλλαγής φωτεινότητας εικόνας. Δεξιά εικόνα, σελίδα αλλαγής αντίθεσης εικόνας



Σχήμα 30: Αριστερή εικόνα, σελίδα εύρεσης ακμών εικόνας. Δεξιά εικόνα, σελίδα χρωματικής αντιστροφής εικόνας

Ακόμη μια δυνατότητα του συστήματος είναι η επικοινωνία του χρήστη με του διαχειριστές του συστήματος. Αυτό γίνεται επιλέγοντας από το κύριο μενού τη λειτουργία επικοινωνία Contact. Στη σελίδα που εμφανίζεται (Σχήμα 31) μπορεί να συμπληρώσει την ειδική φόρμα και να αποστείλει τις παρατηρήσεις, σχόλια, προβλήματα και οτιδήποτε άλλο σχετικό με την εφαρμογή στους διαχειριστές του συστήματος.

Home

**Online  
Image  
Processing**

Info

Contact information

*Online Image Processing Inc.  
Tsimiski 0, Thessaloniki,  
GREECE.*

Telephone: +30 2310 123456  
FAX: +30 2310 123456  
E-mail: [info@onimpro.com](mailto:info@onimpro.com)

Contact form

Name:

E-mail:

Message:

Home

Σχήμα 31: Σελίδα επικοινωνίας με τους διαχειριστές του συστήματος

Τέλος ο χρήστης ολοκληρώνοντας την περιήγησή του στο σύστημα μπορεί να αποσυνδεθεί από το σύστημα πατώντας την τελευταία λειτουργία του κύριου μενού Log out. Σε περίπτωση που ο χρήστης φύγει απλώς από τη σελίδα του συστήματος χωρίς να κάνει αποσύνδεση, τα δεδομένα που προέκυψαν από τη χρήση του

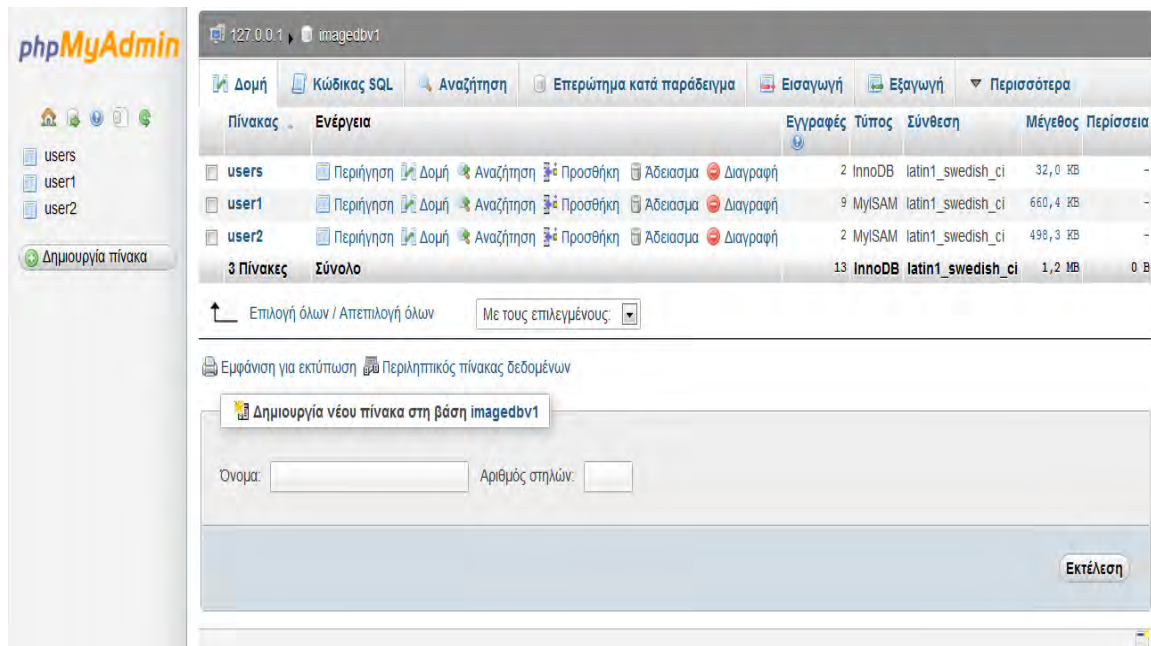


συστήματος μένουν στον εξυπηρετητή και αργότερα μπορεί να επανέλθει στο σημείο όπου ήταν. Αν ο χρήστης αποσυνδεθεί από το σύστημα τότε όλα τα δεδομένα που έχουν δημιουργηθεί κατά τη τελευταία συνεδρία με το σύστημα διαγράφονται και εμφανίζεται η αρχική σελίδα, όπου μπορεί να συνδεθεί και πάλι ή να επισπευτεί άλλη σελίδα.

### **Διαχειριστής**

Οι διαχειριστές θα έχουν πρόσβαση στη βάση δεδομένων του συστήματος μέσω φυλλομετρητή, επισκεπτόμενοι τη κατάλληλη σελίδα από οποιαδήποτε συσκευή που συνδέεται στο διαδίκτυο. Το πρόγραμμα που χρησιμοποιείται για τη διαχείριση της βάσης είναι το `phpmyadmin` [16]. Πρόκειται για ένα δωρεάν εργαλείο λογισμικού γραμμένο σε `php`, που προσφέρει υπηρεσίες για τη διαχείριση της βάσης μέσω του διαδικτύου. Το `phpmyadmin` υποστηρίζει ένα ευρύ φάσμα λειτουργιών με τη χρήση `MySQL`. Από το περιβάλλον χρήστη, υποστηρίζονται οι πιο συχνά χρησιμοποιούμενες λειτουργίες (διαχείριση βάσεων δεδομένων, πινάκων, πεδίων, συσχετίσεων, ευρετηρίων, χρηστών, αδειών, κτλ.), ενώ υπάρχει και η δυνατότητα άμεσης εκτέλεσης εκφράσεων `SQL`.

Οι διαχειριστές πληκτρολογώντας τη κατάλληλη διεύθυνση στο φυλλομετρητή μπορούν να επισπευτούν τη σελίδα διαχείρισης της βάσης. Από εκεί μπορούν να επιλέξουν τη δημιουργία νέου πίνακα χρήστη για την εισαγωγή νέου χρήστη στο σύστημα, ή να επεξεργαστούν τα ήδη υπάρχοντα στοιχεία στη βάση και να αλλάξουν τα δεδομένα που αφορούν κάποια εικόνα χρήστη, ή δεδομένα που αφορούν τον ίδιο τον χρήστη. Επίσης μπορούν να εκτελέσουν λειτουργίες συντήρησης και αναβάθμισης του συστήματος.



Σχήμα 32: Περιβάλλον phpmyadmin

## 5.4 Βάση Δεδομένων

Η βάση δεδομένων του συστήματος έχει υλοποιηθεί έτσι ώστε να είναι εύκολα επεκτάσιμη και συντηρήσιμη. Στη βάση υπάρχει ένα πίνακας ο οποίος διατηρεί τα στοιχεία του κάθε χρήστη και επιμέρους πίνακες οι οποίοι περιέχουν τα δεδομένα που έχει αποστείλει ο κάθε χρήστης στη βάση του συστήματος.



Σχήμα 33: Πίνακας με χρήστες

Ο πίνακας users όπως φαίνεται στο Σχήμα 33 είναι ένας βασικός πίνακας ο οποίος περιέχει τα κύρια στοιχεία που πρέπει να κρατά το σύστημα για τους χρήστες. Το κλειδί σε αυτό το πίνακα είναι σύνθετο, τα πεδία που απαρτίζουν το κλειδί είναι η ταυτότητα και το όνομα του χρήστη (το σύστημα απαιτεί το όνομα χρήστη νέων

χρηστών να είναι διαφορετικό από αυτά που ήδη υπάρχουν, δηλαδή δεν επιτρέπεται δύο χρήστες να έχουν το ίδιο όνομα χρήστη), τα οποία χαρακτηρίζουν μοναδικά τον κάθε χρήστη, ενώ η ταυτότητα παράγεται με αυτόματο τρόπο με την εισαγωγή νέου χρήστη.



Σχήμα 34: Πίνακας χρήστη

Με την εισαγωγή κάθε νέου χρήστη οι διαχειριστές του συστήματος θα πρέπει να δημιουργούν ένα νέο πίνακα για αυτόν τον χρήστη. Ο πίνακας αυτός θα έχει τίτλο το όνομα του χρήστη και θα περιλαμβάνει στοιχεία που αφορούν τα δεδομένα εικόνας που αποστέλλει ο χρήστης στη βάση. Όπως φαίνεται στο Σχήμα 34 κάθε πίνακας χρήστη έχει κλειδί τη ταυτότητα της εικόνας, ενώ εκτός από την εικόνα αποθηκεύονται και πληροφορίες που αφορούν τη κάθε εικόνα και μπορεί να χρησιμοποιηθούν κατά την επεξεργασία.

Για κάθε νέο χρήστη οι διαχειριστές εκτός από τη δημιουργία πίνακα με το όνομα του χρήστη θα πρέπει να ενημερώνουν και τον πίνακα που περιέχει τους χρήστες και να δημιουργούν με αυτόματο τρόπο το κωδικό χρήστη τον οποίο στη συνέχεια θα αποστέλλουν και στο χρήστη.

## 5.5 Το Υπολογιστικό Σύστημα που χρησιμοποιήθηκε

Το υπολογιστικό σύστημα που χρησιμοποιήθηκε είναι ένα οικονομικό σύστημα οικιακής χρήσης και αποτελείται από τον επεξεργαστή Intel Pentium D 930 [18] και την κάρτα γραφικών Nvidia Geforce 9500 Gt [17].

### 5.5.1 Κάρτα Γραφικών Nvidia Geforce 9500 Gt

Η συγκεκριμένη κάρτα γραφικών έχει υπολογιστική δυνατότητα 1.1 και 32 ομάδες μικροεπεξεργαστών.

Αναλυτικά οι δυνατότητες αυτές, μεταφράζονται ως εξής:

- Ο αριθμός των νημάτων σε κάθε μπλοκ μπορεί να είναι μέχρι 512.
- Η μέγιστη διάσταση ενός μπλοκ μπορεί να είναι κατά x 512 , κατά y 512 και κατά z 64.
- Η μέγιστη τιμή της κάθε διάστασης του συνολικού πλέγματος ( grid ) μπορεί να είναι 65535.
- Το κάθε warp (όπως αναφέρεται και στο κεφάλαιο A.2 όπου περιγράφεται η αρχιτεκτονική Cuda) μπορεί να περιέχει μέχρι 32 νήματα.
- αριθμός των registers κάθε ομάδας πολυεπεξεργαστών είναι 8192.
- Το μέγεθος της shared μνήμης κάθε ομάδας πολυεπεξεργαστών είναι 16KB .
- Το συνολικό μέγεθος της Constant μνήμης είναι 64KB.
- Κάθε ομάδα πολυεπεξεργαστών αποτελείται από 8 βαθμωτούς επεξεργαστές έτσι ώστε όλη η ομάδα να μπορεί να επεξεργαστεί 32 threads σε 4 κύκλους του ρολογιού.

Οι όροι που αναφέρονται σε αυτό το σημείο για να περιγράψουν τις δυνατότητες της κάρτας, αναλύονται στα παραρτήματα όπου περιγράφεται η αρχιτεκτονική και η γλώσσα προγραμματισμού Cuda.

### 5.5.2 Επεξεργαστής Intel Pentium D 930

Ο Intel Pentium D 930 είναι ένας διπύρηνος επεξεργαστής με συχνότητα 3.000 Mhz. Ακόμη ένα χαρακτηριστικό αυτού του μοντέλου είναι ότι διαθέτει προσωρινή μνήμη L2 cache μεγέθους 2 MB ανά πυρήνα.

## Κεφάλαιο 6

### Αξιολόγηση και Συμπεράσματα

---

#### 6.1 Αξιολόγηση

#### 6.2 Συμπεράσματα

#### 6.3 Μελλοντική εργασία

---

### 6.1 Αξιολόγηση

Στην συνέχεια θα περιγραφούν τα κυριότερα συμπεράσματα που εξήχθησαν από τη δημιουργία, λειτουργία και έλεγχο του συστήματος παράλληλης επεξεργασίας εικόνων υψηλής ανάλυσης μέσω διαδικτύου.

Η χρήση του συστήματος από φορητές συσκευές έγινε με επιτυχία. Το περιεχόμενο της ιστοσελίδας του συστήματος προβαλλόταν χωρείς να αποκρύπτεται κάποια πληροφορία λόγω του μικρού μεγέθους της οθόνης. Επιπλέον οι επιλογές ήταν διακριτές και εύκολα επιλέξιμες από μικρές οθόνες αφής. Η ιστοσελίδα του συστήματος ήταν εύχρηστη και από συσκευές με μεγαλύτερες οθόνες όπως σταθεροί και φορητοί υπολογιστές, και ο ορισμός ενός μέγιστου επιτρεπόμενου μεγέθους εικόνας έκανε το σύστημα λειτουργικό ακόμη και για τις μεγαλύτερες οθόνες, ξεπερνώντας τα προβλήματα που θα προέκυπταν από τη προσαρμογή της εικόνας στο μέγεθος της οθόνης και τη παραγωγή πολύ μεγάλων εικόνων που θα απαιτούσαν πολλαπλάσιο χρόνο για επεξεργασία και ως αποτέλεσμα το σύστημα θα γινόταν απαγορευτικά αργό.

Η υπολογιστική δύναμη του συστήματος στο οποίο υλοποιήθηκε το σύστημα επεξεργασίας εικόνων υψηλής ανάλυσης είναι μικρή, ωστόσο η επεξεργασία των εικόνων γινόταν με ικανοποιητική ταχύτητα και προσέφερε στο χρήστη ευχάριστη περιήγηση. Οι τεχνικές προ-επεξεργασίας που χρησιμοποιήθηκαν προ-υπολογίζουν

τις απαιτούμενες εικόνες έγκαιρα και συμβάλουν στην αύξηση της ταχύτητας απόκρισης του συστήματος.

Η κωδικοποίηση του περιεχομένου της ιστοσελίδας του συστήματος, η αποστολή στιγμιότυπων της εικόνας και όχι ολόκληρης της εικόνας από τη βάση, η κωδικοποίηση του κωδικού χρήστη κατά την αποστολή από τη συσκευή του χρήστη στο σύστημα για επιβεβαίωση και η χρήση τειχών προστασίας κρίνονται ως ικανοποιητικά μέτρα για τη διασφάλιση των δεδομένων των χρηστών τόσο στην αποθήκευση όσο και στη μεταφορά των δεδομένων, ακόμη και για συστήματα που απαιτούν υψηλή ασφάλεια όπως ένα σύστημα με ιατρικά δεδομένα.

## **6.2 Συμπεράσματα**

Η εφαρμογή που αναπτύχθηκε θα μπορούσε να χρησιμοποιηθεί είτε από μόνη της είτε σε συνδυασμό με κάποιο άλλο σύστημα καθώς το σύστημα προσφέρει υπηρεσίες αποθήκευσης, προβολής και επεξεργασίας για ένα μεγάλο εύρος τύπων εικόνας. Σε αυτή την εργασία μελετήθηκε η περίπτωση προσκόμισης και επεξεργασίας εικόνων από ιατρικό φάκελο [12] με θετικά αποτελέσματα.

Το σύστημα συμβαδίζοντας με την τάση της εποχής προσφέρει υπηρεσίες που είναι διαθέσιμες στο χρήστη από παντού. Δίνει τη δυνατότητα για πρόσβαση στη πληροφορία από παντού και για επεξεργασία εικόνων από οποιαδήποτε συσκευή που συνδέεται στο διαδίκτυο.

## **6.3 Μελλοντικά**

Το σύστημα που αναπτύχθηκε υλοποιεί τις βασικές λειτουργίες επεξεργασίας εικόνας και έχει αναπτυχθεί ώστε να είναι εύχρηστο από τους χρήστες φορητών συσκευών. Ωστόσο υπάρχουν πεδία τα οποία μπορούν να αναπτυχθούν και να εξελιχθούν στο μέλλον.

Η χρησιμοποίηση των μέχρι τώρα βιβλιοθηκών της Nvidia περιόρισαν το σύστημα ώστε να επεξεργάζεται ασπρόμαυρες εικόνες. Η επεξεργασία έγχρωμων

εικόνων είναι μια σημαντική δυνατότητα που θα μπορούσε να ενταχθεί στο μέλλον προκειμένου να κάνει το σύστημα να απευθύνεται σε περισσότερους χρήστες. Σε αυτή τη πτυχιακή εργασία μελετήθηκε η επεξεργασία ιατρικών εικόνων οι οποίες ως επί το πλείστον είναι ασπρόμαυρες, έτσι το σύστημα δεν εμφάνισε κανένα πρόβλημα.

Επίσης οι επεξεργασίες που υποστηρίζονται από το σύστημα είναι οι βασικές επεξεργασίες εικόνας. Θα ήταν καλό να εισαχθούν και νέες δυνατότητες επεξεργασίας εμπλουτίζοντας το σύστημα με νέες δυνατότητες.

Το σύστημα δοκιμάστηκε και πέτυχε να επεξεργαστεί με αποτελεσματικότητα ιατρικές εικόνες οι οποίες ήταν στη μορφή DICOM [22]. Θα ήταν χρήσιμο εάν στο μέλλον αυτό το σύστημα γινόταν μέρος του ευρύτερου συστήματος ενός νοσοκομείου ή κλινικής. Θα μπορούσε να χρησιμοποιηθεί σε κάποιο νοσοκομείο σε άλλα συστήματα άλλων οργανισμών καθώς υποστηρίζει πληθώρα μορφών εικόνας και έτσι ξεπερνά τις ασυμβατότητες που προκύπτουν από τους διάφορους τύπους εικόνων.

Τέλος θα ήταν καλό να δίνεται η δυνατότητα χρήσης των διαθέσιμων υπολογιστικών συστημάτων που συνδέονται στο δίκτυο και διαθέτουν κάρτα γραφικών που υποστηρίζει cuda. Π.χ. η διάθεση και χρησιμοποίηση όταν κρίνεται αναγκαίο των υπολογιστών που έχουν cuda και βρίσκονται στο δίκτυο ενός νοσοκομείου, δίνοντας ιδιαίτερη προσοχή στα δεδομένα που διαμοιράζονται για να αποφευχθούν προβλήματα ασφαλείας.

# Παράρτημα Α

## Αρχιτεκτονική CUDA

---

A.1 Εισαγωγή

A.2 Οργάνωση και εκτέλεση νημάτων

A.3 Πυρήνες-Επεξεργαστές της Κάρτας Γραφικών

A.4 Εκτέλεση των νημάτων

A.5 Θέσεις Μνήμης της Κάρτας

---

### **A.1 Εισαγωγή**

Στο κεφάλαιο 3.4 έχει περιγραφεί με συνοπτικό τρόπο η αρχιτεκτονική των καρτών γραφικών, οι διαφορές από τη κεντρική μονάδα επεξεργασίας και ο λόγος για τον οποίο είναι τόσο αποδοτική και επιλέχθηκε να αξιοποιηθεί σε αυτό το σύστημα. Σε αυτό το παράρτημα θα αναλυθεί περαιτέρω η βασική δομή αυτής της αρχιτεκτονικής και θα περιγραφούν οι βασικές λειτουργίες που εκτελούνται στη κάρτα γραφικών [19].

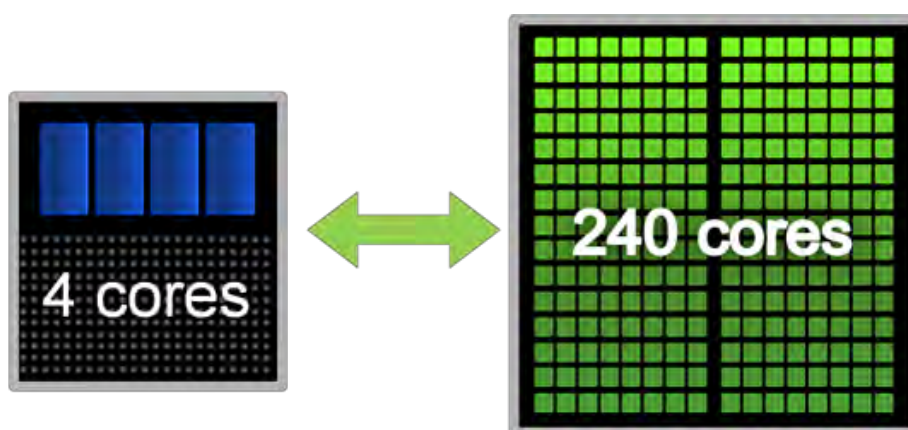
### **A.2 Οργάνωση των Εκτελέσιμων Νημάτων**

Κάθε πρόγραμμα το οποίο εκτελείται στη κάρτα γραφικών αποτελείται από μια ακολουθία εντολών οι οποίες εκτελούνται σειριακά, η μια μετά την άλλη. Η ακολουθία εκτελείται πάρα πολλές φορές κατά τη διάρκεια του προγράμματος αλλά κάθε φορά για διαφορετικά δεδομένα εισόδου ώστε να προκύπτουν αντίστοιχα διαφορετικά αποτελέσματα. Αυτή η ακολουθία ενεργειών, από ονομάζεται νήμα



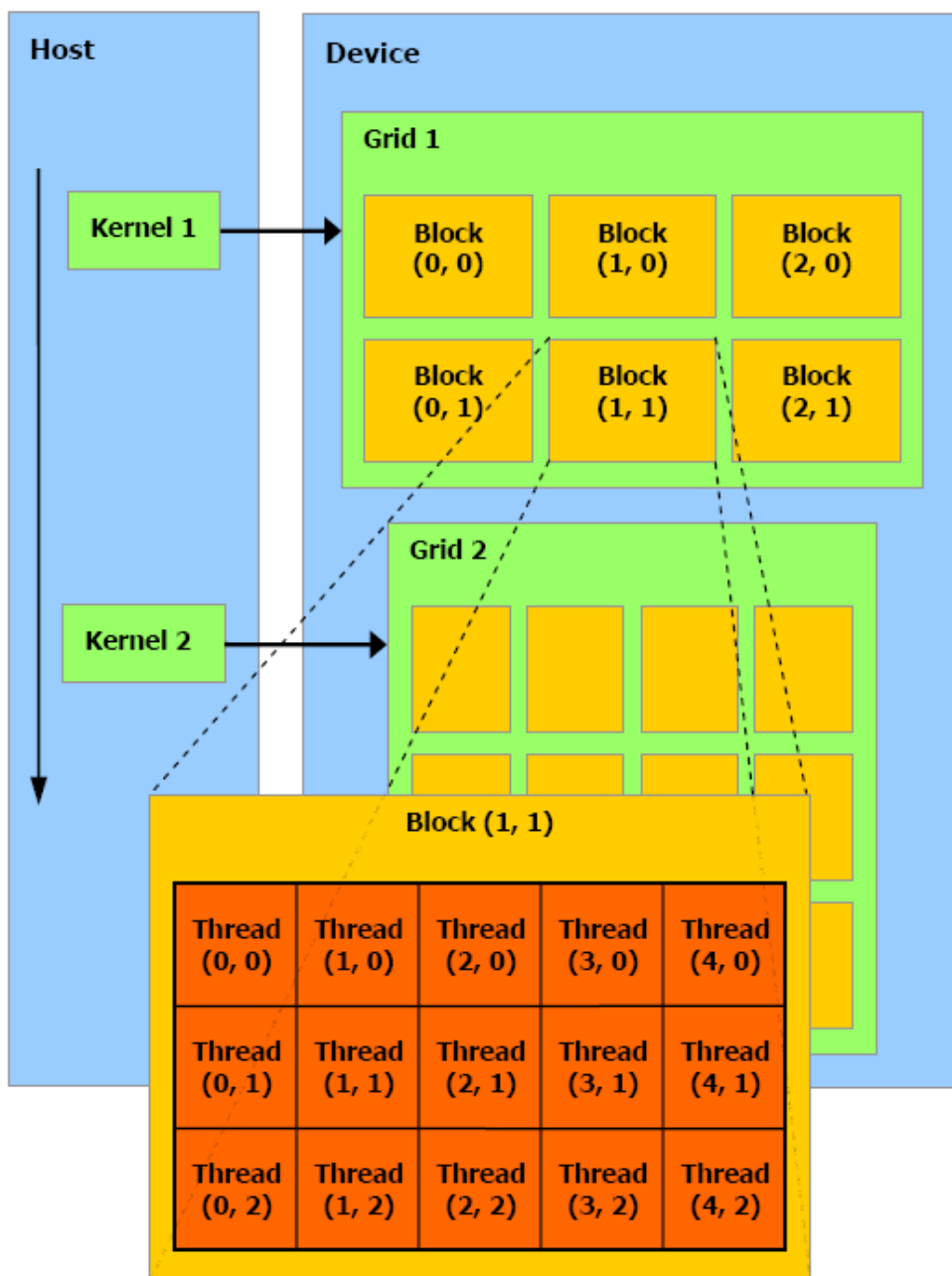
(thread). Πολλά νήματα συγκροτούν μια υπορουτίνα του αρχικού προγράμματος η οποία θα σταλεί στην συσκευή και ονομάζεται kernel ή πυρήνας εκτέλεσης.

Το χαρακτηριστικό που αξιοποιείται στη χρήση της κάρτα γραφικών είναι η παράλληλη αποστολή και εκτέλεση νημάτων, μειώνοντας τον χρόνο εκτέλεσης σημαντικά σε σχέση με την σειριακή εκτέλεση των ίδιων νημάτων στον επεξεργαστή. Έτσι ιδιαίτερα σημαντικός είναι ο αριθμός των πυρήνων (cores). Λαμβάνοντας υπόψη πως οι επεξεργαστές τελευταίας τεχνολογίας μόλις έφτασαν τους 4 με 8 πυρήνες, και ότι οι αντίστοιχες κάρτες γραφικών έχουν ήδη αγγίξει τους 240 πυρήνες, γίνεται κατανοητός ένας από τους λόγους της υψηλής υπολογιστικής ισχύς των καρτών γραφικών.



Σχήμα 35: Πυρήνες κεντρικής μονάδας επεξεργασίας και μονάδας επεξεργασίας γραφικών [28]

Το σύνολο των νημάτων που εκτελούνται σε κάθε πυρήνα είναι οργανωμένα σε επιμέρους ομάδες, μπλοκ (blocks). Αυτά, στη συνέχεια, οργανώνονται ως τμήματα του συνολικού υπολογιστικού πλέγματος (grid) (Σχήμα 36).



Σχήμα 36: ομαδοποίηση και οργάνωση των νημάτων [29]

Το κάθε νήμα χαρακτηρίζεται από δύο αριθμούς που δείχνουν τη θέση του μέσα στο μπλοκ, το κάθε μπλοκ χαρακτηρίζεται και αυτό από δύο αριθμούς που θα δείχνουν τη θέση του μέσα στο πλέγμα. Αυτή η διάταξη των νημάτων και η αρίθμηση τους χρησιμεύει ώστε το εκτελέσιμο πρόγραμμα να ξέρει σε κάθε στιγμή σε ποιο νήμα θα αναφέρεται για μεταφορά δεδομένων και για εκτέλεση. Παράλληλα υπάρχει και ένας άλλος αριθμός, ο threadID ο οποίος δείχνει μονοσήμαντα τη θέση του νήματος μέσα σε μπλοκ έως και 3 διαστάσεων. Π.χ. για ένα μονοδιάστατο μπλοκ

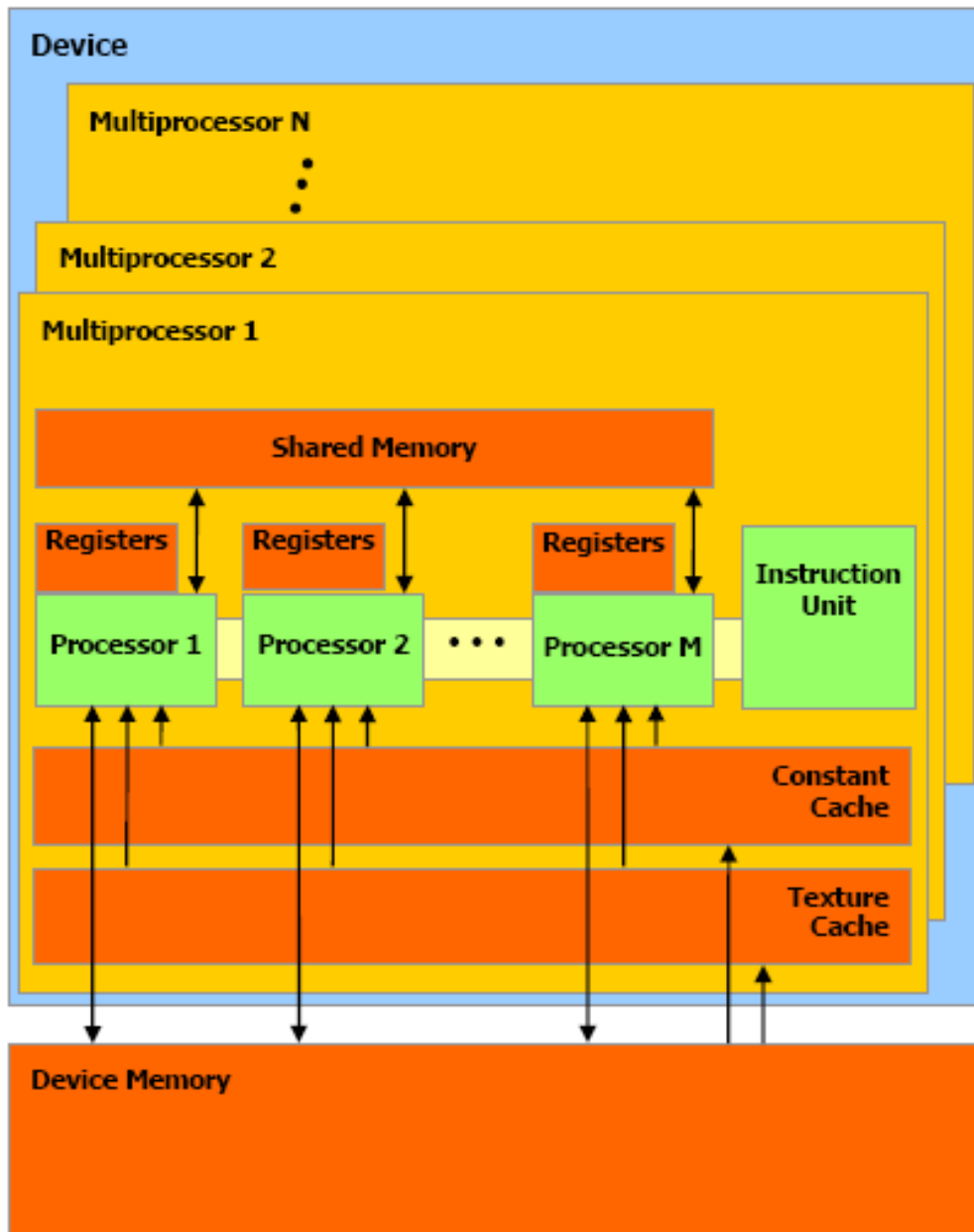
μήκους  $D_x$ , το νήμα με θέση  $(x)$  θα έχει threadID ίσο με  $(x)$ , για ένα διδιάστατο μπλοκ μήκους  $D_x$  και πλάτος  $D_y$ , το νήμα με θέση  $(x,y)$  θα έχει threadID ίσο με  $(x + y \cdot D_x)$  και τέλος για ένα τριδιάστατο μπλοκ με μήκος  $D_x$ , πλάτος  $D_y$  και ύψος  $D_z$ , το νήμα με θέση  $(x,y,z)$  θα έχει threadID ίσο με  $(x + y \cdot D_x + z \cdot D_x \cdot D_y)$ . Αντίστοιχα μπορεί να οριστεί και ένας αριθμός blockID ο οποίος δείχνει μονοσήμαντα τη θέση ενός μπλοκ μέσα σε πλέγμα έως και 2 διαστάσεων.

Τα νήματα που είναι οργανωμένα σε μπλοκ έχουν το πλεονέκτημα να χρησιμοποιούν την κοινή μνήμη του μπλοκ (shared memory) (Σχήμα 37) η οποία προσφέρει πιο γρήγορη μετάδοση δεδομένων, έτσι μπορούν να πετύχουν ακόμα μεγαλύτερη επιτάχυνση της εκτέλεσης της εφαρμογής.

Ένα στοιχείο που πρέπει να ληφθεί σοβαρά υπόψη κατά την ανάπτυξη του λογισμικού είναι ότι ο αριθμός των διαθέσιμων νημάτων αλλά και των μπλοκ είναι συγκεκριμένος και διαφέρει από κάρτα σε κάρτα γραφικών. Για αυτό το λόγο η ανάπτυξη του λογισμικού θα πρέπει να γίνεται με βάση τη κάρτα γραφικών που θα εκτελείται ή να προσαρμόζεται στις δυνατότητες της κάθε κάρτας γραφικών που εκτελείται, αν πάλι η συσκευή δεν έχει δυνατότητες παραλληλίας που να ανταποκρίνονται στο λογισμικό, το πρόγραμμα θα εκτελέσει αυτόματα, ομάδες μπλοκ σειριακά, όπου τα νήματα του κάθε μπλοκ θα εκτελούνται παράλληλα.

### **A.3 Πυρήνες-Επεξεργαστές της Κάρτας Γραφικών**

Όταν μια εφαρμογή προγραμματισμένη σε CUDA φτάσει στο σημείο να κάνει κλήση ενός kernel για εκτέλεση στη συσκευή, τα μπλοκ του συνολικού πλέγματος, αριθμούνται όπως αναφέρεται στο κεφάλαιο A.2 και διαμοιράζονται ανά ένα σε κάθε ομάδα πολυεπεξεργαστών (Multithreaded Streaming Multiprocessors) που είναι διαθέσιμοι. Στη συνέχεια κάθε ξεχωριστό νήμα εκτελείται σε κάθε πολυεπεξεργαστή ξεχωριστά και ταυτόχρονα με όλα τα νήματα του ίδιου μπλοκ. Από εκεί και πέρα μόλις ένα υπολογιστικό μπλοκ τερματίσει την εκτέλεση του, τη θέση του παίρνει το επόμενο.



Σχήμα 37: Το μοντέλο των επεξεργαστών [19]

Κάθε ομάδα πολυεπεξεργαστών (multiprocessors) όπως φαίνεται στο Σχήμα 37 αποτελείται από  $N$  πυρήνες βαθμωτών επεξεργαστών (Scalar Processor Cores), δύο ειδικές μονάδες μνήμης (Constant Cache, Texture Cache), μια κοινή μνήμη (Shared Memory) και μια μονάδα ελέγχου ροής και εντολών που είναι υπεύθυνη για τη δημιουργία, τη διαχείριση και την εκτέλεση των παράλληλων νημάτων στη συσκευή με μηδενική επιβάρυνση στην απόδοση αυτής. Οι θέσεις μνήμης που αναφέρθηκαν και φαίνονται στην εικόνα θα περιγραφούν αναλυτικότερα στη συνέχεια.

## A.4 Εκτέλεση των νημάτων

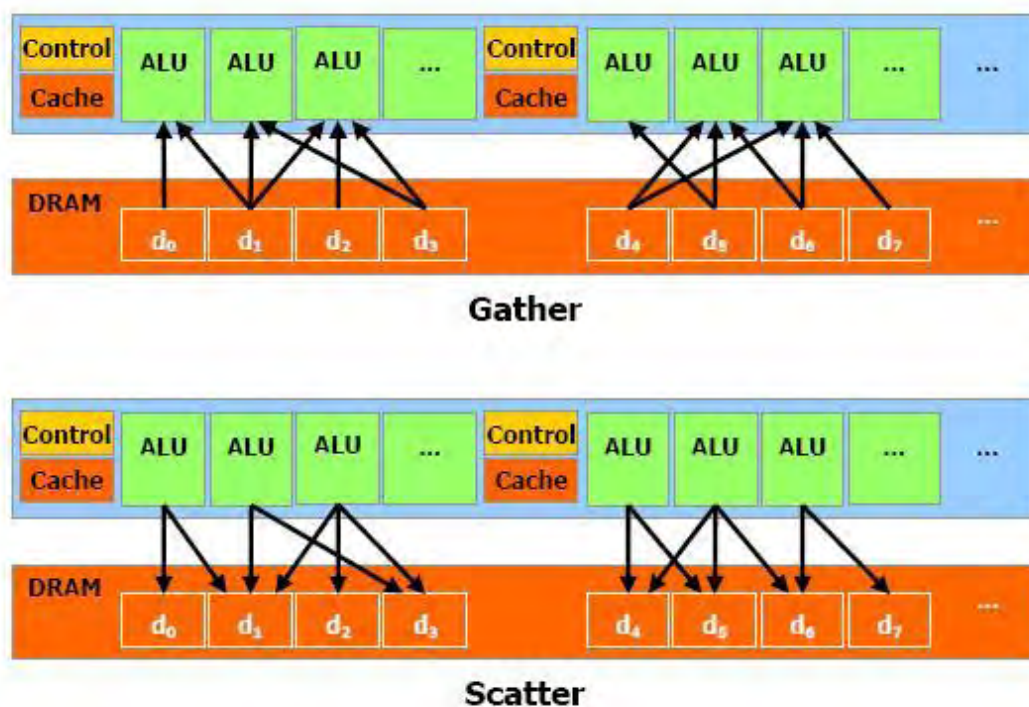
Κατά την εκτέλεση των νημάτων ο πολυεπεξεργαστής αντιστοιχεί καθένα από αυτά σε κάθε βαθμωτό επεξεργαστή όπου πλέον θα εκτελείται ανεξάρτητα, με τη δική του διεύθυνση εντολών και κατάσταση καταχώρησης. Για να μπορέσουν να διαχειριστούν και να οργανωθούν τα εκατοντάδες νήματα των διάφορων προγραμμάτων που τρέχουν κάθε στιγμή στη συσκευή, οι πολυεπεξεργαστές λειτουργούν κάτω από έναν νέο σύστημα το SIMT (Single Instruction Multiple Thread) σύμφωνα με το οποίο, μπορεί να επεξεργαστεί με την ίδια εντολή, διαφορετικά εισερχόμενα δεδομένα, σε αντίθεση με τον κεντρικό επεξεργαστή που είναι της μορφής MIMD (Multiple Instruction Multiple Data) όπου κάθε πυρήνας εκτελεί διαφορετική λειτουργία σε διαφορετικά δεδομένα. Ο σκοπός του SIMT είναι να δημιουργεί να διαχειρίζεται να προγραμματίζει και να εκτελεί νήματα σε ομάδες των 32 παράλληλων νημάτων, οι οποίες ονομάζονται warps (στημόνι). Κύριο χαρακτηριστικό είναι ότι κάθε ανεξάρτητο νήμα που αποτελεί μέλος ενός warp, ξεκινά από την ίδια διεύθυνση του προγράμματος μαζί με τα υπόλοιπα αλλά από εκεί και πέρα είναι ελεύθερο να εκτελεστεί ανεξάρτητα. Έτσι όταν ένα μπλοκ από νήματα στέλνεται σε μια ομάδα πολυεπεξεργαστών, η μονάδα SIMT τα διαχωρίζει σε warps και τα προγραμματίζει προς εκτέλεση. Ο διαχωρισμός αυτός γίνεται σύμφωνα με τον αύξοντα αριθμό του κάθε νήματος και είναι πάντα ο ίδιος. Στη συνέχεια, η SIMT επιλέγει ένα warp που είναι ελεύθερο προς χρήση και του δίνει την εντολή εκτέλεσης. Και βέβαια όπως είναι κατανοητό, η εντολή αυτή είναι το ίδιο εφαρμόσιμη για όλα τα νήματα μέσα στο warp τα οποία θα εκτελεστούν ανεξάρτητα, ακόμα και αν χρησιμοποιούν κοινά κομμάτια του κώδικα όπως για παράδειγμα μια υπορουτίνα.

Πολύ σημαντικό θέμα για την μέγιστη απόδοση της συσκευής κατά την εκτέλεση μιας εφαρμογής είναι το πόσα μπλοκ μπορεί να επεξεργαστεί κάθε ομάδα μικροεπεξεργαστών. Αυτό εξαρτάται από το πόσοι registers και πόση από τη διαθέσιμη shared memory χρειάζεται για το κάθε kernel που στέλνεται στην συσκευή, (οι registers και shared memory είναι θέσεις μνήμης και θα περιγραφούν στη συνέχεια). Και αυτό γιατί η διαθέσιμη μνήμη, που έχει συγκεκριμένο μέγεθος για την κάθε συσκευή, θα διασπαστεί και θα χρησιμοποιηθεί ανάλογα στα νήματα των μπλοκ που θα σταλούν. Έτσι, θα πρέπει κατά τον προγραμματισμό της εφαρμογής, να είναι γνωστές οι απαιτήσεις του κάθε πυρήνα εκτέλεσης που θα στέλνεται στη συσκευή ώστε να επιτυγχάνεται η βέλτιστη λειτουργία του. Σε περίπτωση που δεν υπάρχουν

αρκετοί registers ή διαθέσιμη μνήμη για την εκτέλεση έστω ενός μπλοκ ο πυρήνας θα αποτύχει να εκτελεστεί. Αντίστοιχα μια ομάδα πολυεπεξεργαστών μπορεί να εκτελέσει ταυτόχρονα μέχρι 8 μπλοκ.

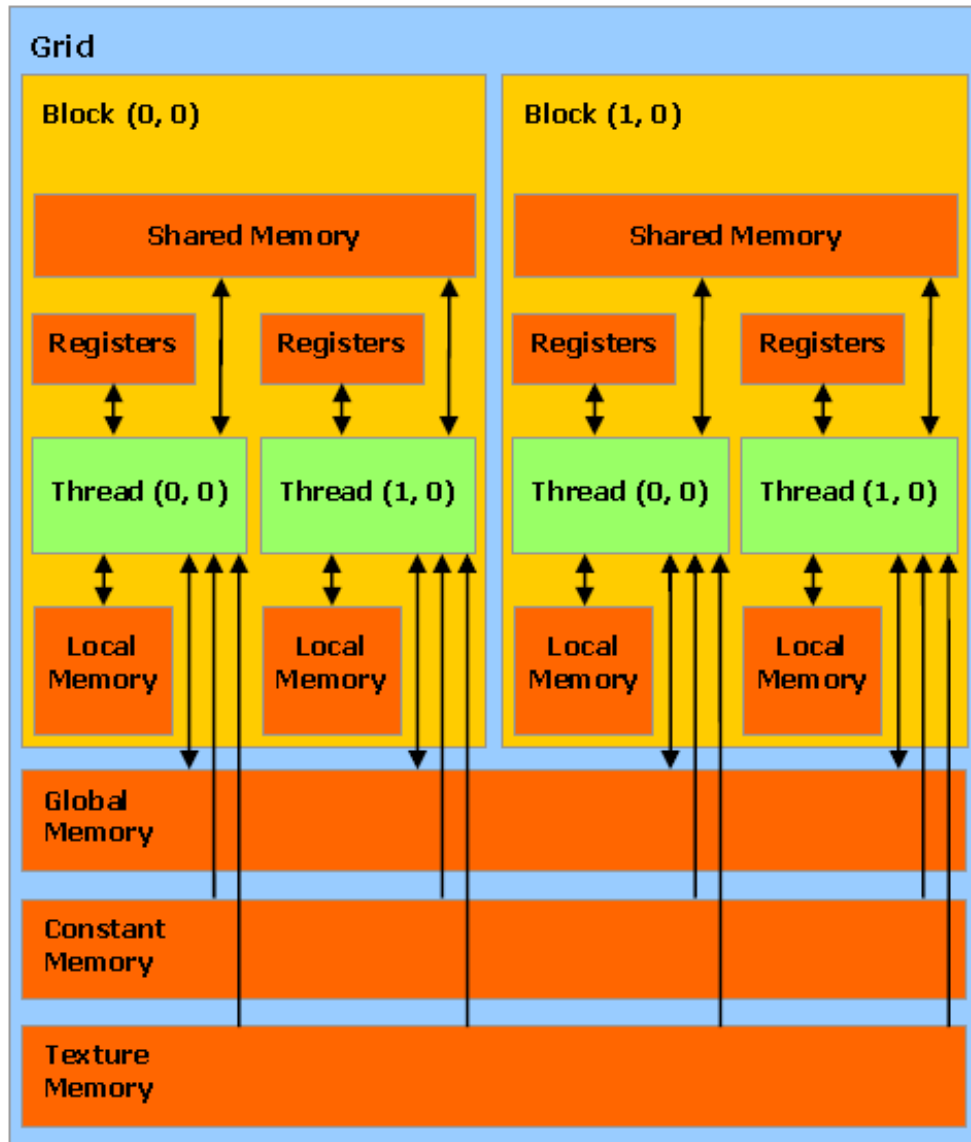
## A.5 Θέσεις Μνήμης της Κάρτας

Οι γενικές μνήμες (DRAM) της κάρτας γραφικών, έχουν σχεδιαστεί για πολύ μεγάλη ευελιξία, αντίστοιχη με αυτές των κεντρικών επεξεργαστών. Αυτό επιτυγχάνεται μέσω της δυνατότητας τους τόσο για ανάγνωση όσο και για εγγραφή πληροφοριών από τη συσκευή προς αυτές σε οποιαδήποτε θέση τους όπως φαίνεται στο παρακάτω Σχήμα 38.



Σχήμα 38: Λειτουργία DRAM [29]

Στο Σχήμα 39 που ακολουθεί διακρίνονται οι 6 διαθέσιμες μνήμες που βρίσκονται στην κάρτα γραφικών. Συγκεκριμένα υπάρχουν οι shared memory, local memory, global memory, constant, texture memory και οι registers.



Σχήμα 39: Το Μοντέλο της Μνήμης [29]

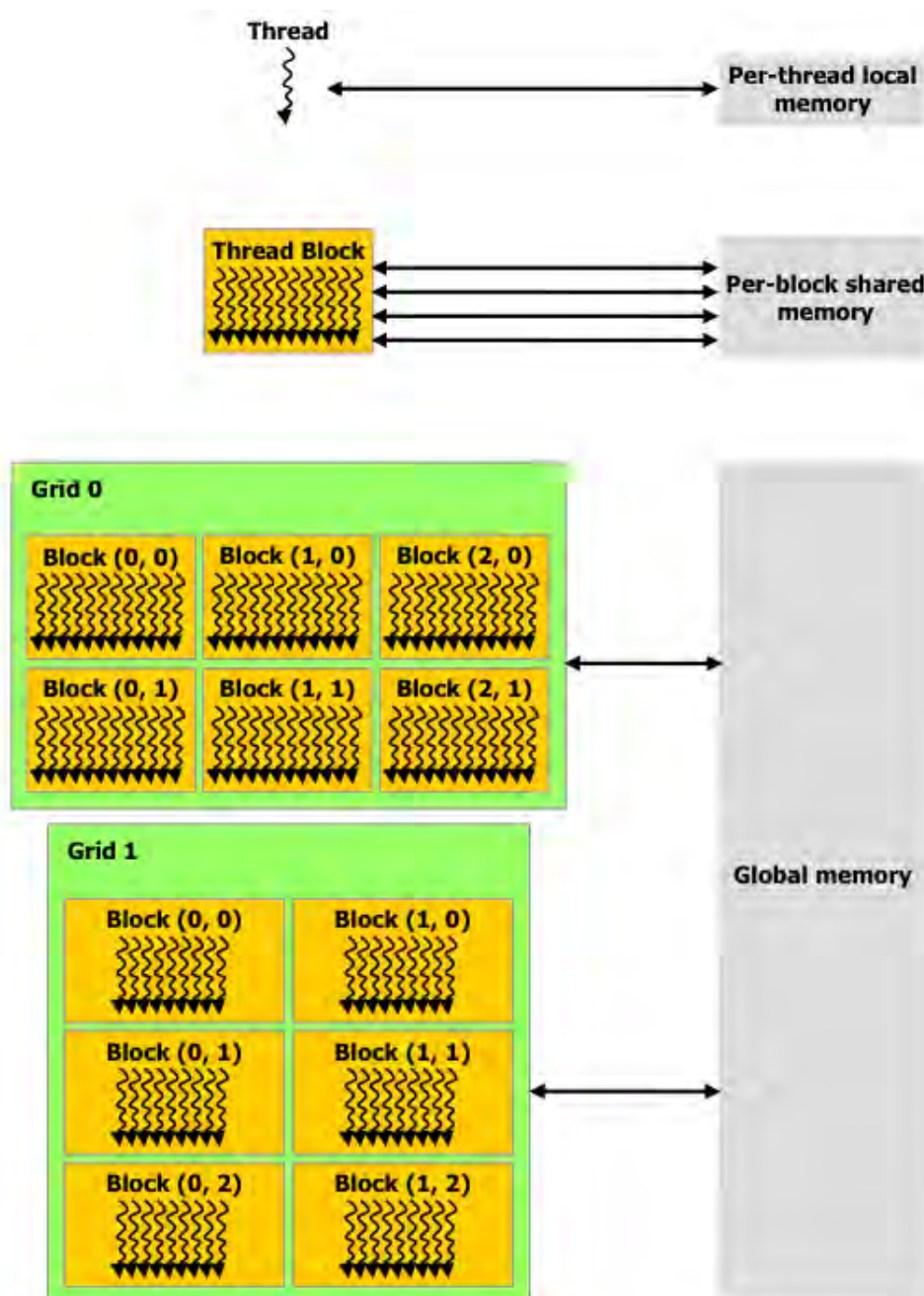
Είναι πολύ χρήσιμο για τον προγραμματισμό στη κάρτα γραφικών να γίνει κατανοητό τι μπορεί να έχει πρόσβαση σε κάθε μια από τις παραπάνω μνήμες.

Παρατηρώντας το παραπάνω σχήμα βλέπουμε:

- Στη Global memory (καθολική μνήμη) μπορεί να διαβάσει και να γράψει το κάθε πλέγμα.
- Στη Shared memory (κοινόχρηστη μνήμη) μπορεί να διαβάσει και να γράψει το κάθε μπλοκ.
- Στη Local Memory (τοπική μνήμη) μπορεί να διαβάσει και να γράψει το κάθε νήμα.

- Τους Registers (καταχωρητές) μπορεί να διαβάσει και να γράψει το κάθε νήμα.
- Στη Constant Memory (μνήμη σταθερών) μπορεί να διαβάσει μόνο το κάθε πλέγμα.
- Στη Texture Memory (μνήμη υφής) μπορεί να διαβάσει μόνο το κάθε πλέγμα.

Οι παραπάνω αλληλοεπιδράσεις αναπαριστώνται στο παρακάτω Σχήμα 40.



Σχήμα 40: Η Χρήση της κάθε Μνήμης [19]



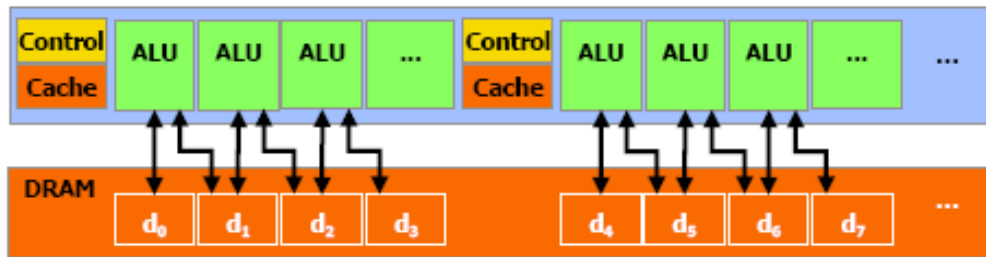
Σημαντικό χαρακτηριστικό είναι ότι στις global, constant και texture μνήμες μπορεί να γράψει και να διαβάσει και κεντρική μονάδα επεξεργασίας, αλλά με μεγάλο πρόσθετο υπολογιστικό κόστος. Επίσης πολύ σημαντική για τη βέλτιστη εκτέλεση ενός προγράμματος είναι η ταχύτητα ανάγνωσης της κάθε μνήμης δηλαδή το λεγόμενο memory bandwidth. Στη συνέχεια, ακολουθεί αναλυτική παρουσίαση για το πώς λειτουργεί η κάθε μνήμη.

### **A.5.1 Global Memory**

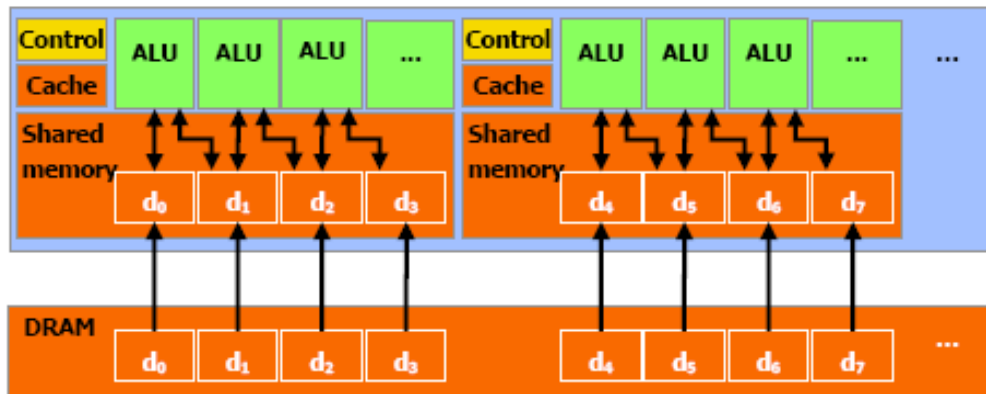
Η global μνήμη είναι η πιο βασική και είναι μέρος της μνήμης DRAM της συσκευής. Είναι η πιο μεγάλη μνήμη σε χωρητικότητα (φτάνει σε χωρητικότητα μερικά GBytes) αλλά και η πιο αργή, επομένως είναι πολύ σημαντικό να ακολουθούνται τα σωστά πρότυπα καταχώρησης για να επιτυγχάνεται μέγιστο εύρος μεταφοράς δεδομένων. Αρχικά, η συσκευή έχει την δυνατότητα να διαβάζει λέξεις μεγέθους 32-bit, 64-bit και 128-bit από την global μνήμη και να τις καταχωρεί στους registers με μια μονή εντολή.

### **A.5.2 Shared Memory**

Η Shared memory αποτελεί μνήμη η οποία βρίσκεται στα κυκλώματα του κάθε πολυεπεξεργαστή (on chip), για αυτό το λόγο είναι πολύ πιο γρήγορη από τη local και την global μνήμη αλλά περιορισμένη σε μέγεθος. Έχει συνολική χωρητικότητα μόλις 16 Kbyte σε κάθε πολυεπεξεργαστή (Στην αρχιτεκτονική Fermi μπορεί να φτάσει τα 48Kbyte) και έχει την ζωή του μπλοκ στο οποίο ορίζεται. Η λειτουργία της συσκευής με χρήση της μνήμης έχει ως αποτέλεσμα τη μείωση του απαιτούμενου εύρους μεταφοράς δεδομένων από τη μνήμη DRAM (Σχήμα 41).



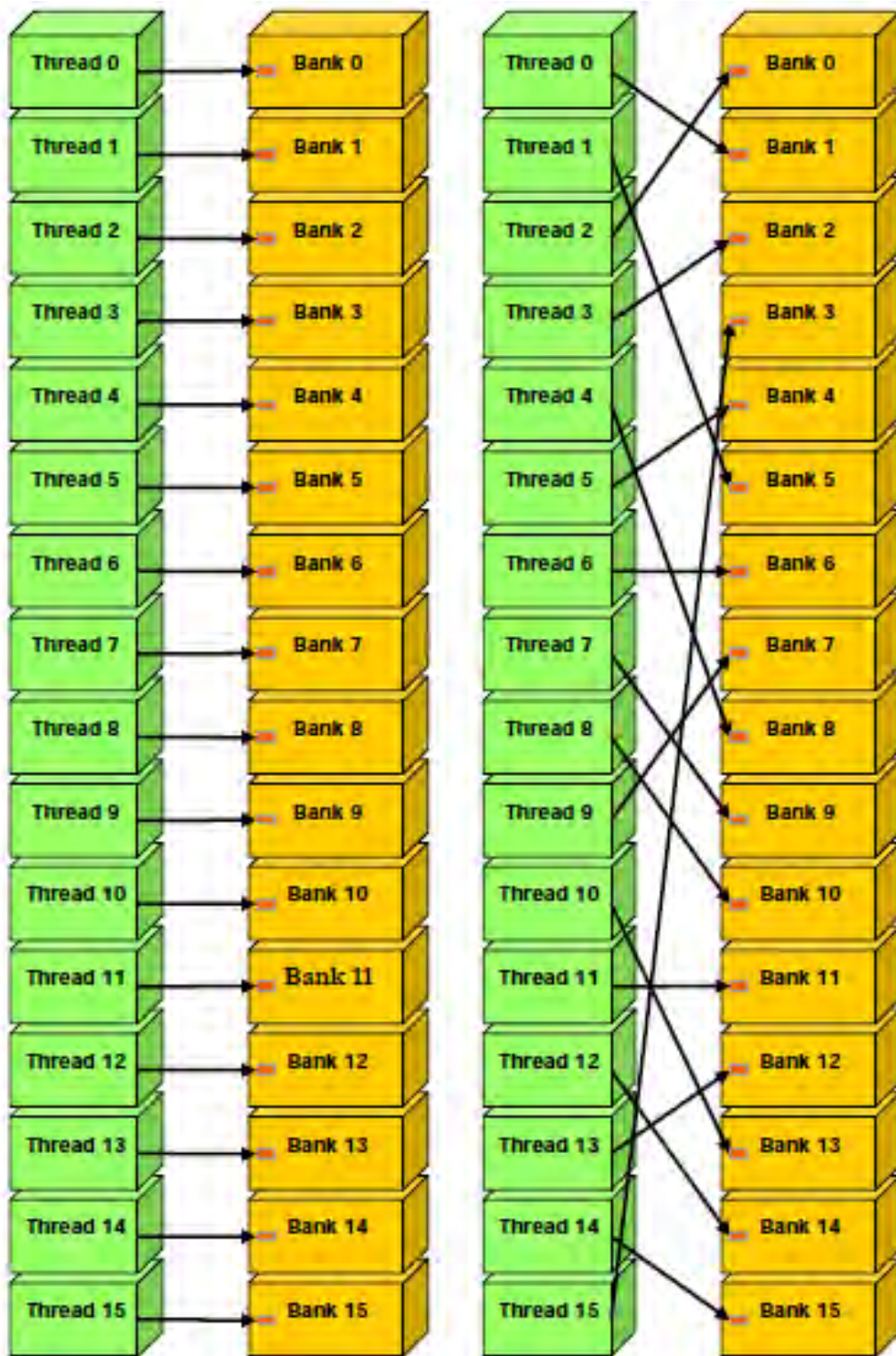
Without shared memory



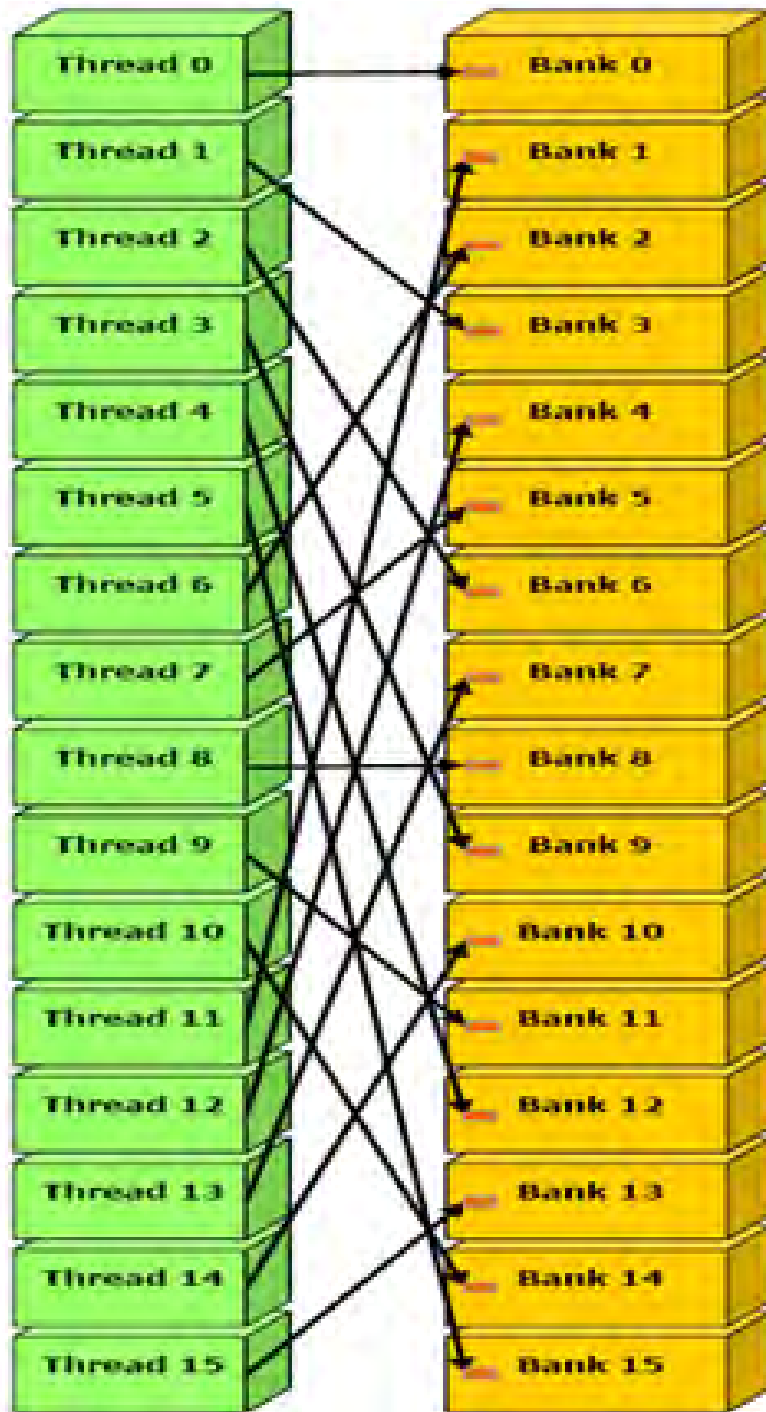
With shared memory

Σχήμα 41: Χρήση Κοινής Μνήμης [29]

Η shared μνήμη, έχει χωριστεί σε ίσου μεγέθους επιμέρους τμήματα τα οποία ονομάζονται **banks** (τράπεζες), σε καθένα από τα οποία μπορεί να υπάρχει πρόσβαση ταυτόχρονα. Στα σχήματα που ακολουθούν υπάρχουν 3 παραδείγματα τέτοιων καταχωρήσεων:

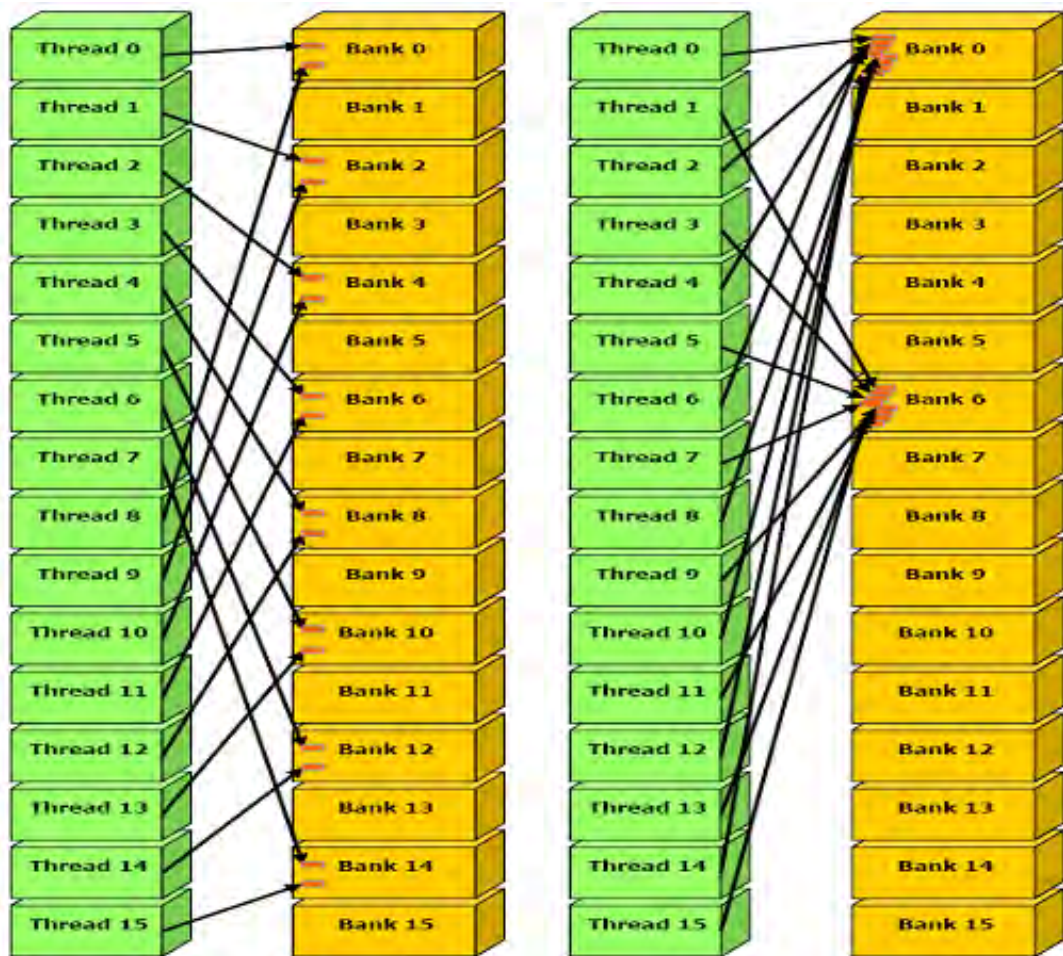


Σχήμα 42: Στα αριστερά έχουμε ομαλή γραμμική καταχώρηση μιας λέξης για κάθε ένα νήμα σε κάθε ένα bank, ενώ στα δεξιά έχουμε ομαλή τυχαία καταχώρηση μιας λέξης για κάθε ένα νήμα σε κάθε ένα bank [19]



Σχήμα 43: Ομαλή γραμμική καταχώρηση 3 λέξεων για κάθε βήμα σε θέσεις μνήμης [19]

Εάν δύο διευθύνσεις μνήμης πέσουν πάνω στο ίδιο bank μνήμης, δημιουργείται πρόβλημα, και για αυτό η πρόσβαση στον συγκεκριμένο bank πρέπει να γίνει σειριακά. Θα διαχωριστεί η αίτηση πρόσβασης στο συγκεκριμένο bank σε επιμέρους αιτήσεις πρόσβασης που δεν αντιμετωπίζουν πρόβλημα σύγκρουσης, μειώνοντας έτσι την ταχύτητα μεταφοράς δεδομένων. Στο σχήμα που ακολουθεί έχουμε δυο τέτοια παραδείγματα.

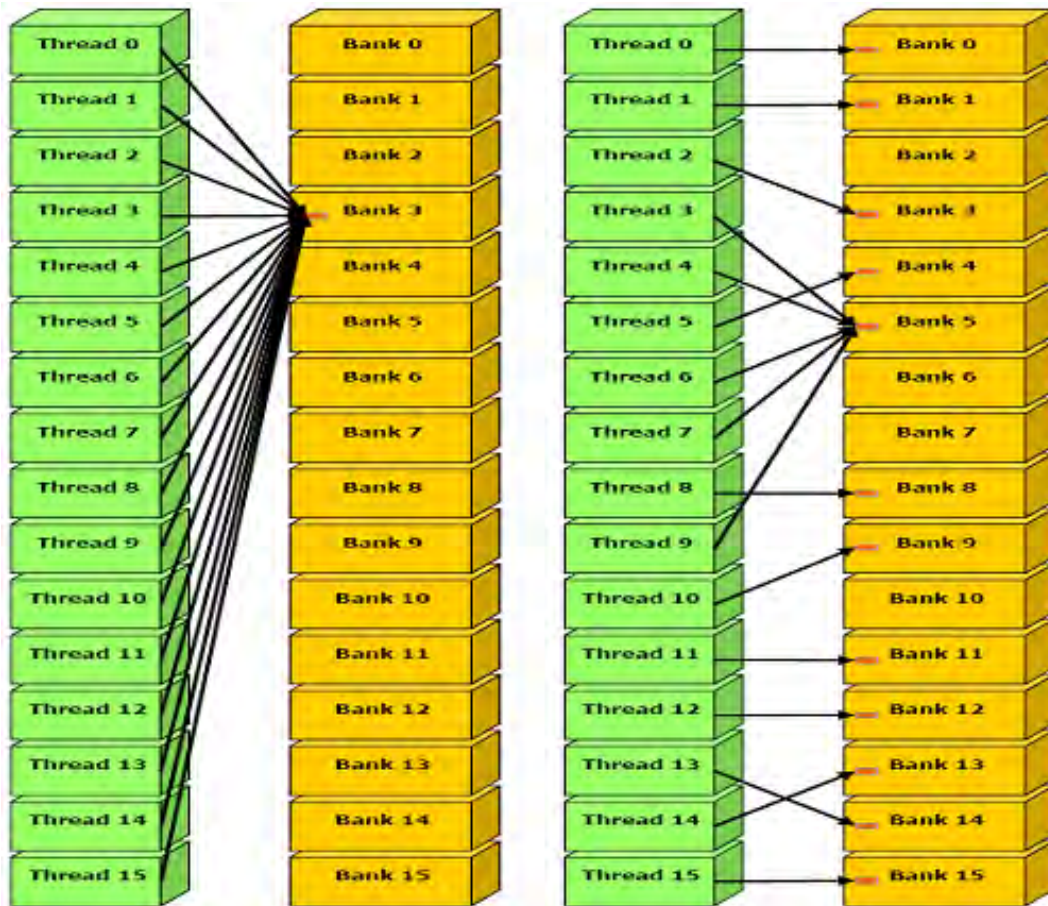


**Σχήμα 44: - Γραμμική καταχώρηση δύο λέξεων ανά νήμα που προκαλεί σύγκρουση διευθύνσεων 2<sup>η</sup> τάξης και γραμμική καταχώρηση 8 λέξεων ανά νήμα που προκαλεί σύγκρουση διευθύνσεων 8<sup>ης</sup> τάξης [19]**

Για την αποφυγή των συγκρούσεων κατά της πρόσβαση σε μνήμες να οι συσκευές με υπολογιστική δύναμη 1.0 όπου το μέγεθος του κάθε warp (ομάδα με παράλληλα νήματα) είναι 32 και ο αριθμός των banks είναι 16, μια εντολή από την κοινή μνήμη προς το warp διασπάται σε μια εντολή για το πρώτο μισό του warp και μια εντολή για το δεύτερο μισό του warp. Έτσι, ένα νήμα που ανήκει στο πρώτο μισό του warp δεν μπορεί να συγκρουστεί με ένα άλλο νήμα που ανήκει στο δεύτερο μισό του ίδιου warp.

Επίσης, η κοινή μνήμη διαθέτει ένα μηχανισμό, όπου αν μια λέξη των 32-bit απαιτείται να εκχωρηθεί σε μία κλήση μνήμης, μπορεί να διαβαστεί και να αναμεταδοθεί σε πολλά νήματα που την χρειάζονται ταυτόχρονα.

Στο σχήμα που ακολουθεί, υπάρχουν δύο παραδείγματα για αυτήν τη διαδικασία.



**Σχήμα 45:** Στο αριστερό σχήμα οι καταχωρήσεις της μνήμης γίνονται χωρίς σύγκρουση, μιας και όλα τα νήματα διαβάζουν από την ίδια θέση μνήμης με χρήση αναμετάδοσης, ενώ στο δεξί δεν θα προκληθεί σύγκρουση αν η λέξη στο bank 5 είναι αυτή που θα αναμεταδοθεί, αλλά σε αντίθετη περίπτωση θα υπάρξει σύγκρουση 2ης τάξης [19]

### A.5.3 Local Memory

Η local memory είναι η «προσωπική» μνήμη κάθε νήματος και βρίσκεται στην DRAM της κάρτας γραφικών. Έχει, επομένως, την ίδια ταχύτητα με την καθολική μνήμη. Πρακτικά αυτή η μνήμη χρησιμοποιείται πάρα πολύ σπάνια, γιατί δεν χρειάζεται σχεδόν ποτέ. Οι πολυεπεξεργαστές έχουν μία μορφή μνήμης, τους registers, η οποία παίζει τον ρόλο της local memory.

Ουσιαστικά η local memory είναι η τελευταία διέξοδος όταν οι απαιτήσεις σε μνήμη κάθε νήματος είναι πολύ μεγάλες. Δεν μπορούμε να ορίσουμε καν την τοπική μνήμη. Όποτε αφιερώνεται μνήμη σε κάθε νήμα, ο μεταγλωττιστής όταν ερμηνεύει την εντολή θα δεσμεύσει έναν ή περισσότερους registers. Η local memory θα δεσμευτεί εφόσον ξεπεραστεί το όριο των διαθέσιμων registers ή αν δεσμευτεί

μεγάλος πίνακας ή αν ο μεταγλωττιστής δεν μπορεί να καταλάβει αν ο πίνακας θα είναι σταθερός ή/και μικρός σε μέγεθος πίνακας στο νήμα.

#### **A.5.4 Registers**

Οι registers είναι μέρος των κυκλωμάτων κάθε πολυεπεξεργαστή και το μέγεθος τους εξαρτάται από την αρχιτεκτονική. Οι κάρτες γραφικών με αρχιτεκτονική 1.0 και 1.1 έχουν μόλις 8192 registers των 32bit, ενώ στις αρχιτεκτονικές 1.2 και 1.3 υπάρχουν 16834 registers των 32bit. Γενικά η πρόσβαση στους registers δεν επιβαρύνει τους κύκλους του ρολογιού ανά εντολή κλήσης, αλλά καθυστερήσεις μπορεί να προκύψουν για αναγνώσεις που ακολουθούν αμέσως μετά από εγγραφή αλλά και με συγκρούσεις όπως στην περίπτωση της shared μνήμης. Οι μεν καθυστερήσεις εγγραφής ανάγνωσης μπορούν να αγνοηθούν αν τα ενεργά νήματα για κάθε ομάδα μικροεπεξεργαστών είναι περισσότερα από 192, οι δε συγκρούσεις μπορούν να ελαχιστοποιηθούν όταν τα νήματα που υπάρχουν σε κάθε μπλοκ είναι πολλαπλάσια του 64.

#### **A.5.5 Constant Memory**

Η Constant Memory αποτελεί λανθάνουσα μνήμη (cached memory) γεγονός που την κάνει πολύ γρήγορη. Είναι βελτιστοποιημένη για broadcasting (εκπομπή) και το κόστος ανάγνωσης είναι ο ίδιος με το χρόνο που θα χρειάζονταν η ανάγνωση από έναν register, αρκεί βέβαια όλα τα νήματα να διαβάζουν στην ίδια διεύθυνση μνήμης. Από εκεί και πέρα ο χρόνος ανάγνωσης αυξάνει γραμμικά με την αύξηση του αριθμού των διευθύνσεων από τις οποίες διαβάζουν τα threads.

### A.5.6 Texture Memory

Η texture memory είναι λανθάνουσα μνήμη (cached memory). Το κύριο χαρακτηριστικό της είναι ότι είναι σχεδιασμένη βέλτιστα για δισδιάστατη χωρική κατανομή του πλέγματος υπολογισμού που σημαίνει ότι τα νήματα ενός warp που διαβάζουν από θέσεις της texture μνήμης, θα πετύχουν βέλτιστη απόδοση όταν αυτές οι θέσεις μνήμης έχουν διευθύνσεις που είναι κοντά η μία στην άλλη. Επίσης, δεν έχει περιορισμούς για το πώς πρέπει να γίνεται η πρόσβαση σε αυτή, για να έχει βέλτιστη απόδοση, όπως οι constant και global μνήμες.

Γενικότερα, με τη χρήση της συγκεκριμένης μνήμης πετυχαίνουμε μείωση της απαίτησης για μεγάλο συνολικό εύρος διαμεταγωγής πληροφοριών από και προς τις μνήμες. Τέλος, παρατηρείται ότι το εύρος συχνότητας μεταφοράς δεδομένων μεταξύ της συσκευής και της μνήμης της συσκευής είναι πολύ μεγαλύτερο μεταξύ του εύρους συχνότητας της συσκευής και του host. Έτσι για να καταφέρουμε να επιτύχουμε τη βέλτιστη απόδοση της συσκευής, πρέπει να προσπαθούμε να ελαχιστοποιήσουμε τη μεταφορά δεδομένων μεταξύ host και device, ακόμα και αν αυτό συνεπάγεται με εκτέλεση κάποιων kernels σε χαμηλού επιπέδου παραλληλία. Επίσης, η μεταφορά των δεδομένων στη συσκευή, εκτελείται ταχύτερα όταν τα δεδομένα αυτά μεταφέρονται όλα μαζί και όχι σε μικρές ομάδες.



# Παράρτημα Β

## Γλώσσα Προγραμματισμού CUDA

---

B.1 Εισαγωγή

B.2 Προέκταση της C

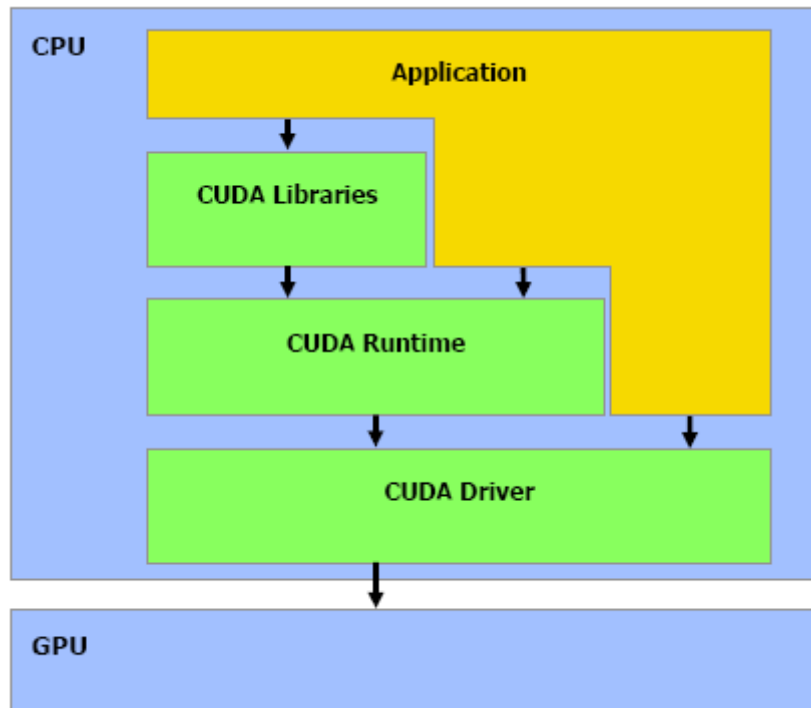
B.3 Προσθήκες Βιβλιοθήκης Εκτελέσιμων Εντολών

---

### **B.1 Εισαγωγή**

Το μοντέλο προγραμματισμού CUDA (Compute Unified Device Architecture) από την NVIDIA, έχει ως σκοπό να αντιμετωπίσει τους περιορισμούς που ήθελαν τη μονάδα επεξεργασίας γραφικών να προγραμματίζεται μόνο μέσω API γραφικών (όπου API είναι το μοντέλο επικοινωνίας το οποίο παρεμβάλλεται μεταξύ μιας εφαρμογής και του λειτουργικού συστήματος και καθορίζει τον τρόπο με τον οποίο η εφαρμογή αντλεί βιβλιοθήκες από το λειτουργικό). Με τη χρήση API γραφικών μειωνόταν η ταχύτητα εκτέλεσης των εφαρμογών και δεν μπορούσε να χρησιμοποιηθεί η μεγάλη υπολογιστική δύναμη της κάρτας για άλλες επεξεργασίες εκτός της επεξεργασίας γραφικών.

Το πακέτο λογισμικού της CUDA διακρίνεται σε επιμέρους διακριτά τμήματα όπως φαίνεται στο Σχήμα 46 που ακολουθεί, και είναι οι οδηγοί (CUDA drivers) της συσκευής, η διεπαφή εφαρμογής προγραμματισμού (API) μαζί με τα εκτελέσιμα της (runtime) και υψηλού επιπέδου βιβλιοθήκες κοινής χρήσης.



Σχήμα 46: Πακέτο Λογισμικού CUDA [19]

Παρακάτω, θα αναλυθεί η μέθοδος προγραμματισμού εφαρμογών με βάση τη γλώσσα προγραμματισμού CUDA, θα παρουσιαστούν οι νέες εντολές που εισάγονται και με τις οποίες επιτυγχάνεται η συνεργασία κεντρικού επεξεργαστή και κάρτας γραφικών.

## B.2 Προέκταση της C

Η γλώσσα προγραμματισμού CUDA, στηρίζεται στην υπάρχουσα γλώσσα προγραμματισμού C. Έτσι για κάποιον που είναι εξοικειωμένος στο να προγραμματίζει σε C με τη μάθηση κάποιων επιπλέον γνώσεων θα είναι σε θέση να γράφει και να εκτελεί προγράμματα και στη κάρτα γραφικών. Οι νέες εντολές που εισάγονται ως προέκταση της C, δίνουν τη δυνατότητα στο χρήστη να ορίσει τμήματα κώδικα του προγράμματος που θα εκτελεστούν στη κεντρική μονάδα επεξεργασίας (cpu) και στη μονάδα επεξεργασίας γραφικών (gpu).

Για λόγους συντομίας, στη συνέχεια, ο κεντρικός επεξεργαστής του υπολογιστή θα αναφέρεται με την αγγλική λέξη host ή την λέξη επεξεργαστής, καθώς

είναι η μητρική συσκευή από την οποία ξεκινάει η εκτέλεση του προγράμματος και η κάρτα γραφικών με την αγγλική λέξη `device` ή την ελληνική λέξη συσκευή.

Αυτές οι εντολές που αποτελούν προέκταση της C μπορούν να διακριθούν στις παρακάτω βασικές κατηγορίες:

- Στους προσδιοριστές των υπορουτινών που καθορίζουν από πού θα κληθεί μια υπορουτίνα και που θα εκτελεστεί, δηλαδή στον `host` (κεντρική μονάδα επεξεργασίας) ή στη `device` (μονάδα επεξεργασίας γραφικών).
- Στους προσδιοριστές του είδους των μεταβλητών, που καθορίζουν αν η θέση της μνήμης μιας μεταβλητής βρίσκεται στο `host` ή στο `device`.
- Στον τρόπο κλήσης μιας υπορουτίνας η οποία καλείται από το `host` και εκτελείται στο `device`.
- Σε μια κατηγορία τεσσάρων μεταβλητών που καθορίζουν τη διάσταση του πλέγματος και του κάθε μπλοκ και πληροφορούν σε ποίο νήμα, ποιού μπλοκ βρίσκεται η εκτέλεση του προγράμματος.

Ακολουθεί αναλυτική περιγραφή της κάθε κατηγορίας.

### B.2.1 Προσδιοριστές Υπορουτινών

Υπάρχουν τρεις προσδιοριστές που τοποθετούνται πριν από το όνομα της κάθε υπορουτίνας κατά τον ορισμό της και δείχνουν, από πού θα κληθεί η υπορουτίνα και που θα εκτελεστεί.

- Για να κληθεί μια υπορουτίνα από την `device` και να εκτελεστεί στην `device`, πρέπει να χρησιμοποιήσουμε το προσδιοριστικό `__device__`.
- Για να κληθεί μια υπορουτίνα από το `host` και να εκτελεστεί στην `device`, πρέπει να χρησιμοποιήσουμε το προσδιοριστικό `__global__`. Κάθε υπορουτίνα `__global__` καλείται με συγκεκριμένο τρόπο που περιγράφετε στο κεφάλαιο B.2.3, ενώ πρέπει να είναι τύπου `void` όσον αφορά τον τρόπο επιστροφής του αποτελέσματος της .
- Για να κληθεί μια υπορουτίνα από το `host` και να εκτελεστεί στο `host`, πρέπει να χρησιμοποιήσουμε το προσδιοριστικό `__host__`.

Στην περίπτωση όπου μια υπορουτίνα πρέπει να είναι εκτελέσιμη από την device και το host, χρησιμοποιούνται και τα δύο προσδιοριστικά μαζί, δηλαδή το προσδιοριστικό `__device__` και το `__host__`. Επίσης αξιοσημείωτο γεγονός είναι ότι δεν επιτρέπεται η αναδρομική κλήση υπορουτίνων που εκτελούνται στην device.

## B.2.2 Προσδιοριστές Μεταβλητών

Υπάρχουν τρεις προσδιοριστές που τοποθετούνται πριν το όνομα της μεταβλητής στο σημείο ορισμού της και υποδηλώνουν σε ποια μνήμη είναι αποθηκευμένη η μεταβλητή και ανάλογα, ποια είναι η διάρκεια ζωής της. Στο Σχήμα 37 που αναφερθήκαμε στο παράρτημα Α στο κεφάλαιο Α.5 υπάρχει μια σχηματική αναπαράσταση για τη θέση της κάθε μνήμης μέσα στην κάρτα γραφικών.

- Το προσδιοριστικό `__device__` καθορίζει μια μεταβλητή που είναι αποθηκευμένη στην μνήμη της device και συγκεκριμένα στην global memory, η διάρκεια ζωής της είναι ίση με την διάρκεια ζωής της εφαρμογής και είναι προσβάσιμη από οποιοδήποτε νήμα στην κάρτα αλλά και από τον host.
- Το προσδιοριστικό `__constant__` καθορίζει μια μεταβλητή που είναι αποθηκευμένη στη μνήμη της device και συγκεκριμένα στην constant memory, η διάρκεια ζωής της είναι ίση με τη διάρκεια ζωής της εφαρμογής και είναι προσβάσιμη από οποιοδήποτε νήμα στην device αλλά και από το host.
- Το προσδιοριστικό `__shared__` καθορίζει μια μεταβλητή που είναι αποθηκευμένη στη μνήμη της device και συγκεκριμένα στην shared memory ενός μπλοκ, η διάρκεια ζωής της είναι ίση με τη διάρκεια ζωής του μπλοκ και είναι προσβάσιμη μόνο από το ίδιο το μπλοκ.

Σημαντικό στοιχείο είναι ότι οι μεταβλητές `__device__` μπορούν να οριστούν και ως `__constant__` ή `__shared__` και έτσι να πάρουν τις αντίστοιχες ιδιότητες, ενώ αν δεν υπάρχει τέτοιος προσδιοριστής κατά τον ορισμό μιας μεταβλητής στην device τότε θα τοποθετηθεί αυτόματα σε κάποιον register ή στην local memory.

### B.2.3 Προσδιοριστικά Κλήσης Υπορουτίνας

Κάθε `__global__` υπορουτίνα, πρέπει να συνοδεύεται από ένα προσδιοριστικό το οποίο θα καθορίσει τις διαστάσεις του πλέγματος και των μπλοκ που θα χρησιμοποιηθούν κατά την εκτέλεση της συγκεκριμένης υπορουτίνας. Για παράδειγμα έστω η συνάρτηση της μορφής `__global__ void function(int* parameter);` το προσδιοριστικό κλήσης της θα είναι `function <<<Dg, Db, Ns>>> (parameter);`

Όπου:

- Το `Dg` δείχνει τις διαστάσεις του πλέγματος
- Το `Db` δείχνει τις διαστάσεις του μπλοκ
- Το `Ns` δείχνει τον αριθμό των bytes της shared memory που είναι δυναμικά δεσμευμένα για κάθε μπλοκ για τη συγκεκριμένη κλήση της συνάρτησης, γενικά αυτός ο αριθμός δεν είναι απαραίτητο να οριστεί.

### B.2.4 Μεταβλητές Θέσης και Διαστάσεων

Υπάρχουν τέσσερις μεταβλητές οι οποίες μπορούν να χρησιμοποιηθούν μέσα σε κάθε `__global__` υπορουτίνα προκειμένου να προσδιοριστεί η διάσταση του συνολικού πλέγματος αλλά και ποιο νήμα εκτελείται κάθε στιγμή.

Αυτές είναι:

- Η μεταβλητή **gridDim** η οποία όπως φανερώνει το όνομά της, δίνει την διάσταση του πλέγματος. Με το `gridDim.x` και `gridDim.y` να παίρνουν αντίστοιχα το πλάτος και το ύψος του πλέγματος.
- Η μεταβλητή **blockIdx** η οποία δίνει τον αύξοντα αριθμό του μπλοκ μέσα στο πλέγμα. Με το `blockIdx.x` παίρνουμε την x θέση και με την `blockIdx.y` την y.
- Η μεταβλητή **blockDim** δίνει τη διάσταση του μπλοκ, σε αντιστοιχία με τη **gridDim**.
- Η μεταβλητή **threadIdx** που δίνει τον αύξοντα αριθμό του νήματος μέσα στο μπλοκ.

### **B.3 Προσθήκες Βιβλιοθήκης Εκτελέσιμων Εντολών**

Η βιβλιοθήκη των εκτελέσιμων εντολών της CUDA χωρίζεται στα παρακάτω τμήματα:

- Το τμήμα του host περιλαμβάνονται εντολές που εκτελούνται στο host και χρησιμοποιούνται για τον έλεγχο μιας ή περισσότερες κάρτες γραφικών από το host.
- Το τμήμα της device που περιλαμβάνει εντολές που εκτελούνται και απευθύνονται μόνο στην device.
- Το τμήμα που είναι κοινό και για τον host και για την device.

#### **B.3.1 Βιβλιοθήκη Κοινών Εντολών**

Σε αυτό το τμήμα ανήκουν :

- Οι ορισμοί μεταβλητών διανυσματικού τύπου όπως int1, int2, float1, float2, long1, long2.
- Οι γνωστές μαθηματικές συναρτήσεις.
- Η συνάρτηση clock\_t clock(); Που χρησιμοποιείται για να τη μέτρηση του χρόνου που κάνει κάθε βήμα ξεχωριστά για να εκτελεστεί στην κάρτα.
- Τέλος, οι εντολές που μας επιτρέπουν να αποκτήσουμε πρόσβαση στην texture memory της κάρτας γραφικών.

#### **B.3.2 Βιβλιοθήκη Εντολών Συσκευής**

Σε αυτό το τμήμα ανήκουν:

- Οι γνωστές μαθηματικές συναρτήσεις οι οποίες όμως είναι υλοποιημένες με άλλο τρόπο ώστε να εκμεταλλεύονται την κάρτα γραφικών, και έχουν διαφορετικό όνομα.
- Η εντολή συγχρονισμού των νημάτων.

- Οι συναρτήσεις στρογγυλοποίησης των τιμών των μεταβλητών.
- Εντολές διαχείρισης της μνήμης texture.

### B.3.3 Βιβλιοθήκη Εντολών Συσκευής

Είναι τμήμα στο οποίο ανήκουν μερικές πολύ σημαντικές εντολές όπως:

- Εντολές διαχείρισης των διαθέσιμων συσκευών. Αυτές οι εντολές είναι χρήσιμες αν το υπολογιστικό σύστημα αποτελείται από περισσότερες από μια κάρτες γραφικών. Με τις κατάλληλες εντολές μπορεί να γίνει αρίθμηση αυτών των συσκευών και να οριστεί σε ποια θα γίνει η εκτέλεση του προγράμματος.
- Εντολές με τις οποίες γίνεται η διαχείριση της μνήμης μεταξύ του host και της device. Τέτοιες εντολές είναι οι **cudaMalloc()** και **cudaFree()** οι οποίες δεσμεύουν και απελευθερώνουν θέσεις μνήμης στην device από τον host, καθώς και οι εντολές **cudaMemcpy()** με τις οποίες γίνεται αντιγραφή των τιμών των μεταβλητών από τον host στην device αλλά και το αντίστροφο.
- Τέλος μια μεγάλη κατηγορία εντολών για τη διαχείριση των streams (ακολουθία ενεργειών που εκτελούνται σε σειρά) στην device.

# Παράρτημα Γ

## Υποστηριζόμενοι τύποι εικόνας από το σύστημα

Το σύστημα για την μετατροπή των εικόνων από ένα πλήθος τύπων σε jpg και την μετατροπή από jpg σε pbm χρησιμοποιεί το ελεύθερο πρόγραμμα ανοιχτού λογισμικού ImageMagick [31]. Ο παρακάτω πίνακας περιέχει του τύπους των εικόνων, τη λειτουργία και μια σύντομη περιγραφή για κάθε τύπο που υποστηρίζει το ImageMagick και ως συνέπεια το σύστημα που αναπτύχθηκε σε αυτή την εργασία.

Πίνακας 9: Υποστηριζόμενοι τύποι εικόνας

Tag	Mode	Description
AAI	RW	AAI Dune image
ART	R	PFS: 1st Publisher
ARW	R	Sony Digital Camera Alpha Raw Image Format
<a href="#">AVI</a>	R	Microsoft Audio/Visual Interleaved
AVS	RW	AVS X image
<a href="#">BMP, BMP2, BMP3</a>	RW	Microsoft Windows bitmap
<a href="#">CAL5</a>	R	Continuous Acquisition and Life-cycle Support Type 1 image
<a href="#">CGM</a>	R	Computer Graphics Metafile
<a href="#">CIN</a>	RW	Kodak Cineon Image Format
CMYK	RW	Raw cyan, magenta, yellow, and black samples
CMYKA	RW	Raw cyan, magenta, yellow, black, and alpha samples
CR2	R	Canon Digital Camera Raw Image Format
CRW	R	Canon Digital Camera Raw Image Format
CUR	R	Microsoft Cursor Icon



CUT	R	DR Halo
DCM	R	Digital Imaging and Communications in Medicine (DICOM) image
DCR	R	Kodak Digital Camera Raw Image File
DCX	RW	ZSoft IBM PC multi-page Paintbrush image
DIB	RW	Microsoft Windows Device Independent Bitmap
<a href="#">DJVU</a>	R	
<a href="#">DNG</a>	R	Digital Negative
<a href="#">DOT</a>	R	Graph Visualization
<a href="#">DPX</a>	RW	SMPTE Digital Moving Picture Exchange 2.0 (SMPTE 268M-2003)
EMF	R	Microsoft Enhanced Metafile (32-bit)
EPDF	RW	Encapsulated Portable Document Format
EPI	RW	Adobe Encapsulated PostScript Interchange format
EPS	RW	Adobe Encapsulated PostScript
EPS2	W	Adobe Level II Encapsulated PostScript
EPS3	W	Adobe Level III Encapsulated PostScript
EPSF	RW	Adobe Encapsulated PostScript
EPSI	RW	Adobe Encapsulated PostScript Interchange format
EPT	RW	<a href="#">Adobe Encapsulated PostScript Interchange format with TIFFpreview</a>
<a href="#">EXR</a>	RW	High dynamic-range (HDR) file format developed by Industrial Light & Magic
FAX	RW	Group 3 TIFF
<a href="#">FIG</a>	R	FIG graphics format
<a href="#">FITS</a>	RW	Flexible Image Transport System
FPX	RW	FlashPix Format
<a href="#">GIF</a>	RW	CompuServe Graphics Interchange Format
GPLT	R	Gnuplot plot files
GRAY	RW	Raw gray samples
<a href="#">HDR</a>	RW	Radiance RGBE image format
HPGL	R	HP-GL plotter language

HRZ	RW	Slow Scane TeleVision
HTML	RW	Hypertext Markup Language with a client-side image map
ICO	R	Microsoft icon
INFO	W	Format and characteristics of the image
INLINE	R	Base64-encoded inline image
JBIG	RW	Joint Bi-level Image experts Group file interchange format
<a href="#">JNG</a>	RW	Multiple-image Network Graphics
<a href="#">JP2</a>	RW	JPEG-2000 JP2 File Format Syntax
<a href="#">JPC</a>	RW	JPEG-2000 Code Stream Syntax
<a href="#">JPEG</a>	RW	Joint Photographic Experts Group JFIF format
MAN	R	Unix reference manual pages
MAT	R	MATLAB image format
<a href="#">MIFF</a>	RW	Magick image file format
MONO	RW	Bi-level bitmap in least-significant-byte first order
<a href="#">MNG</a>	RW	Multiple-image Network Graphics
<a href="#">M2V</a>	RW	Motion Picture Experts Group file interchange format (version 2)
<a href="#">MPEG</a>	RW	Motion Picture Experts Group file interchange format (version 1)
MPC	RW	Magick Persistent Cache image file format
R	RW	Magick Persistent Registry
MRW	R	Sony (Minolta) Raw Image File
MSL	RW	Magick Scripting Language
<a href="#">MTV</a>	RW	MTV Raytracing image format
<a href="#">MVG</a>	RW	Magick Vector Graphics.

NEF	R	Nikon Digital SLR Camera Raw Image File
ORF	R	Olympus Digital Camera Raw Image File
OTB	RW	On-the-air Bitmap
P7	RW	Xv's Visual Schnauzer thumbnail format
PALM	RW	Palm pixmap
<a href="#">PAM</a>	W	Common 2-dimensional bitmap format
CLIPBOARD	RW	Windows Clipboard
<a href="#">PBM</a>	RW	Portable bitmap format (black and white)
PCD	RW	Photo CD
PCDS	RW	Photo CD
PCL	W	HP Page Control Language
<a href="#">PCX</a>	RW	ZSoft IBM PC Paintbrush file
PDB	RW	Palm Database ImageViewer Format
PDF	RW	Portable Document Format
PEF	R	Pentax Electronic File
PFA	R	Postscript Type 1 font (ASCII)
PFB	R	Postscript Type 1 font (binary)
<a href="#">PFM</a>	RW	Portable float map format
<a href="#">PGM</a>	RW	Portable graymap format (gray scale)
PICON	RW	Personal Icon
PICT	RW	Apple Macintosh QuickDraw/PICT file
PIX	R	Alias/Wavefront RLE image format
<a href="#">PNG</a>	RW	Portable Network Graphics
<a href="#">PNG8</a>	RW	Portable Network Graphics
<a href="#">PNG24</a>	RW	Portable Network Graphics
<a href="#">PNG32</a>	RW	Portable Network Graphics
<a href="#">PNM</a>	RW	Portable anymap
<a href="#">PPM</a>	RW	Portable pixmap format (color)
PS	RW	Adobe PostScript file
PS2	RW	Adobe Level II PostScript file
PS3	RW	Adobe Level III PostScript file
<a href="#">PSB</a>	RW	Adobe Large Document Format
<a href="#">PSD</a>	RW	Adobe Photoshop bitmap file

PTIF	RW	<a href="#">Pyramid encoded TIFF</a>
<a href="#">PWP</a>	R	Seattle File Works multi-image file
RAD	R	Radiance image file
RAF	R	Fuji CCD-RAW Graphic File
RGB	RW	Raw red, green, and blue samples
RGBA	RW	Raw red, green, blue, and alpha samples
RLA	R	Alias/Wavefront image file
RLE	R	Utah Run length encoded image file
<a href="#">SCT</a>	R	Scitex Continuous Tone Picture
<a href="#">SFW</a>	R	Seattle File Works image
SGI	RW	Irix RGB image
SHTML	W	Hypertext Markup Language client-side image map
SID, MrSID	R	Multiresolution seamless image
SUN	RW	SUN Rasterfile
<a href="#">SVG</a>	RW	Scalable Vector Graphics
TGA	RW	Truevision Targa image
<a href="#">TIFF</a>	RW	Tagged Image File Format
TIM	R	PSX TIM file
<a href="#">TTF</a>	R	TrueType font file
TXT	RW	Raw text file
UIL	W	X-Motif UIL table
UYVY	RW	Interleaved YUV raw image
VICAR	RW	VICAR rasterfile format
<a href="#">VIFF</a>	RW	Khoros Visualization Image File Format
<a href="#">WBMP</a>	RW	Wireless bitmap
<a href="#">WEBP</a>	RW	Weppy image format
<a href="#">WMF</a>	R	Windows Metafile
<a href="#">WPG</a>	R	Word Perfect Graphics File
X	RW	display or import an image to or from an X11 server
<a href="#">XBM</a>	RW	X Windows system bitmap, black and white only
XCF	R	GIMP image
<a href="#">XPM</a>	RW	X Windows system pixmap
<a href="#">XWD</a>	RW	X Windows system window dump
X3F	R	Sigma Camera RAW Picture File
YCbCr	RW	Raw Y, Cb, and Cr samples
YCbCrA	RW	Raw Y, Cb, Cr, and alpha samples
YUV	RW	CCIR 601 4:1:1

# Αναφορές

- [1] Κοσμόπουλος Δ., *Διαλέξεις στο Μάθημα "Ανάλυση Συστημάτων"*. 2009. Πανεπιστήμιο Στερεάς Ελλάδος, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική.
- [2] Αναγνωστόπουλος, Χ. *Διαλέξεις Μαθήματος "Επικοινωνία Ανθρώπου Υπολογιστή"*. in *Ανάλυση Απαιτήσεων και Καθορισμός Προδιαγραφών, Χρηστικότητα*. 2009. Πανεπιστήμιο Στερεάς Ελλάδος, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική.
- [3] Alex Berson, "*Client/server architecture*", 2η έκδοση, εκδόσεις *McGraw-Hill*, USA, 1996
- [4] Yadav, Subhash Chandra, "*An introduction to client server computing*", εκδόσεις *New Age International (P) Ltd*, 2009
- [5] *Σημειώσεις στο Εργαστήριο Παράλληλης Κατανεμημένης Επεξεργασίας*, Πανεπιστήμιο Μακεδονίας, Τμήμα Εφαρμοσμένης Πληροφορικής
- [6] Ηλεκτρονική ιστοσελίδα του SAP, <http://www1.sap.com/index.epx>
- [7] David B. Kirk, Wen-mei W. Hwu, "*Programming Massively Parallel Processors: A Hands-on Approach*", εκδόσεις Morgan Kaufmann, 2010
- [8] Σακκόπουλος, Ε., *Διαλέξεις Μαθήματος "Τεχνολογίες Εφαρμογών Διαδικτύου"*. 2009, Πανεπιστήμιο Στερεάς Ελλάδος, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική.
- [9] Ηλεκτρονική Ιστοσελίδα προτύπου ιστοσελίδας <http://www.templatemonster.com/free-mobile-templates.html>
- [10] Ηλεκτρονική Ιστοσελίδα προτύπου ιστοσελίδας [http://themeforest.net/item/my-mobile-page-v3-csshtml/1048037?WT.ac=search\\_item&WT.seg\\_1=search\\_item&WT.z\\_autho r=FamousThemes](http://themeforest.net/item/my-mobile-page-v3-csshtml/1048037?WT.ac=search_item&WT.seg_1=search_item&WT.z_autho r=FamousThemes)

- [11] Ηλεκτρονική ιστοσελίδα microsoft [http://msdn.microsoft.com/en-us/library/aa767735\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa767735(v=vs.85).aspx)
- [12] A. Kakarountas, I. Mavridis, "Efficient Exploitation of Parallel Computing on the Server-Side of Health Organizations' Intranet for Distributing Medical Images to Smart Devices", mobihealth, 2011
- [13] Ηλεκτρονική ιστοσελίδα βιβλιοθήκης NPP <http://developer.nvidia.com/npp>
- [14] Μαγκλογιάννης Η., *Διαλέξεις Μαθήματος Ανάλυση Βιοϊατρικών Εικόνων*, 2011. Πανεπιστήμιο Στερεάς Ελλάδος, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική.
- [15] Ηλεκτρονική ιστοσελίδα computer vision <http://www.csie.ntu.edu.tw/~r93944019/cv/>
- [16] Ηλεκτρονική ιστοσελίδα phpmyadmin <http://www.phpmyadmin.net>
- [17] Ηλεκτρονική ιστοσελίδα Nvidia geforce <http://www.geforce.com/hardware/desktop-gpus/geforce-9500-gt/specifications>
- [18] Ηλεκτρονική ιστοσελίδα intel [http://ark.intel.com/products/27518/Intel-Pentium-D-Processor-930-\(4M-Cache-3\\_00-GHz-800-MHz-FSB\)](http://ark.intel.com/products/27518/Intel-Pentium-D-Processor-930-(4M-Cache-3_00-GHz-800-MHz-FSB))
- [19] CUDA Programming Guide 2.0
- [20] Αναγνωστόπουλος Χ., *Διαλέξεις Μαθήματος "Επικοινωνία Ανθρώπου Υπολογιστή"*, 2009. Πανεπιστήμιο Στερεάς Ελλάδος, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική.
- [21] Ηλεκτρονική ιστοσελίδα της php, περιέχει τη λειτουργία των sessions <http://gr2.php.net/manual/en/book.session.php>
- [22] Oleg S. Pianykh, *"Digital Imaging and Communications in Medicine (DICOM) A Practical Introduction and Survival Guide"*, Springer, 2008
- [23] Ηλεκτρονική ιστοσελίδα thecryptmag [http://thecryptmag.com/Online/56/imgproc\\_5.html](http://thecryptmag.com/Online/56/imgproc_5.html)

- [24] Παπαμάρκος Η. Νικόλαος, Ψηφιακή Επεξεργασία & Ανάλυση Εικόνας, Αυτοέκδοση, 2010
- [25] Ηλεκτρονική ιστοσελίδα πηγή εικόνων  
<http://en.kioskea.net/contents/cs/cs3tier.php3>
- [26] Ηλεκτρονική ιστοσελίδα <http://theinf2.informatik.uni-jena.de/Lectures/Programming+with+CUDA/SS+2011.html>
- [27] Ηλεκτρονική ιστοσελίδα πηγή εικόνων <http://www.bdlot.com/resource/what-is-cuda.htm>
- [28] Ηλεκτρονική ιστοσελίδα πηγή εικόνων  
<http://www.evermotion.org/tutorials/show/7961/octane-render-review>
- [29] Ηλεκτρονική ιστοσελίδα πηγή εικόνων  
<http://www.beyond3d.com/content/articles/12/>
- [30] Mishra A, Lu Y, Meng J, Anderson AW, Ding Z, "Unified framework for anisotropic interpolation and smoothing of diffusion tensor images"
- [31] Ηλεκτρονική ιστοσελίδα ImageMagick <http://www.imagemagick.org>