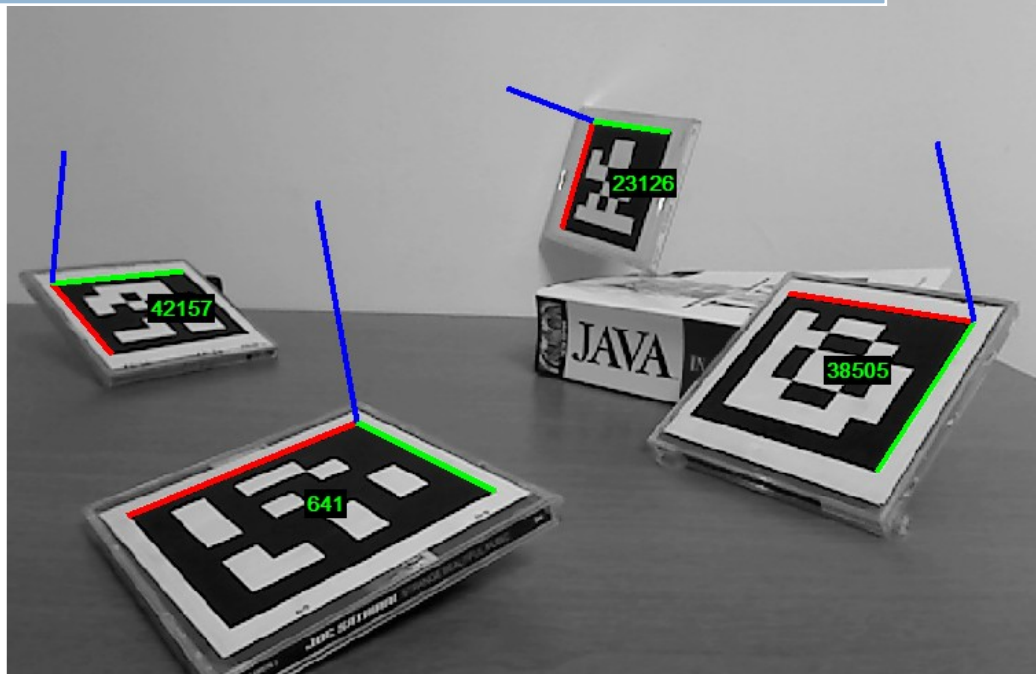


Ευέλικτοι αλγόριθμοι ανίχνευσης markers σε εικόνα



Εφαρμογή στην Επαυξημένη Πραγματικότητα

Τζιώνας Δημήτριος

Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, ΑΠΘ

9/11/2009

*“The most profound technologies
are those that disappear”*

Mark Weiser

*“Everything should be made as simple
as possible, but not one bit simpler”*

Albert Einstein

Ευχαριστίες

Σαν εισαγωγικό σημείωμα, θα ήθελα να ευχαριστήσω τον κ. Χατζηλεοντιάδη για την εμπιστοσύνη που μου υπέδειξε με την ανάθεση της παρούσας διπλωματικής εργασίας, για την ελευθερία κινήσεων που μου δόθηκε, ανάλογα με τα ενδιαφέροντά μου, καθώς και για τις (πολύωρες μερικές φορές) συζητήσεις που αποτέλεσαν έμπνευση.

Με την εργασία αυτή κλείνει ένας κύκλος πολυετών προσπαθειών που κάθε άλλο παρά ατομικές ήταν.

Ένα ευχαριστώ στους γονείς μου, Κώστα και Μαρία, καθώς και στην αδερφή μου, Γλύκα, δε θα ήταν σε καμία περίπτωση αρκετό. Είναι οι άνθρωποι που στήριξαν ανιδιοτελώς κάθε μου επιθυμία και προσπάθεια και αποτέλεσαν την κινητήρια δύναμη μετατροπής των ονείρων σε πραγματικότητα.

Κλείνοντας, δε θα μπορούσα φυσικά να μην αναφέρω τον Όμηρο και τον Αλέξη, που ήταν όλα αυτά τα χρόνια δίπλα, πολύτιμο στήριγμα και κυριολεκτική προσωποποίηση της έννοιας “φίλος”.

Τζιώνας Δημήτρης

Περίληψη

Ο βασικότερος τρόπος που αλληλεπιδρούμε με τους υπολογιστές, το GUI (Graphical User Interface) προέκυψε πριν από 28 χρόνια και έκτοτε έχει κυριαρχήσει παραμένοντας απaráλλακτο, όσον αφορά την κεντρική φιλοσοφία του: Ο χρήστης αποκόπτεται κατά κάποιο τρόπο από το περιβάλλον του, τα χέρια του δεσμεύονται στο πληκτρολόγιο και το ποντίκι, ενώ η προσοχή του περιορίζεται σε μία οθόνη.

Η “προβληματική” αυτή επικοινωνία ανθρώπου-υπολογιστή θα μπορούσε να αναβαθμιστεί μέσω ενός είδους User Interface με διαφορετική φιλοσοφία. Θα μπορούσε να αποτελεί ένα “συμπλήρωμα” προς τις αισθήσεις του ανθρώπου, δίνοντάς του σε πραγματικό χρόνο πληροφορίες που δε θα μπορούσε να λάβει αλλιώς με τις αισθήσεις του, ενώ οι πληροφορίες θα μπορούσαν να παρουσιάζονται ακριβώς εκεί όπου ανήκουν: στον πραγματικό κόσμο. Αυτή η μίξη του πραγματικού με τον εικονικό κόσμο, με τρόπο όσο το δυνατόν “φυσικότερο”, ονομάζεται Επαυξημένη Πραγματικότητα (Augmented Reality).

Ανακύπτει όμως το πρόβλημα, πώς ο υπολογιστής εντοπίζει το ενδιαφέρον του χρήστη και πού πρέπει να παρουσιάσει την επιθυμητή πληροφορία. Αυτό το πρόβλημα προσπαθεί να επιλύσει η παρούσα εργασία, με τον ακόλουθο τρόπο:

Εισάγονται στο περιβάλλον ορισμένα ασπρόμαυρα markers εν είδει δισδιάστατου barcode. Αυτά δημιουργούνται με έναν απλό εκτυπωτή, ενώ κάθε marker έχει μία μοναδική ταυτότητα. Προσκολλώντας ένα marker σε κάποιο αντικείμενο, αυτό αποκτάει την ταυτότητα του marker, το οποίο λειτουργεί σαν “οπτικό εικονίδιο”: δίνεται η δυνατότητα αναζήτησης πληροφοριών σχετικά με αντικείμενα/χώρους του πραγματικού κόσμου απλά με τη χρήση ενός υπολογιστή με κάμερα (ή camera phone) και του κατάλληλου λογισμικού. Ο χρήστης αναζητάει πληροφορίες με μία “point ‘n shoot” λογική, καθώς εξερευνά τον περιβάλλοντα χώρο, ενώ οι όποιες πληροφορίες επιστρέφονται, προβάλλονται “πάνω” στα αντικείμενα του ενδιαφέροντος.

Η παρούσα εργασία λειτουργεί προς αυτή την κατεύθυνση με την ανάπτυξη ενός εργαλείου δημιουργίας marker και ενός ευέλικτου αλγορίθμου “ανάγνωσης” του χώρου και εντοπισμού/ταυτοποίησης των markers που βρίσκονται σε αυτό. Παράλληλα εξασφαλίζονται τα απαραίτητα “εργαλεία” για πρόσβαση σε οποιαδήποτε εικονική πληροφορία συνδέεται με τα markers ή για προβολή 3D (εικονικών) γραφικών αντικειμένων πάνω σε αυτά.

Abstract

The basic way of human-computer interaction (HCI), the graphical user interface (GUI), has emerged 28 years ago and its main idea remains since then unchanged. The user is isolated from the “real world”: his hands are attached to the mouse and the keyboard, while his eyes are glued to a large screen.

The domain of human-computer interaction needs thus a completely new approach. This new approach could supplement the user’s feelings and point of view, presenting him in real time the needed information right where it belongs: attached to the real world. This way of combining the real and the virtual world is called Augmented Reality (AR).

The basic question that emerges is how the computer receives user input in a novel, subtle way and where should the information be received.

This thesis comes up with the following solution:

Special 2D barcodes, called visual markers are created with a simple printer. Every marker, which contains a unique ID, is attached to an object of the real world, playing thus the role of a phycon (physical icon). This implementation of “the internet of things” enables the user to receive digital information about his surroundings with a “point ‘n shoot” approach, using just a computer with a camera (or ideally a camera phone).

The following tools have been implemented towards the above mentioned approach:

- An application for marker creation
- An adaptive algorithm able to scan the environment, efficiently detect the markers attached to it and decode their unique ID. Moreover, it provides the right “tools” to access every kind of digital information relevant to each marker or to render 3D models that appear attached to them.

Πρόλογος

Τις τελευταίες δεκαετίες γινόμαστε μάρτυρες της συνεχούς και αλματώδους ανάπτυξης στην επιστήμη των υπολογιστών, ανάπτυξη η οποία συνεχίζεται και σήμερα με τους ίδιους εντυπωσιακούς ρυθμούς. Κάνοντας μία σύγκριση ανάμεσα στο παρόν και στο παρελθόν γίνεται εμφανής ο τεράστιος βαθμός στον οποίο έχει αλλάξει η ζωή μας με τη διείσδυση των υπολογιστών στην καθημερινότητά μας.

Οι υπολογιστές γίνονται συνεχώς γρηγορότεροι, μικρότεροι, λιγότερο ενεργοβόροι και υιοθετούνται με ταχύτατους ρυθμούς από ολοένα και περισσότερους ανθρώπους. Ενώ παλιότερα αποτελούσαν αγαθό μίας προνομιούχου επιστημονικής ελίτ, σήμερα έχουν διεισδύσει σε όλα τα κοινωνικά και οικονομικά στρώματα.

Ειδικά τα τελευταία χρόνια έχει αρχίσει να αποκτά κυριολεκτικό νόημα ο όρος “προσωπικός υπολογιστής”, που αρχικά είχε ταυτιστεί με την έννοια του desktop. Με την έλευση των laptops, των netbooks, και των smartphones έχουμε φτάσει στο σημείο κάθε άνθρωπος να έχει στη διάθεση του περισσότερους από έναν υπολογιστές, ενώ ειδικά το smartphone είναι ένας μικρός υπολογιστής που μας συνοδεύει κάθε χρονική στιγμή, σε όλες τις πτυχές της ζωής μας.

Την ταχύτητα ανάπτυξης του υλικού των υπολογιστών ακολουθεί όπως είναι φυσικό και το λογισμικό, με τις αυξανόμενες δυνατότητες που αποκτά. Η χρήση του κατάλληλου λογισμικού έχει γίνει προϋπόθεση σχεδόν για κάθε δραστηριότητα.

Παρατηρείται όμως το εξής παράδοξο: ενώ οι υπολογιστές απευθύνονται στο κοινό κάθε επιπέδου μόρφωσης, οι αυξανόμενες δυνατότητες του software το καθιστούν ολοένα και πολυπλοκότερο, ολοένα και πιο δύσχρηστο. Ο κόσμος των υπολογιστών απαιτεί από τους χρήστες να προσαρμοστούν σε αυτόν, να εξοικειωθούν με ειδική ορολογία και μεθόδους. Έτσι πολλές φορές οι χρήστες νιώθουν αποστροφή προς αυτόν, ενώ είναι απρόθυμοι να χρησιμοποιήσουν καινούριες, πλουσιότερες εκδόσεις λογισμικού, λόγω του πολυπλοκότερου interface του και του νέου φόρτου επιπλέον επιμόρφωσης.

Φυσικά δεν έχουν άδικο για κάτι τέτοιο. Το κυριότερο είδος διεπαφής υπολογιστή-χρήστη παραμένουν τα παράθυρα και τα μενού, στα οποία “κρύβονται βαθύτερα” οι νέες δυνατότητες του λογισμικού, τις οποίες ο χρήστης καλείται να ανακαλύψει.

Αυτό το είδος γραφικής διεπαφής ανθρώπου-μηχανής (GUI) έχει τις ρίζες του στο Xerox Star workstation που εμφανίστηκε το 1981, ενώ έγινε ευρύτερα γνωστό το 1984, με τον Apple Macintosh. Έκτοτε, με την βοήθεια των Microsoft Windows που κυριάρχησαν την αγορά, παραμένει ο απόλυτος κυρίαρχος στον τομέα των προσωπικών υπολογιστών μέχρι και σήμερα [1].

Ο βασικός τρόπος, λοιπόν, με τον οποίο αλληλεπιδρούμε με τους υπολογιστές προέκυψε πριν από 28 χρόνια, και παρέμεινε κατά βάση ο ίδιος, παρόλη τη δραστική αλλαγή του υπόλοιπου σκηνικού, πράγμα που δείχνει πόσο δύσκολη υπόθεση είναι η δημιουργία νέων λειτουργικών interface και η αποδοχή τους από το κοινό.

Η ανάγκη, όμως, για μία δραστική αλλαγή είναι επιτακτικότερη παρά ποτέ. Οι υπολογιστές απευθύνονται πλέον στο σύνολο της κοινωνίας, σε ανθρώπους κάθε οικονομικού και μορφωτικού επιπέδου. Χρειάζεται επομένως να “μιλάνε” τη γλώσσα των ανθρώπων, δηλαδή η διάδραση ανθρώπου-υπολογιστή να γίνεται με πιο φυσικούς τρόπους, περίπου όπως ο άνθρωπος αλληλεπιδρά με το φυσικό του περιβάλλον.

Το 1991 ο Mark Weiser περιέγραψε σε ένα δημοφιλές άρθρο του [2] το όραμά του για τον “υπολογιστή του 21^{ου} αιώνα”, ορίζοντας έτσι ένα νέο επιστημονικό τομέα, το ubiquitous (pervasive) computing. Το όραμα αυτό με το πέρασμα του χρόνου πραγματώνεται σταδιακά, με το μεγαλύτερο εμπόδιο να είναι ίσως η εξέλιξη του κατάλληλου λογισμικού. Κλείνοντας το άρθρο του, επισημαίνει:

“Το πιο σημαντικό είναι, ότι, οι ubiquitous computers θα βοηθήσουν να ξεπεράσουμε το πρόβλημα της υπερπληθώρας πληροφοριών. Υπάρχουν περισσότερες εν δυνάμει πληροφορίες κατά τη διάρκεια ενός περιπάτου στο δάσος, σε σχέση με οποιοδήποτε σύστημα υπολογιστών, κι όμως οι άνθρωποι βρίσκουν τον περίπατο στο δάσος χαλαρωτικό και τους υπολογιστές απωθητικούς. Μηχανές που θα προσαρμόζονται στον φυσικό κόσμο των ανθρώπων, αντί να υποχρεώνουν τους ανθρώπους να μπαίνουν στο δικό τους, θα κάνουν την χρήση του υπολογιστή τόσο ανανεωτική, όσο έναν περίπατο στο δάσος”

Μία λύση σε αυτή την κατεύθυνση, θα ήταν αυτή που θα συμπλήρωνε την φυσική αντίληψη μας για το περιβάλλον, προσθέτοντας ένα “πληροφοριακό στρώμα”. Οι πληροφορίες θα μπορούσαν να δίνονται στο χρήστη με ένα απόλυτα φυσικό τρόπο, σε αρμονία με το πως αυτός αντιλαμβάνεται και αλληλεπιδρά με το φυσικό περιβάλλον, εμπλουτίζοντας κατά κάποιο τρόπο τις ανθρώπινες αισθήσεις.

Η λύση αυτή θα μπορούσε να είναι η Augmented Reality, που θα μας απασχολήσει στο υπόλοιπο της εργασίας.

Περιεχόμενα

Ευχαριστίες	II
Περίληψη	III
Abstract	IV
Πρόλογος	V
Περιεχόμενα	VII
Ευρετήριο Αλγορίθμων	X
Ευρετήριο Εικόνων	XI
Ευρετήριο Πινάκων	XIV
1. Εισαγωγή στην AR	1
Ορισμός	1
Εφαρμογές.....	5
Τρόποι Επαύξησης της Πραγματικότητας.....	7
Η εξέλιξη της Mobile Augmented Reality	9
Handheld Augmented Reality	11
Ιστορική Αναδρομή της (Handheld) Augmented Reality	12
2. Εισαγωγή στο Visual Tracking	16
Feature Detection.....	16
Feature Extraction	17
Natural Feature Tracking,.....	17
Fiducial Marker Tracking	20
Visual Markers	22
Οι επιθυμητές ιδιότητες ενός AR marker system	27
3. Βιβλιογραφική επισκόπηση.....	29
ARToolkit	30
ARTag.....	34
ARToolkit+	39
Μερικές επιπλέον αναφορές	43
Unobtrusive Markers.....	43
Designable Markers.....	44
Bokode.....	45
4. Μαθηματικό υπόβαθρο	46

Εντοπίζοντας τις Edges	46
Ορισμός της Intensity Gradient.....	47
Εκτιμώντας την image Gradient	48
Derivative of Gaussian.....	50
Εντοπισμός της θέσης των edgels.....	54
Sub-Pixel Ακρίβεια.....	55
Ταυτοποιώντας το marker	57
Homogenous Coordinates.....	57
Projective Mapping	58
“Ευθύ” projective mapping.....	59
“Αντίστροφο” projective mapping	61
Μοντελοποίηση της κάμερας	68
Απλό Pinhole μοντέλο	68
Σχετικά με το principal point	71
Σχετικά με τα συστήματα συντεταγμένων.....	72
Σχετικά με τον αισθητήρα της κάμερας.....	73
Σχετικά με το φακό.....	74
Παράμετροι της κάμερας και εύρεσή τους.....	76
5. Προτεινόμενη Μεθοδολογία	79
Γενικά λόγια.....	79
Εντοπισμός των markers	80
Περί line detection	80
Εύρεση των Edgels	81
Εύρεση των Γραμμικών Τμημάτων	88
Συνένωση γραμμικών τμημάτων σε μεγαλύτερα.....	91
Επιμήκυνση των γραμμών	94
Εντοπισμός των περιγράμματα των markers	98
Ταυτοποίηση του ID των markers.....	104
Δειγματοληψία του marker	104
Αποκωδικοποίηση του marker.....	106
Ενώνοντας τα κομμάτια: Ο συνολικός αλγόριθμος	107
6. Πειραματική εφαρμογή	109
Εξεταζόμενο σενάριο εφαρμογής.....	109
Marker δεδομένων.....	109

Markers απλού ID.....	112
Camera Calibration.....	116
Pose Estimation	120
7. Συμπεράσματα	123
Γενικά	123
Τεχνικά χαρακτηριστικά.....	123
Τεχνικές παρατηρήσεις	124
Πιθανά προβλήματα	125
8. Μελλοντικές Επεκτάσεις	126
Βιβλιογραφία.....	127
Παράρτημα - Επεξήγηση κώδικα.....	135

Ευρετήριο Αλγορίθμων

- Προτεινόμενος αλγόριθμος για εντοπισμό edgels	81
- Προτεινόμενος αλγόριθμος για εντοπισμό μικρών γραμμικών τμημάτων.....	88
- Προτεινόμενος αλγόριθμος συνένωσης μικρών γραμμικών τμημάτων σε μεγαλύτερα.....	91
- Προτεινόμενος αλγόριθμος επιμήκυνσης γραμμών και προς τα 2 άκρα.....	95
- Προτεινόμενος αλγόριθμος επιμήκυνσης γραμμής προς ένα άκρο	96
- Προτεινόμενος αλγόριθμος εντοπισμού του περιγράμματος ενός marker.....	100
- Προτεινόμενος αλγόριθμος ανάγνωσης-δειγματοληψίας των "printed bits" ενός marker	105

Ευρετήριο Εικόνων

- Virtuality Continuum	2
- Διάφοροι τύποι HCI.....	3
- Εφαρμογή της AR στην ιατρική.....	5
- Εφαρμογή της AR στην πλοήγηση σε εσωτερικό και σε εξωτερικό χώρο (με παρουσίαση μνημείων - wiktitude).....	6
- Monitor-based AR σύστημα.....	7
- Optical-see-through HMD	8
- Ένα Video-see-through HMD	8
- Η εξέλιξη των φορητών AR συστημάτων	9
- Mobile AR σύστημα πλοήγησης σε εξωτερικό χώρο.....	9
- Ένα σύγχρονο HMD (Vuzix Wrap 920AV).....	10
- Ένα ανερχόμενο είδος φακών επαφής για AR εφαρμογές.....	10
- Keypoint Descriptor μέσω image gradients.....	17
- Εύρεση Features σε εικόνα.....	18
- Αντιστοίχιση των features σε δύο εικόνες-όψεις του ίδιου αντικειμένου.....	19
- Παράδειγμα marker-based AR εφαρμογής.....	21
- Παράδειγμα χρήση marker σε εφημερίδα για πρόσβαση σε ψηφιακή πληροφορία	22
- Σύγκριση 1D και 2D barcode συστημάτων	23
- Σύγκριση “database” και “index-based” 2D marker συστημάτων	25
- Μερικά παραδείγματα διάφορων 2D barcode συστημάτων.....	26
- Μερικά παραδείγματα markers του ARToolkit	30
- Απεικόνιση των βασικών βημάτων της λειτουργίας του ARToolkit	31
- Σχεδιασμός markers με χρήση DCT	32
- Παραδείγματα false-positive ανίχνευσης markers με το ARToolkit.....	33
- Η επίδοση του ARToolkit υπό ανομοιόμορφο φωτισμό	33
- Τα scanlines για την ανάγνωση και τα 12 προαποθηκευμένα patterns για κάθε marker (ARToolkit).....	33
- Παραδείγματα markers του ARTag.....	34
- Απεικόνιση των βασικών βημάτων της λειτουργίας του ARTag	35
- Η επίδοση του ARTag υπό ανομοιόμορφο φωτισμό.....	36
- Ανίχνευση με το ARTag ενός marker υπό μερική επικάλυψη	36
- Η διαδικασία κωδικοποίησης της πληροφορίας από το ARTag	37
- Η διαδικασία αποκωδικοποίησης της πληροφορίας από το ARTag	38
- Απεικόνιση των βασικών βημάτων της λειτουργίας του ARToolkitplus	39
- Η επίδραση του vignetting κατά το binarization της εικόνας και η αντιμετώπισή του από το ARToolkitplus.....	41
- Παρουσίαση των δομικών-λειτουργικών μονάδων του ARToolkitplus	42
- Μερικά είδη marker που ελαχιστοποιούν την προκαλούμενη “οπτική μόλυνση”	43
- Designable markers, πλήρως “αναγνώσιμα” από τον άνθρωπο	44
- Η χρήση του φαινομένου bokeh στην ανίχνευσή των markers	45
- Διάφορες περιπτώσεις εμφάνισης edges σε μία εικόνα.....	46

- Παραδείγματα της intensity gradient σε μία εικόνα	46
- Οι διάφορες περιπτώσεις της intensity gradient σε μία εικόνα.....	47
- Η επίδραση του θορύβου σε ένα 1D σήμα κατά την παραγωγή του	50
- Μορφή μιας Gaussian συνάρτησης	51
- Συνέλιξη ενός σήματος με Gaussian kernel και παραγωγή του αποτελέσματος.....	52
- Συνέλιξη ενός σήματος με Derivative of Gaussian kernel	53
- Δυσδιάστατη γραφική παράσταση μιας Gaussian και μιας Derivative of Gaussian συνάρτησης.....	53
- Εντοπισμός των edgels μέσω της intensity gradient	54
- Projective mapping.....	62
- Διάφοροι τύποι planar mapping	64
- Οι 3 κύριες περιπτώσεις κατά το projective mapping	65
- Pinhole φωτογραφική μηχανή.....	68
- Λειτουργία του pinhole μοντέλου	68
- Απλούστευση του pinhole μοντέλου	69
- Αναλυτικό διάγραμμα του pinhole μοντέλου.....	69
- Το σύστημα συντεταγμένων της κάμερας και το WDS (World Coordinate System).....	72
- Η σχέση των διαφόρων συστημάτων συντεταγμένων με τις παραμέτρους της κάμερας... ..	76
- Rotation, Translation και Scaling ενός σώματος.....	78
- Αναπαράσταση Rotation και Translation μιας κάμερας	78
- Η εικόνα που θα υποστεί επεξεργασία από τους προτεινόμενους αλγορίθμους για την παρουσίαση της λειτουργίας τους.....	82
- Διαχωρισμός της εικόνας σε ισομερείς περιοχές για τοπική επεξεργασία, και απεικόνιση του	82
- Γραφική απεικόνιση της intensity gradient σε κάθε σημείο μιας εικόνας.....	83
- Ανίχνευση edgels χωρίς sub-pixel ακρίβεια.....	84
- Ανίχνευση edgels με sub-pixel ακρίβεια.....	84
- Ανίχνευση edgels σε ολόκληρη την εικόνα (πυκνό πλέγμα scanlines).....	85
- Ανίχνευση edgels σε ολόκληρη την εικόνα (αραιό πλέγμα scanlines)	85
- Εντοπισμός των edgels ενός marker με πυκνό πλέγμα scanlines	86
- Εντοπισμός των edgels ενός marker με αραιό πλέγμα scanlines.....	86
- Επίδραση χαμηλής τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels	87
- Επίδραση μέσης τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels	87
- Επίδραση υψηλής τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels	87
- Εντοπισμός γραμμικών τμημάτων (πυκνό πλέγμα scanlines)	89
- Εντοπισμός γραμμικών τμημάτων (πυκνό πλέγμα scanlines)	89
- Επίδραση της orthogonal regression στην κλίση μιας γραμμής	90
- Γραμμικά τμήματα πριν τη συνένωση (πυκνό πλέγμα scanlines)	92
- Γραμμικά τμήματα μετά τη συνένωση (πυκνό πλέγμα scanlines)	92
- Γραμμικά τμήματα πριν τη συνένωση (αραιό πλέγμα scanlines)	93
- Γραμμικά τμήματα μετά τη συνένωση (αραιό πλέγμα scanlines).....	93
- Επέκταση παραμορφωμένου γραμμικού τμήματος.....	94
- Επέκταση γραμμικού τμήματος και συνθήκη τερματισμού	94
- Γραμμικά τμήματα πριν την επιμήκυνση	97
- Γραμμικά τμήματα μετά την επιμήκυνση.....	97

- Παράδειγμα “κατευθύνσεων” των πλευρών ενός τετραπλεύρου	98
- Οι διευθύνσεις και οι “κατευθύνσεις” όλων των edges σε ένα frame.....	99
- Περίπτωση positive false detection (γραμμές)	102
- Περίπτωση positive false detection (τετράπλευρο)	102
- Οι edges του marker πριν την ανίχνευση του περιγράμματός του	103
- Ανίχνευση του περιγράμματος του marker	103
- Projective mapping στον προτεινόμενο αλγόριθμο	104
- Αντιστοίχιση σημείων από μοναδιαίο τετράγωνο προς τετράπλευρο στην εικόνα	104
- Το περίγραμμα του marker και η δειγματοληψία του εσωτερικού του	106
- Δειγματοληψία στο εσωτερικό του marker (canonical view).....	106
- Το αποτέλεσμα της δειγματοληψίας του εσωτερικού του marker	106
- Εύρεση των “printed bits” του marker με δυαδικοποίηση (Otsu)	106
- Το ανώτατο επίπεδο της αλγοριθμικής ακολουθίας.....	107
- Αλγοριθμική ιεραρχία	108
- Παράδειγμα κώδικα ASCII.....	110
- Markers δεδομένων - Δειγματοληψία του εσωτερικού τους.....	110
- Markers δεδομένων - Αποκωδικοποίηση και παρουσίαση των δεδομένων	111
- Εφαρμογή για την ανάπτυξη markers μεγέθους 6x6 με BCH(31,16)	112
- Δημιουργία ενός 6x6 BCH marker (1 ^η εικόνα)	113
- Δημιουργία ενός 6x6 BCH marker (2 ^η εικόνα - προσθήκη ενός επιπλέον bit πληροφορίας)	113
- Δειγματοληψία μερικών 6x6 BCH υπό καλές φωτιστικές συνθήκες.....	114
- Ανίχνευση και αποκωδικοποίηση μερικών 6x6 BCH υπό καλές φωτιστικές συνθήκες	114
- Δειγματοληψία μερικών 6x6 BCH υπό κακές φωτιστικές συνθήκες	115
- Ανίχνευση και αποκωδικοποίηση μερικών 6x6 BCH υπό κακές φωτιστικές συνθήκες	115
- Οι διάφορες όψεις της “σκακιέρας” για το calibration σε ανάλυση 320x240	116
- Οι εξωτερικές παράμετροι του calibration set ως προς την κάμερα	117
- Οι εξωτερικές παράμετροι του calibration set ως προς το WCS	117
- Εσωτερικές παράμετροι μετά το calibration για ανάλυση 320x240	118
- Εσωτερικές παράμετροι μετά το calibration για ανάλυση 800x600	118
- Η ακτινική παραμόρφωση του φακού.....	118
- Η tangential παραμόρφωση του φακού	119
- Η συνολική παραμόρφωση του φακού	119
- Το μοντέλο που χρησιμοποιείται για την pose estimation με τον RPP	120
- Τα σημεία εισόδου στον αλγόριθμο RPP για την pose estimation	121
- Το αποτέλεσμα μετά την pose estimation.....	121
- Παράδειγμα επόπτευσης ιατρικών εξετάσεων με AR μέθοδο.....	122
- Τα ονόματα των αρχείων του MATLAB και η αλγοριθμική ιεραρχία	135
- Τα ονόματα των αρχείων του MATLAB και το ανώτατο επίπεδο της αλγοριθμικής ιεραρχίας	136

Ευρετήριο Πινάκων

Πίνακας 1 - Παραδείγματα μερικών 2D Barcode συστημάτων	25
Πίνακας 2 - Τα βασικά βήματα της λειτουργίας του ARToolkit.....	30
Πίνακας 3 - Διάφορες περιπτώσεις προσέγγισης της μερικής παραγωγού	48
Πίνακας 4 - Παράσταση σημείων με Ομογενείς συντεταγμένες.....	57
Πίνακας 5 - Συνήθης παράσταση 2D σημείου με ομογενείς συντεταγμένες.....	58
Πίνακας 6 - Αλλαγή από Ομογενείς σε Καρτεσιανές συντεταγμένες.....	58
Πίνακας 7 - Αντιστοίχιση σημείων για projective mapping.....	63
Πίνακας 8 - Οι εξισώσεις του projective mapping	63
Πίνακας 9 - Αντιστοίχιση σημείων: από μοναδιαίο τετράγωνο σε τετράπλευρο	65

1. Εισαγωγή στην AR

Ορισμός

Η “επαυξημένη πραγματικότητα” είναι ένα εκ βάθρων διαφορετικό είδος User Interface, που υπόσχεται να φέρει την επανάσταση στον τρόπο που αλληλεπιδρούμε με τους υπολογιστές, αναζητούμε και λαμβάνουμε πληροφορίες, και κατ’ επέκταση στον τρόπο που αντιλαμβανόμαστε ή αλληλεπιδρούμε με το περιβάλλον μας.

Η φιλοσοφία του είναι εκ διαμέτρων αντίθετη με αυτή του τύπου HCI¹ που επικρατεί εδώ και δεκαετίες, μέχρι και σήμερα. Η υπάρχουσα κατάσταση απαιτεί την πλήρη προσοχή του χρήστη σε μία οθόνη, “αποκόπτοντάς” τον από τον περιβάλλοντα χώρο του και “δεσμεύοντας” τα χέρια του πάνω στο πληκτρολόγιο και το ποντίκι (βλ εικόνα 2). Έτσι, κατά κάποιο τρόπο, ο χρήστης ξεφεύγει από τον φυσικό του κόσμο και εισέρχεται στον κόσμο των υπολογιστών προς αναζήτηση πληροφοριών.

Η “επαυξημένη πραγματικότητα”, αντιθέτως, προσπαθεί να προσαρμόσει τους υπολογιστές στον πραγματικό κόσμο, και να αποτελέσει ένα “πρόσθετο στρώμα πληροφορίας” ανάμεσα στον πραγματικό κόσμο και στον άνθρωπο. Αυτό το πρόσθετο στρώμα “εμπλουτίζει” τις αισθήσεις του ανθρώπου, δίνοντας του πληροφορίες που δε θα μπορούσε να συλλάβει με τα αισθητήρια όργανά του και τις παρουσιάζει όχι σε κάποια ειδική οθόνη, αλλά ακριβώς εκεί όπου ανήκουν: στον πραγματικό κόσμο.

Ας φανταστούμε για παράδειγμα έναν χρήστη να μπαίνει σε ένα εμπορικό κέντρο ψάχνοντας τους φίλους του. Τα γυαλιά ηλίου του “καθαρίζουν” και εμφανίζουν στο πάτωμα εικονικά βέλη, που του δείχνουν την κατεύθυνση στην οποία πρέπει να κατευθυνθεί. Ή, σαν δεύτερο παράδειγμα, έναν πυροσβέστη που μπαίνει σε ένα φλεγόμενο νοσοκομείο γεμάτο καπνό. Η μάσκα του απεικονίζει το περίγραμμα των τοίχων, των πορτών και των σκαλιών, και με εικονικά σημάδια δείχνει την θέση των ασθενών. Αυτά τα παραδείγματα μπορεί να ακούγονται προς το παρόν σαν σενάρια επιστημονικής φαντασίας, θα είναι όμως εφικτά στο κοντινό μέλλον χάρη σε φορητούς (wearable) υπολογιστές με λογισμικό επαυξημένης πραγματικότητας [3].

Ακόμα και σήμερα δεν υπάρχει κάποιος καθολικά αποδεκτός ορισμός της επαυξημένης πραγματικότητας, με αποτέλεσμα να υπάρχουν διάφορες ερμηνείες του όρου, είτε πιο αυστηρές είτε πιο ελεύθερες. Ο πιο ευρέως αποδεκτός, όμως, ορισμός είναι ο ορισμός του Azuma σύμφωνα με τον οποίο ένα AR σύστημα θα πρέπει να ικανοποιεί 3 συνθήκες [4]:

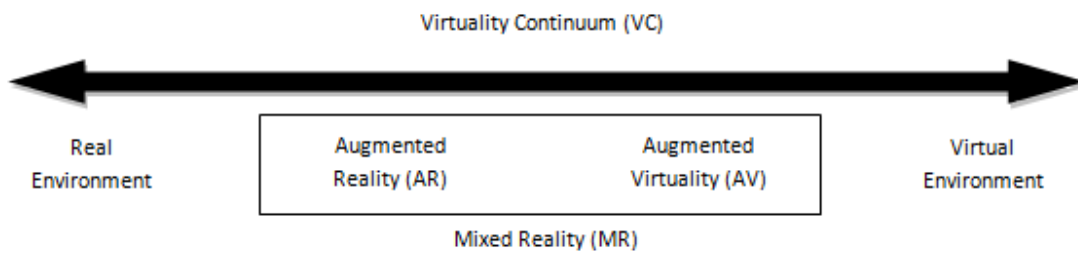
- Να συνδυάζει το πραγματικό με το εικονικό
- Να είναι αλληλεπιδραστικό (interactive) σε πραγματικό χρόνο
- Να λειτουργεί αναφορικά με τον πραγματικό τρισδιάστατο κόσμο²

Πάνω στον ορισμό του Azuma, θα πρέπει να κάνουμε τις ακόλουθες παρατηρήσεις:

¹ Human-Computer Interaction

² “to be registered in the 3D world”

- Η πρώτη συνθήκη είναι το βασικότερο ίσως χαρακτηριστικό της AR: ο συνδυασμός πραγματικού και εικονικού. Έτσι, η AR εμπλουτίζει τον τρόπο με τον οποίο ο άνθρωπος αντιλαμβάνεται τον πραγματικό κόσμο, σε σχέση με:
 - Την όραση, στην οποία εστιάζουν οι περισσότεροι ερευνητές του χώρου (πχ [5]),
 - Την ακοή (πχ [6], [7])
 - Την όσφρηση
 - Την αφή
- Η δεύτερη συνθήκη επιβάλλει την AR να λειτουργεί σε πραγματικό χρόνο, διαφοροποιώντας την από το post-processing, που γίνεται παραδείγματος χάρη στα στούντιο κινηματογραφικών ταινιών.
- Η τρίτη συνθήκη διαφοροποιεί την AR από τις διάφορες εκφάνσεις της “μικτής πραγματικότητας” (Mixed Reality), απαιτώντας το πραγματικό στοιχείο του πραγματικού 3D κόσμου να είναι κυρίαρχο, και το εικονικό να “διδασκάλει” σε αυτό. Η απαίτηση αυτή γίνεται καλύτερα κατανοητή διαισθητικά στην εργασία των Milgrim και Kishino [8].



Εικόνα 1 - Virtuality Continuum

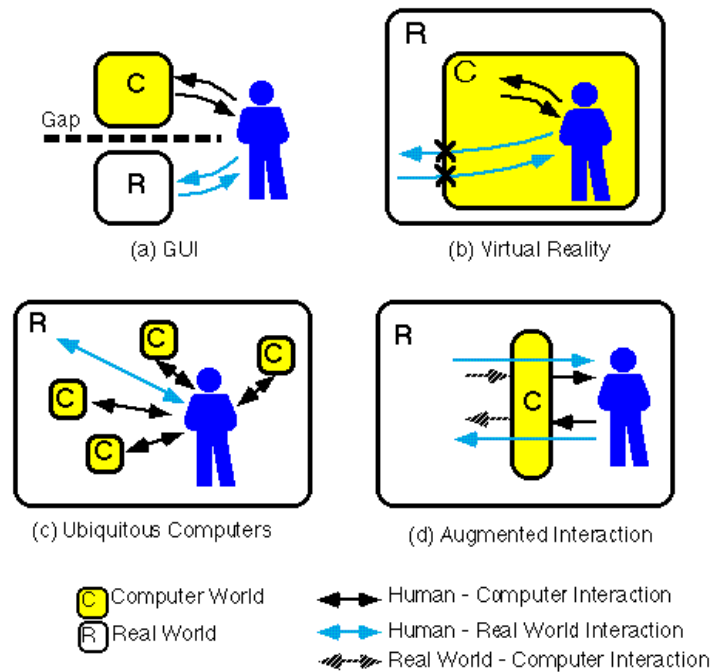
Στην εργασία τους αυτή οι Milgrim και Kishino - απεικονίζουν τις διάφορες εκφάνσεις της μίξης του πραγματικού με τον εικονικό κόσμο πάνω στο “virtuality continuum”, όπως φαίνεται στην παραπάνω εικόνα. Σε αυτό παρατηρούμε:

- Τον πραγματικό κόσμο (αριστερά) έτσι όπως τον αντιλαμβανόμαστε με τις αισθήσεις μας ή μέσω μίας “απευθείας” πιστής απεικόνισής του (πχ μέσω κάμερας, ηχείων κτλ).
- Τον εικονικό κόσμο (δεξιά), γνωστότερο και σαν “εικονική πραγματικότητα” (Virtual Reality), στον οποίο διεισδύει ολοκληρωτικά ο χρήστης και αλληλεπιδρά με αποκλειστικά εικονικά κατασκευάσματα (πχ “second life”³).
- Την Μικτή Πραγματικότητα (Mixed Reality), στο ενδιάμεσο, που περιέχει σε κάποιο βαθμό στοιχεία και των δύο κόσμων. Μπορούμε να ορίσουμε δύο εκφάνσεις της Μικτής Πραγματικότητας, χωρίς όμως σαφή διαχωρισμό ανάμεσα τους:

³ <http://secondlife.com/>

- Την “επαυξημένη πραγματικότητα” (Augmented Reality), στην οποία είναι κυρίαρχο το στοιχείο του πραγματικού κόσμου, έναντι του εικονικού.
- Την “επαυξημένη εικονικότητα” (Augmented Virtuality), στην οποία είναι κυρίαρχο το στοιχείο του εικονικού κόσμου, έναντι του πραγματικού.

Γενικές έρευνες πάνω στην AR, που θα μπορούσαν να λειτουργήσουν και σαν εισαγωγή στην ενασχόληση κάποιου ενδιαφερόμενου με το αντικείμενο, είναι οι εργασίες [4] και [9].



Εικόνα 2 - Διάφοροι τύποι HCI

Η πρώτη εργασία σχετική με την AR, και το πρώτο AR σύστημα (θεωρήθηκε επίσης και το πρώτο VR σύστημα), παρουσιάστηκαν από τον Ivan Sutherland και τους φοιτητές του στο Harvard University και στο University of Utah, κατά τη δεκαετία του '60 [10] [11]. Στο σύστημα αυτό, το οποίο μάλιστα είναι συμβατό με τον (κατά πολύ μεταγενέστερο) ορισμό του Azuma, ο χρήστης “φορούσε” μία ημιδιαφανή οθόνη προσαρμοσμένη στο ύψος των ματιών (see-through Head Mounted Display) στην οποία προβάλλονταν απλά 3D μοντέλα, αποτελούμενα μόνο από τις κύριες ακμές τους, λόγω των τεχνολογικών περιορισμών της εποχής.

Στις δεκαετίες του '70 και του '80 μόνο ένας μικρός αριθμός ερευνητών ασχολούνταν με την AR, σε ερευνητικά ινστιτούτα όπως το U.S. Air Force's Armstrong Laboratory, το NASA Ames Research Center και το University of North Carolina at Chapter Hill.

Ο όρος “Augmented Reality” επινοήθηκε μόλις το 1992 από τους Tom Caudell και David Mizel, οι οποίοι δούλευαν για την Boeing πάνω σε ένα σύστημα υποβοήθησης των εργατών στη συναρμολόγηση των καλωδιώσεων [12].

Τότε διαχωρίστηκε η επαυξημένη από την εικονική πραγματικότητα, σηματοδοτώντας τη δημιουργία ενός νέου επιστημονικού κλάδου. Το πρώτο εξειδικευμένο επιστημονικό συνέδριο⁴ πάνω στην AR έγινε, όμως, αρκετά αργότερα, μόλις 10 χρόνια πριν από σήμερα.

Η κυριότερη έρευνα αυτής της τελευταίας δεκαετίας συντελέστηκε στα πλαίσια του συνεδρίου ISMAR⁵ και των προγόνων του (IWAR⁶, ISMR⁷, ISAR⁸), ενώ μία σύνοψη αυτής δίνεται στην εργασία των Zhou et al [13].

Παρόλο που τα τελευταία χρόνια έχει γίνει σημαντική έρευνα πάνω στον τομέα και έχουν αναπτυχθεί πολλά prototypes και frameworks, δεν έχει εμφανιστεί ακόμα κάποια εφαρμογή αρκετά πρακτική, εύχρηστη και αποτελεσματική ώστε να βρει καθολική ανταπόκριση από το κοινό.

Η AR φαίνεται, όμως, να έχει ωριμάσει πλέον αρκετά, ώστε αρκετά projects είναι έτοιμα να περάσουν στο ευρύ κοινό και να δοκιμαστούν. Ειδικότερα με την έλευση των smartphones και την διάδοσή τους, το έδαφος είναι πιο πρόσφορο παρά ποτέ για νέες εφαρμογές που θα απογειώσουν την AR και τη διεισδυτικότητα των υπολογιστών στη ζωή μας.

Έτσι, δεν είναι τυχαίο που με αφορμή την ενασχόληση της NOKIA με την AR, το ηλεκτρονικό περιοδικό Technology Review⁹, στο τεύχος Μαρτίου/Απριλίου του 2007, παρουσιάζει την AR σαν μία εκ των 10 σημαντικότερων αναδυόμενων τεχνολογιών της τότε χρονιάς [14].

⁴ IWAR '98

⁵ ISMAR - International Symposium on Mixed and Augmented Reality

⁶ IWAR - International Workshop on Augmented Reality

⁷ ISMR - International Symposium on Mixed Reality

⁸ ISAR - International Symposium on Augmented Reality

⁹ <http://www.technologyreview.com/>

Εφαρμογές

Η AR θα μπορούσε να βρει εφαρμογή σε κάθε ανθρώπινη δραστηριότητα, λειτουργώντας σαν “Intelligence Amplification” (ενισχυτής νοημοσύνης), όπως ορίζει ο Brooks τη χρήση του υπολογιστή σαν εργαλείο υποβοήθησης του ανθρώπου στις διάφορες εργασίες του [15].

Γενικότερα, οι εν δυνάμει εφαρμογές θα μπορούσαν να εμφανιστούν στους παρακάτω τομείς [4], [9], [16], [17]:

Ιατρική

Τα συστήματα AR με HMD μπορούν να λειτουργήσουν στον τομέα της ιατρικής σαν:

- Εργαλείο εκπαίδευσης (πχ [18]). Εικονικές πληροφορίες μπορούν να βοηθήσουν τον εκπαιδευόμενο με τα απαιτούμενα βήματα μίας εξέτασης ή επέμβασης, χωρίς να απαιτείται η απόσπαση της προσοχής του σε κάποια οθόνη. Επίσης μπορεί να γίνεται εντοπισμός των διαφόρων οργάνων και η αναγνώρισή τους με ειδικές ενδείξεις (annotations).
- Νέα μέθοδος οπτικοποίησης ιατρικών δεδομένων, είτε από διάφορους αισθητήρες είτε με τη χρήση non-invasive μεθόδων, όπως MRI¹⁰, CT¹¹ scans ή ultrasound imaging (πχ [19]). Οι πληροφορίες αυτές θα μπορούσαν να αποδοθούν οπτικά στο HMD σε πραγματικό χρόνο, επικαθήμενες στην πραγματική όψη του ασθενή, και δίνοντας έτσι στο γιατρό τη δυνατότητα “real time X-ray όρασης” (βλ εικόνα 3) μέσα στον ασθενή. Κάτι τέτοιο θα οφελούσε, για παράδειγμα, στη γρηγορότερη εξαγωγή πληροφοριών και συμπερασμάτων από τους γιατρούς, στη διευκόλυνση των χειρουργικών επεμβάσεων (ιδιαίτερα των minimal-invasive) και στη μείωση των τομών πάνω στον ασθενή



Εικόνα 3 - Εφαρμογή της AR στην ιατρική

Κατασκευές/Επισκευές - Manufacturing and Repair

Μία σημαντική εφαρμογή είναι η υποβοήθηση στην ανάπτυξη και συντήρηση ειδικών κατασκευών ή στη συναρμολόγηση και επισκευή πολύπλοκων μηχανών και συστημάτων. Αντί των πολυσέλιδων manuals που χρησιμοποιούνται σήμερα, θα μπορούσαν να παρουσιάζονται σχέδια (πχ υπόγειων σωληνώσεων και καλωδιώσεων [20]) ή οδηγίες και επεξηγήσεις σε πραγματικό χρόνο, με γραφικά επικαθήμενα στην πραγματική όψη του εξοπλισμού [21].

Ψυχαγωγία

Η AR άρχισε πρόσφατα να χρησιμοποιείται και σε ηλεκτρονικά παιχνίδια [22], [23], [24], όπου ακόμα και σήμερα κυριαρχεί αποκλειστικά το εικονικό στοιχείο, καθώς και σε προσπάθειες διαφορετικού τρόπου αφήγησης ιστοριών (storytelling) [25].

¹⁰ MRI - Magnetic Resonance Imaging

¹¹ CT - Computed Tomography

Χειρισμός Οχημάτων

Η AR μπορεί να αποτελέσει σημαντικό βοήθημα στην πλοήγηση οχημάτων, όπως τα αεροπλάνα ή τα αυτοκίνητα. Με τη χρήση ενός HMD ή ενός HUD μπορούν να προβάλλονται χρήσιμες πληροφορίες σχετικά με την πλοήγηση, την κατάσταση του οχήματος, την ανίχνευση πιθανών κινδύνων, το που πρέπει να εστιαστεί η προσοχή του χειριστή, κτλ.

Συνεργατικές Εφαρμογές

Η AR μπορεί να βοηθήσει στη συνεργασία μεταξύ των μελών μίας ομάδας, μέσω συνεδριών με πραγματικούς και εικονικούς συμμετέχοντες, ή γενικότερα με την ταυτόχρονη πρόσβαση στην ψηφιακή πληροφορία και ομαδική αλληλεπίδραση με αυτή.

Εκπαίδευση - Πολιτισμός

Η AR μπορεί να έχει σημαντικές εφαρμογές στον τρόπο που μαθαίνουμε [26] [5] ή που βιώνουμε χώρους ειδικού ενδιαφέροντος, όπως ένα μουσείο [27] ή έναν αρχαιολογικό/ιστορικό χώρο [28] [6], του οποίου η “αύρα”¹² [29] γίνεται αισθητή πιο άμεσα και ρεαλιστικά.

Διαφήμιση

Η AR μπορεί να χρησιμοποιηθεί και για διαφημιστικούς σκοπούς, ενώ μέχρι τώρα χρησιμοποιείται κυρίως μέσα από διαδικτυακές, διαδραστικές εφαρμογές.

Σε streaming ζωντανών εκπομπών

Η AR χρησιμοποιείται και στην εκπομπή εμπλουτισμένου video, ώστε να εισάγει άμεσα ψηφιακές πληροφορίες στο εκπεμπόμενο βίντεο σε πραγματικό χρόνο, όπως για παράδειγμα στους αθλητικούς αγώνες [30].

Πλοήγηση σε άγνωστο χώρο

Μία ακόμη εφαρμογή της AR είναι η πλοήγηση, τόσο σε εσωτερικό (εικόνα 4 - αριστερά) [31] [32], όσο και σε εξωτερικό χώρο (εικόνα 4 - δεξιά) [33] [34]. Μάλιστα, από τη στιγμή που για την πλοήγηση γίνεται “ανάγνωση” της θέσης του χρήστη, η πλοήγηση μπορεί να συνδυαστεί και με “location based services”, δίνοντας έτσι τη δυνατότητα πιο εξειδικευμένων εφαρμογών (όπως τουριστικών οδηγιών κτλ).



Εικόνα 4 - Εφαρμογή της AR στην πλοήγηση σε εσωτερικό και σε εξωτερικό χώρο (με παρουσίαση μνημείων - wikitude)

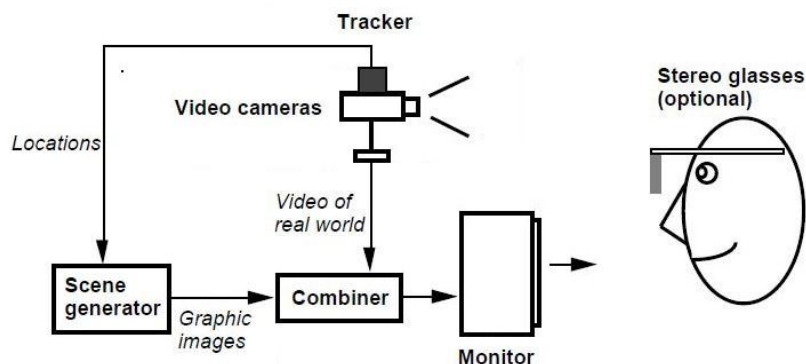
¹² “Aura describes the cultural and personal significance that a place (or object) holds for an individual”

Τρόποι Επαύξησης της Πραγματικότητας

Γενικότερα, σε ένα AR σύστημα μπορούμε να εμπλουτίσουμε την πραγματικότητα κυρίως με 2 τρόπους [35]:

“Μαγικός καθρέφτης”

Με τη χρήση μίας κάμερας και μίας οθόνης, όπως φαίνεται στην παρακάτω εικόνα, οι χρήστες βλέπουν το είδωλο του πραγματικού κόσμου ή/και του εαυτού τους, με το ψηφιακό στοιχείο να επικάθεται στο πραγματικό (πχ [36]).



Εικόνα 5 - Monitor-based AR σύστημα

“Μαγικός φακός”

Με τη χρήση μιας οθόνης και μίας κάμερας που κοιτάζει προς την αντίθετη κατεύθυνση σε σχέση με την εικόνα 5, ο χρήστης “βλέπει” διάμεσο (see-through) του AR “φακού” μία επαυξημένη πραγματικότητα.

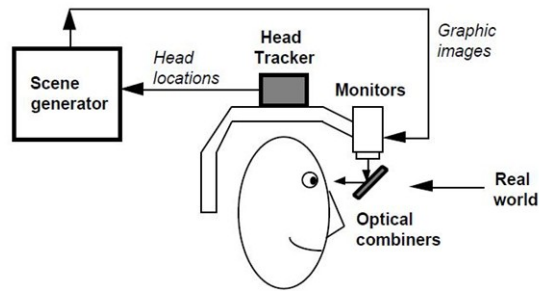
Μπορούμε να διακρίνουμε 2 τρόπους δημιουργίας “μαγικών φακών” [4]:

Optical see-through AR

Ο χρήστης φοράει στο κεφάλι¹³ ένα HMD με optical combiners στο ύψος των ματιών (βλ εικόνα 6). Αυτοί είναι μερικώς διαφανείς, ώστε ο χρήστης να έχει απευθείας αίσθηση του πραγματικού κόσμου.

Ταυτόχρονα είναι μερικώς ανακλαστικοί, έτσι ώστε να αντανακλώνται πάνω του οι ακτίνες από ειδικά προσαρμοσμένες μικρές οθόνες και να καταλήγουν στο μάτι, όπου έχουμε μία μίξη της πραγματικής και της εικονικής πληροφορίας.

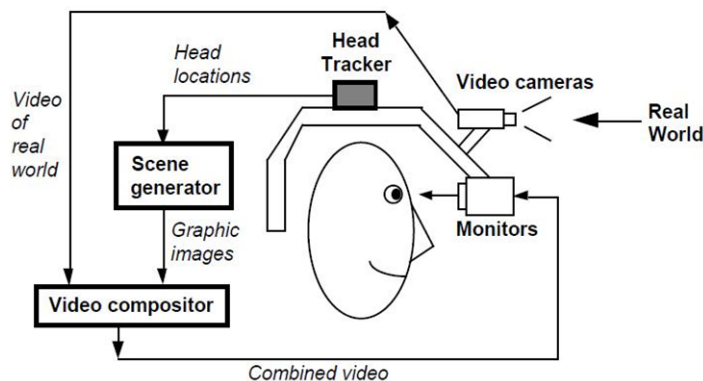
¹³ Έτσι το HMD διαφοροποιείται από το HUD (Head Up Display) που χρησιμοποιείται στα αεροσκάφη ή στα παμπρίζ των αυτοκινήτων, και είναι μία οθόνη στο τζάμι του οχήματος και στο ύψος του κεφαλιού. Κατά κάποιον τρόπο το HMD είναι ένα wearable HUD [43]



Εικόνα 6 - Optical-see-through HMD

Video see-through AR

Μπορούμε να δημιουργήσουμε ένα video see-through AR σύστημα είτε με τη βοήθεια ενός closed view see-through HMD¹⁴ και μίας¹⁵ head mounted κάμερας είτε με τη βοήθεια φορητών συσκευών σε συνδυασμό με μία κάμερα (βλ εικόνα 7).



Εικόνα 7 - Ένα Video-see-through HMD

Ο χρήστης δεν αντιλαμβάνεται τον πραγματικό κόσμο απευθείας με τα μάτια του, αλλά με διαμεσολάβηση μίας κάμερας και μίας οθόνης που τον προβάλλει. Ταυτόχρονα, σε κάθε frame επικάθονται στην εικόνα του πραγματικού κόσμου και εικονικές πληροφορίες, και το τελικό αποτέλεσμα προβάλλεται σε μία οθόνη στο ύψος των ματιών.

Η περίπτωση του “μαγικού καθρέφτη” χρησιμοποιείται κυρίως σε σταθερούς υπολογιστές, όπως σε απλούς desktop ή ειδικότερους υπολογιστές, ενώ η περίπτωση του “μαγικού φακού” χρησιμοποιείται σε φορητά συστήματα, όπως τα κινητά τηλέφωνα.

Περισσότερες πληροφορίες για αυτά τα φορητά συστήματα παρουσιάζονται στην επόμενη παράγραφο.

¹⁴ Τα see-through HMDs (είτε optical είτε video see-through) επιτρέπουν το συνδυασμό του πραγματικού με το εικονικό στοιχείο, σε αντίθεση με τα πιο γνωστά closed-view HMDs που χρησιμοποιούνται στην VR και εισάγουν το χρήστη σε ένα πλήρως εικονικό περιβάλλον.

¹⁵ ή δύο για στέρεο όρασης

Η εξέλιξη της Mobile Augmented Reality

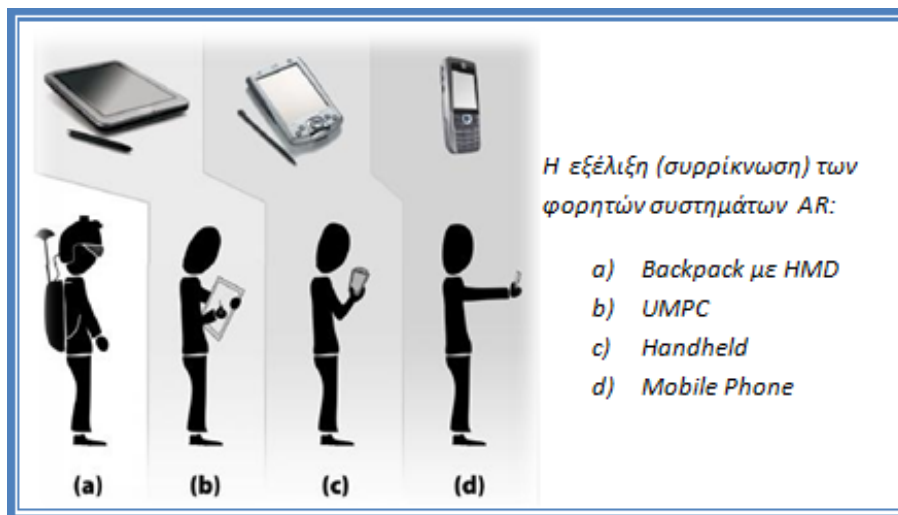
Από την αρχή της ενασχόλησης των ερευνητών με την AR φάνηκε το ενδιαφέρον να ξεφεύγει κατά κύριο λόγο από τους σταθερούς υπολογιστές, και να εστιάζεται στα φορητά συστήματα, όπου φαίνονται οι πραγματικές δυνατότητες της AR με τις πολυποίκιλες δυνατότητες εφαρμογών.

Έτσι άρχισαν από πολύ νωρίς να εμφανίζονται φορητά συστήματα, όπως συστήματα καθοδήγησης σε εξωτερικό χώρο [33], παρά τους τεχνικούς περιορισμούς της εποχής. Εξαιτίας αυτών τα πρώτα φορητά συστήματα αποτελούνταν από ειδικά κατασκευασμένα συστήματα “πλάτης” (backpacks) που περιείχαν τον κατάλληλο υπολογιστικό εξοπλισμό και τους αισθητήρες θέσης ή κατεύθυνσης, καθώς και ογκώδη HMDs¹⁶ (βλ εικόνα 8). Το όλο σύστημα δεν ήταν μόνο βαρύ, ογκώδες και μη αξιόπιστο¹⁷, αλλά εισήγαγε και έναν επιπλέον σημαντικό περιορισμό: ένα τέτοιου είδους σύστημα δε θα γινόταν κοινωνικά αποδεκτό.



Εικόνα 8 - Mobile AR σύστημα πλοήγησης σε εξωτερικό χώρο

Με την εξέλιξη της τεχνολογίας τα backpacks αντικαταστάθηκαν από τα UMPCs¹⁸, και αυτά με τη σειρά τους από τα PDAs και αργότερα από τα κινητά τηλέφωνα (βλ εικόνα 9). Οι δύο τελευταίες κατηγορίες ενσωματώθηκαν πλέον στην κατηγορία των smartphones, τα οποία και επικράτησαν ολοκληρωτικά στα φορητά AR συστήματα [37].



Εικόνα 9 - Η εξέλιξη των φορητών AR συστημάτων

¹⁶ Αρχικά αυτά ήταν μεταποιημένα closed-view HMDs με προσθήκη κάμερας, λειτουργώντας έτσι σαν video see-through HMD

¹⁷ όλες οι τροποποιήσεις και συνδέσεις του εξοπλισμού γίνονταν στο εργαστήριο, ενώ ο εξοπλισμός δεν ήταν προστατευμένος από τις καιρικές συνθήκες

¹⁸ Ultra Mobile PC

Έτσι, ένα σημερινό AR σύστημα αποτελείται απλά από ένα smartphone, που περιέχει σε μία μικρή συσκευή όλους τους απαραίτητους αισθητήρες, ενώ αυτού του είδους οι συσκευές υιοθετούνται από ολοένα και μεγαλύτερο κοινό και είναι πλήρως κοινωνικά αποδεκτές.

Η εξέλιξη αυτή βέβαια, δεν έχει μόνο τα πλεονεκτήματα του μικρότερου μεγέθους, βάρους και κόστους ή της πιο αξιόπιστης κατασκευής. Απεναντίας είχε το σημαντικότερο μειονέκτημα της πτώσης της διαθέσιμης επεξεργαστικής ισχύος.

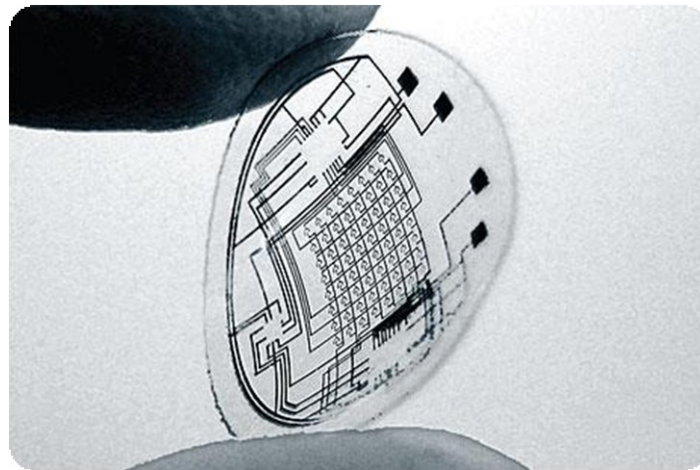
Παράλληλα με τα smartphones εξελίσσονται και άλλοι παρεμφερείς τομείς, όπως:

- Τα HMDs [38] (βλ εικόνα 10)
- Οι φακοί επαφής [39] με δυνατότητα προβολής εικόνας στο μάτι (βλ εικόνα 11)
- Οι Picoprojectors¹⁹



Εικόνα 10 - Ένα σύγχρονο HMD (Vuzix Wrap 920AV)

Με την εξέλιξή τους και τον συνδυασμό τους κυρίως με τα smartphones, αναμένεται να αναπτυχθούν και να γίνουν κοινωνικά αποδεκτά και τα wearable/pervasive AR συστήματα, οπότε θα πραγματοποιηθεί και η περαιτέρω διείσδυση της AR στην καθημερινότητά μας. [3] [40].



Εικόνα 11 - Ένα ανερχόμενο είδος φακών επαφής για AR εφαρμογές

¹⁹ πχ [40], www.microvision.com/pico_projector_displays

Handheld Augmented Reality

Τα κινητά τηλέφωνα είναι ίσως η πιο ευρέως διαδεδομένη ψηφιακή συσκευή στις μέρες μας. Σύμφωνα με το Gartner²⁰, το 2008 πωλήθηκαν πάνω από 1 δισεκατομμύριο (1.22 δισεκατομμύρια) τηλέφωνα, ενώ σύμφωνα με εκτιμήσεις, ο αριθμός αυτός εκτιμάται να αυξηθεί σε 1.8 δισεκατομμύρια το 2012, τα 800 εκατομμύρια εκ των οποίων αναμένεται να είναι smartphones [41].

Η νέα αυτή πλατφόρμα είναι λοιπόν ευρύτατα διαδεδομένη, κοινωνικά αποδεκτή, είναι κτήμα ανθρώπων κάθε κοινωνικού επιπέδου και συνοδεύει τον χρήστη σε κάθε χρονική στιγμή της ημέρας, προσδίδοντας έτσι μεγαλύτερο νόημα στην έννοια “προσωπικός υπολογιστής”. Επιπρόσθετα, όλοι σχεδόν είναι εξοικειωμένοι με τη χρήση της, κάτι που ευνοεί την εμφάνιση και υιοθέτηση νέων εφαρμογών.

Περίπου από το 2002, οπότε και σημειώθηκε μία σημαντική αύξηση της υπολογιστικής ισχύος των επεξεργαστών των τηλεφώνων, το ενδιαφέρον προς αυτή την πλατφόρμα μεγάλωσε [27]. Τα PDAs που μέχρι εκείνη τη στιγμή έτρεχαν απλό λογισμικό σαν organizer, απέκτησαν δυνατότητες multimedia ή gaming συσκευών, ενώ αργότερα ενώθηκαν με τα τηλέφωνα στην κατηγορία των smartphones.

Μιας και αυτά απευθύνονται σε ένα πολύ ευρύτερο κοινό σε σχέση με τους παλιότερους τύπους AR συστημάτων(αυτοσχέδια backpacks, UMPCs, PDAs), η τιμή, το βάρος, το μέγεθος τους και η διάρκεια της μπαταρίας τους προσαρμόστηκαν στις ανάγκες του.

Ταυτόχρονα, στοχεύοντας στη διατήρηση του ενδιαφέροντος του κοινού και για λόγους marketing, προσθέτονταν ολοένα και περισσότερες δυνατότητες, όπως κάμερες, δέκτες GPS, αδρανειακοί αισθητήρες και πρόσθετες δυνατότητες διασύνδεσης (IR, Bluetooth, Wifi και πρόσφατα WiMax).

Σαν αποτέλεσμα όλων των παραπάνω η Augmented Reality αποτελεί ίσως ιδανικό συμπλήρωμα του Mobile/Pervasive Computing, ενώ υπάρχει ένα μεγάλο εν δυνάμει κοινό ενδιαφερόμενο για τις εφαρμογές της.

²⁰ <http://www.gartner.com>

Ιστορική Αναδρομή της (Handheld) Augmented Reality

Στην παρούσα ενότητα παρουσιάζονται τα κυριότερα γεγονότα που επηρέασαν την εξέλιξη της Augmented Reality στα smartphones. Γίνεται αναφορά όχι μόνο σε εφαρμογές πάνω στα τηλέφωνα, αλλά και στις κυριότερες τεχνολογίες που χρησιμοποιούνται, καθώς και σε εφαρμογές που λόγω των τεχνικών περιορισμών της εποχής εφαρμόστηκαν αρχικά σε άλλου είδους συστήματα (laptops/backpacks), αργότερα όμως πέρασαν στα κινητά τηλέφωνα. Μία πιο πλήρης αναφορά μπορεί να βρεθεί στο ιστορικό αφιέρωμα του ISMAR 09 για την mobile AR [42].

1968

Δημιουργείται από τον Ivan Sutherland το πρώτο AR σύστημα (το οποίο ήταν επίσης και το πρώτο VR σύστημα). Στο σύστημα αυτό, με τη χρήση ενός see-through Head Mounted Display προβάλλονται στο χρήστη απλά 3D μοντέλα, αποτελούμενα μόνο από τις κύριες ακμές, λόγω των τεχνολογικών περιορισμών της εποχής. Το tracking του HMD γίνεται με δύο διαφορετικά συστήματα 6DoF [10]

1982

Παρουσιάζεται το πρώτο laptop, το Grid Compass 1100

1992

Επινοείται ο όρος “Augmented Reality” από τους Tom Caudell και David Mizel, οι οποίοι δούλευαν για την Boeing πάνω σε ένα σύστημα υποβοήθησης των εργατών στη συναρμολόγηση των καλωδιώσεων [43].

Επίσης παρουσιάζεται το πρώτο smartphone, το IBM Simon Personal Communicator, από την IBM και την Bellsouth.

1993

Η ανάπτυξη GPS (Global Positioning System) πρωτοφτάνει σε λειτουργικό επίπεδο, ενώ ολοκληρώνεται τον Απρίλιο του 1995 και αρχικά η χρήση του προορίζεται για στρατιωτικούς σκοπούς.

Αναπτύσσεται από τους Loomis et al. ένα πρωτότυπο σύστημα καθοδήγησης σε εξωτερικό χώρο ατόμων με προβλήματα όρασης. Το σύστημα αποτελείται από ένα laptop, έναν δέκτη differential GPS και μία ηλεκτρονική πυξίδα (στο κεφάλι του χρήστη), ενώ χρησιμοποιούνται δεδομένα από μία GIS²¹ βάση δεδομένων και οι οδηγίες γίνονται με ηχητικά σήματα (“acoustic visual display”)

1994

Οι Milgram και Kishimo ορίζουν το Virtual Reality Continuum, όπου φαίνονται οι διάφορες εκφάνσεις της μίξης του πραγματικού με τον εικονικό κόσμο [8] (βλ εικόνα 1). Σήμερα χρησιμοποιείται ευρέως για τον ορισμό της AR, μαζί με τον ορισμό του Azuma.

²¹ Geographical Information System

1995

Οι Rekimoto et al χρησιμοποιούν ένα σύστημα μονοδιάστατων barcodes για την παρουσίαση, μέσω ενός συστήματος με κάμερα, πληροφοριών σχετικών με συγκεκριμένα μέρη-αντικείμενα

1996

Το σύστημα GPS, που μέχρι τότε προοριζόταν μόνο για στρατιωτική χρήση, απελευθερώνεται για χρήση από το ευρύτερο κοινό.

Επίσης παρουσιάζεται από τον Jun Rekimoto ένα από τα πρώτα συστήματα camera tracking με 6DoF, που χρησιμοποιούν τετράγωνα 2D “barcodes”.

1997

Ο Ronal Azuma σε μία γενική εργασία πάνω στην AR δίνει τον δικό του ορισμό για την AR, ο οποίος είναι έκτοτε ο ευρύτερα αποδεκτός ορισμός της.

Οι Feiner et al. Παρουσιάζουν το πρώτο mobile AR σύστημα (Touring Machine) για εξερεύνηση εξωτερικών χώρων. Αποτελείται από ένα see-through HDM με ενσωματωμένους αισθητήρες κατεύθυνσης, ένα backpack με υπολογιστή, dGPS και digital radio για ασύρματη πρόσβαση στο διαδίκτυο.

Ο Philippe Kahn εφευρίσκει το πρώτο camera phone.

1999

Οι Hirokazu Kato και Mark Billinghurst παρουσιάζουν το ARToolkit [44], μία pose tracking βιβλιοθήκη (περισσότερες πληροφορίες στο), ενώ δίνεται στο κοινό σαν open source με GPL άδεια. Εξαιτίας αυτού του γεγονότος είναι πολύ δημοφιλής στον κύκλο της AR ακόμα και σήμερα.

Οι Höllerer et al παρουσιάζουν ένα mobile AR σύστημα, που καθοδηγεί το χρήστη μέσα στο πανεπιστημιακό campus, με τη δυνατότητα να επικάθονται στην όψη του πραγματικού κόσμου μοντέλα των παλαιότερων κτηρίων. Επίσης ο χρήστης έχει τη δυνατότητα να ακούσει ειδήσεις που συνδέονται με τα μέρη στα οποία αφορούν.

Εμφανίζεται το πρώτο GSM τηλέφωνο ([Benefon Esc! NT2002](#)) με ενσωματωμένο δέκτη GPS

Ορίζονται τα πρωτόκολλα 802.11a/802.11b – ευρύτερα γνωστά ως WiFi.

2000

Εμφανίζεται στην αγορά το πρώτο εμπορικό camera phone(Sharp [J-SH04](#))

2001

Οι Fruend et al παρουσιάζουν το AR-PDA ένα concept για μία συσκευή παλάμης και ένα ασύρματο σύστημα που υποβοηθάει τους χρήστες με τεχνικές AR στις καθημερινές δραστηριότητές τους

Οι Vlahakis et al παρουσιάζουν το Archeoguide [28], ένα AR σύστημα για την αρχαία Ολυμπία. Το σύστημα εκτός από υπηρεσίες καθοδήγησης και εμφάνισης των 3D μοντέλων των ναών πάνω στα ερείπια, έχει τη δυνατότητα παρουσίασης των δραστηριοτήτων που

λάμβαναν μέρος στο χώρο, με avatars να επιδίδονται σε αθλήματα. Χρησιμοποιούνται υπολογιστές διαφόρων μεγεθών, από laptops έως handhelds.

Οι Bell et al επεκτείνουν τις δυνατότητες του Touring Machine με έναν mobile AR οδηγό εστιατορίων. Πάνω στην όψη του πραγματικού κόσμου επικάθονται πληροφορίες για κοντινά εστιατόρια, πληροφορίες για μενού, reviews κτλ (πρόσφατα παρουσιάστηκαν πολλές τέτοιες εφαρμογές για το ευρύ κοινό)

Οι Kooper και MacIntyre παρουσιάζουν τον πρώτο AR browser, ένα mobile AR σύστημα που λειτουργεί σαν interface για τη χρήση του διαδικτύου. (το 2008 με το wikitude υλοποιείται η ίδια ιδέα σε κινητό τηλέφωνο)

2002

Οι Kalkusch et al παρουσιάζουν ένα mobile AR σύστημα για την καθοδήγηση των χρηστών σε εσωτερικό χώρο, με τη χρήση του ARToolkit και tracking με αδρανειακούς αισθητήρες.

2003

Οι Wagner και Schmalstieg παρουσιάζουν ένα AR σύστημα καθοδήγησης σε εσωτερικό χώρο. Η όλη επεξεργασία γίνεται αποκλειστικά πάνω στο PDA (χωρίς οποιαδήποτε επεξεργασία σε κάποιον server), και με τη χρήση των ελάχιστων δυνατών υποδομών: ένα PDA με κάμερα και χρήση του ARToolkit σε πλατφόρμα windows mobile.

Οι Raskar et al παρουσιάζουν το πρώτο σύστημα παρουσίασης ψηφιακής πληροφορίας πάνω σε πραγματικά αντικείμενα με τη χρήση μικροσκοπικού φορητού προτζέκτορα. Η προβολή γίνεται επιτυχώς ανεξάρτητα από τη γεωμετρία του εκάστοτε αντικειμένου.

2004

Οι Möhring et al παρουσιάζουν ένα σύστημα για tracking ειδικών 3D marker σε κινητό τηλέφωνο.

Οι Rohs και Gfeller παρουσιάζουν το Visual Codes, ένα σύστημα από 2D markers για χρήση σε κινητά τηλέφωνα.

Παρουσιάζεται το Invisible Train, η πρώτη multi-user AR εφαρμογή για PDAs.

2006

Οι Reitmayr et al παρουσιάζουν ένα σύστημα για tracking σε αστικά περιβάλλοντα, με τη χρήση διάφορων αισθητήρων, επιτρέποντας έτσι ακριβή επικάθιση πληροφοριών σε πραγματικό χρόνο με τη χρήση tablet-PC.

Η Nokia παρουσιάζει το Mara, μία εφαρμογή καθοδήγησης για κινητό τηλέφωνο με χρήση διάφορων αισθητήρων [45].

2007

Παρουσιάζεται η πρώτη διαφημιστική AR εφαρμογή για κινητά τηλέφωνα, με χρήση 2D marker tracking.

Το ηλεκτρονικό περιοδικό Technology Review παρουσιάζει την AR σαν μία εκ των 10 σημαντικότερων αναδυόμενων τεχνολογιών του 2007 [14].

2008

Οι Wagner et al παρουσιάζουν την πρώτη εφαρμογή στην οποία γίνεται natural feature tracking σε πραγματικό χρόνο (με χρήση παραλλαγών των SIFT και Ferns), με επεξεργασία αποκλειστικά σε κινητό τηλέφωνο [46]

Η Mobilizy παρουσιάζει το Wikitude για τηλέφωνα με λειτουργικό Android. Η εφαρμογή αποτελεί έναν browser του πραγματικού κόσμου, με άντληση πληροφοριών από την Wikipedia, με τη χρήση δέκτη GPS και ψηφιακής πυξίδας.

2009

Η SPRXmobile παρουσιάζει το Layar, έναν εξελιγμένο browser του πραγματικού κόσμου, με περισσότερες δυνατότητες από το Wikitude, με τη χρήση GPS και πυξίδας. Γίνεται χρήση πληροφοριών από πολυάριθμα μέρη, όπως Wikipedia, Tweeter, Brightkite, αλλά και τοπικές υπηρεσίες για εντοπισμό καταστημάτων, στάσεων λεωφορείων, τουριστικών οδηγιών κτλ.

Η Mobilizy προτείνει τη δημιουργία της ARML μία mark-up language ανάλογη με την VRML²² που υιοθετήθηκε για την εικονική πραγματικότητα. Η πρότασή τους βασίζεται στην KML²³ (που χρησιμοποιείται σε GIS συστήματα) με ορισμένες επεκτάσεις [47].

Η Maes παρουσιάζει στο TED²⁴ ένα πρωτότυπο wearable AR συστήματος, με χρήση κάμερας, pico-projector, και αναγνώριση χειρονομιών, marker και εικόνας markers σαν είσοδο από το χρήστη [48]. Η απήχηση της παρουσίασης είναι ενδεικτική του ενδιαφέροντος του κοινού για την AR.

²² Virtual Reality Modeling Language

²³ Keyhole Markup Language

²⁴ <http://www.ted.com/>

2. Εισαγωγή στο Visual Tracking

Τα κυριότερα προβλήματα κατά την ανάπτυξη μιας εφαρμογής AR είναι το πού βρίσκεται ο χρήστης, πού εστιάζεται το ενδιαφέρον του, καθώς και πού πρέπει να παρουσιαστεί η επιθυμητή πληροφορία.

Το οπτικό tracking απαντάει τα παραπάνω ερωτήματα προσεγγίζοντας τον τρόπο που ο άνθρωπος εξερευνά με τις αισθήσεις του έναν άγνωστο χώρο (όραση). Έτσι, ενώ οι υπόλοιπες μέθοδοι απαιτούν συνήθως ειδικό τερματικό εξοπλισμό ή υποδομές, το visual tracking μπορεί να επιτευχθεί ακόμη και με off-the-shelf κινητά τηλέφωνα²⁵, χωρίς την ανάγκη extra εξοπλισμού.

Feature Detection

Προκειμένου να αξιοποιήσουμε την πληροφορία του κάθε frame από την κάμερα, καταφεύγουμε σε σύνθετες τεχνικές επεξεργασίας εικόνας, με πρώτο το στάδιο του feature detection [49]. Το γεγονός ότι είναι το πρώτο στάδιο το καθιστά, ίσως, και το σημαντικότερο, καθώς η απόδοσή του επηρεάζει την αποτελεσματικότητα όλων των υπόλοιπων και κατ' επέκταση την αποτελεσματικότητα του όλου αλγορίθμου.

Το feature detection έγκειται στην εύρεση σημείων ή περιοχών (ομάδων σημείων) ιδιαίτερου ενδιαφέροντος στην εικόνα, χωρίς όμως να υπάρχει σαφής ορισμός γι αυτό. Ο χαρακτήρας του αλλάζει κάθε φορά, ανάλογα με την περίπτωση στην οποία καλείται να εφαρμοστεί. Για την εύρεση των features εφαρμόζουμε κάποια μέθοδο η οποία επεξεργάζεται την εικόνα δημιουργώντας κάποια αφηρημένη πληροφορία από αυτή και τέλος εξετάζει αυτή την πληροφορία ώστε να ορίσει ποια pixels θα θεωρηθούν σημεία ενδιαφέροντος.

Γενικότερα, μπορούμε να ξεχωρίσουμε τους παρακάτω βασικούς τύπους feature detection²⁶ [49]:

- Edge detection (δίνει σαν αποτέλεσμα τα pixels των ακμών, τα “edgels” [50])
- Corner detection (δίνει σαν αποτέλεσμα τις γωνίες, σαν Point Of Interest)
- Blob detection (δίνει σαν αποτέλεσμα τα blobs, σαν Region Of Interest)
- Ridge detection (δίνει σαν αποτέλεσμα τα ridges)

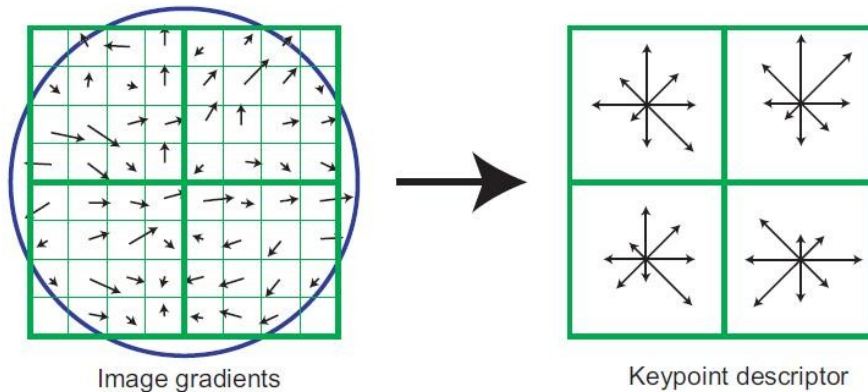
Σαν παραδείγματα απλών και δημοφιλών περιπτώσεων feature detector, μπορούμε να αναφέρουμε τις μεθόδους Canny edge detection [51], Sobel edge detection, Harris corner detection [52], FAST corner detection [53].

²⁵ Πλέον όλα τα κινητά τηλέφωνα είναι camera phones, ενώ πολύ σπάνια κυκλοφορεί κινητό τηλέφωνο χωρίς ενσωματωμένη κάμερα. Οι διάφοροι άλλοι αισθητήρες (δέκτης GPS, πυξίδα, επιταχυνσιόμετρα, κάρτες Wifi/Wimax) συναντώνται κυρίως στα smartphones χωρίς να αποτελούν ακόμη αναπόσπαστο standard σε όλα τα τηλέφωνα.

²⁶ Οπότε μιλάμε αντίστοιχά για edge/corner/blob/ridge detection

Feature Extraction

Το επόμενο βήμα μετά το Feature Detection είναι το Feature Extraction [49], το οποίο συνήθως επεξεργάζεται ιδιαίτερες περιοχές της εικόνας γύρω από σημεία ενδιαφέροντος, δίνοντας έτσι κάποιες πληροφορίες για αυτές, όπως πληροφορίες από το τοπικό ιστόγραμμα (local histogram), ή/και πληροφορίες που περιγράφουν το Feature της περιοχής (πχ το μέτρο της κλίσης της συνάρτησης της φωτεινότητας (intensity gradient) μιας ακμής). Το αποτέλεσμα του Feature extraction είναι τα Feature Vectors ή Feature Descriptors της εικόνας.



Εικόνα 12 - Keypoint Descriptor μέσω image gradients

Γενικότερα, μπορούμε να διακρίνουμε δύο κατηγορίες Visual Tracking, βασιζόμενες σε διαφορετικό τρόπο Feature Detection/Extraction, όπως παρουσιάζεται παρακάτω:

Natural Feature Tracking,

Η μέθοδος αυτή έγκειται στη χρήση εικόνων του “φυσικού” περιβάλλοντος και είναι ενδεχομένως από τις ιδανικότερες μεθόδους tracking για AR: πέρα από μία κάμερα (που πλέον περιλαμβάνονται στα camera phones) δε χρειάζεται κάποιου είδους επιπλέον εξοπλισμός (πχ RFID reader), ούτε παρουσιάζεται κάποιος “επεμβατισμός” στο περιβάλλον με την ανάπτυξη ειδικών υποδομών (πχ Wifi κεραίες κτλ) ή απλών βοηθητικών tags (πχ RFID tags, markers κτλ)

Μπορούμε να πούμε, λοιπόν, ότι απλά λαμβάνονται με την κάμερα τα “Visual bits”²⁷ του περιβάλλοντος, όμοια με τον τρόπο που ο άνθρωπος λαμβάνει την οπτική πληροφορία με τα μάτια του όταν εξερευνεί έναν άγνωστο χώρο.

Συνήθως, χρησιμοποιούνται ειδικοί Feature Descriptors, που ανταποκρίνονται στις απαιτήσεις των AR εφαρμογών. Ας αναφέρουμε για παράδειγμα τις απαιτήσεις ενός Outdoor Navigation συστήματος [54], που είναι ίσως από τα πιο χαρακτηριστικά και πιο απαιτητικά συστήματα ταυτόχρονα. Σε μία τέτοια εφαρμογή, είναι επιτακτικής ανάγκης η

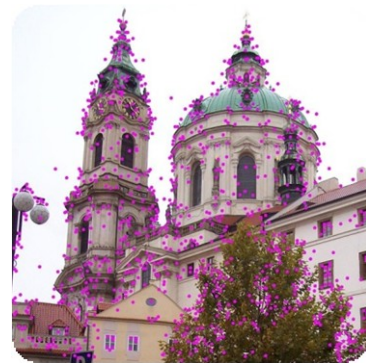
²⁷ Σε αντιδιαστολή με τα “Painted Bits”, όπως αυτά ορίζονται παρακάτω

αναγνώριση κτηρίων, μνημείων ή και τοποθεσιών. Στο εξωτερικό περιβάλλον όμως, οι συνθήκες είναι πολύ απρόβλεπτες (αναφέρονται μόνο μερικά ενδεικτικά χαρακτηριστικά ενός εξωτερικού αστικού περιβάλλοντος):

- Έχουμε συνεχή (μερική ή ολική) επικάλυψη των προς αναγνώριση κτιρίων από διερχόμενους πεζούς, αυτοκίνητα, δέντρα, κώλωνες ή εναέρια καλώδια.
- Οι συνθήκες φωτισμού μπορεί να μεταβάλλονται δραματικά, ακόμα και σε πολύ σύντομο χρονικό διάστημα, λόγω των καιρικών συνθηκών.
- Ο φωτισμός δεν είναι σχεδόν ποτέ ομοιόμορφος, λόγω αντανάκλασεων στις διάφορες επιφάνειες.
- Πρέπει να αναγνωρίζονται αποτελεσματικά ακόμα και κτίρια/τοποθεσίες με αρκετά όμοια όψη και χαρακτηριστικά.
- Η κάμερα του χρήστη μπορεί να βρίσκεται σε οποιαδήποτε απόσταση ή περιστροφή σε σχέση με τον προς αναγνώριση στόχο.

Από τα παραπάνω ενδεικτικά προβλήματα που πρέπει να αντιμετωπιστούν κατά την ανάπτυξη ενός Outdoor AR navigation συστήματος, συμπεραίνουμε ότι είναι αναγκαία η χρήση κάποιου ειδικού Feature Descriptor, με [55]

- όσο το δυνατόν μεγαλύτερη ανοσία
 - στο scaling,
 - στο rotation,
 - στο θόρυβο,
 - στις παραμορφώσεις,
 - στη μερική επικάλυψη,
 - στις αλλαγές του φωτισμού,
 - στην αλλαγή όψης (στον 3D χώρο).



Εικόνα 13 - Εύρεση Features σε εικόνα

- υψηλή διακριτοποίηση, με την έννοια του να παίρνουμε αρκετά διαφορετικά features για εικόνες αντικειμένων αρκετά όμοιες μεταξύ τους

Για τους παραπάνω λόγους, στις AR εφαρμογές με Natural Feature Tracking, είναι πολύ δημοφιλείς οι παρακάτω Feature Descriptors:

- SIFT [55] [56], ένα παράδειγμα εφαρμογής των οποίων φαίνεται στην εικόνα 13
- SURF [57] [58] [59]
- FERNS [60] [61]
- GLOH [62]
- LESH [63]
- DAISY [64] [65]

Μάλιστα, πρόσφατα έγινε δυνατή και η χρήση μερικών από αυτές τις μεθόδους σε κινητά τηλέφωνα, με αρκετά έντονη όμως τροποποίηση των αλγορίθμων, ώστε να γίνουν αρκετά γρήγοροι και να προσαρμοστούν στους περιορισμούς της υπολογιστικής ισχύος, της μνήμης και της ποιότητας του οπτικού συστήματος των κινητών τηλεφώνων.

Τα κυριότερα παραδείγματα είναι

- Η χρήση τροποποιήσεων των SIFT και FERNs από τους Wagner et al [46], σε real time tracking εφαρμογή, με frame rate έως 20 Hz. Μάλιστα αυτή είναι και η πρώτη υλοποίηση εφαρμογής πραγματικού χρόνου με natural feature tracking 6 βαθμών ελευθερίας σε κινητό τηλέφωνο.
- Η χρήση τροποποίησης του SURF από τους Takacs et al κατά τη σχεδίαση ενός Outdoor AR συστήματος για κινητά τηλέφωνα [54]. Η εφαρμογή τους, όμως, δεν είναι πραγματικού χρόνου, μιας και στόχευαν στην καλύτερη δυνατή αποτελεσματικότητα στην αναγνώριση των κτιρίων, παρά στο γρήγορο frame rate.
- Η ανάπτυξη του SURFTrac από τους TA et al [66], μία τροποποίηση του SURF για συνεχή αναγνώριση και tracking, φτάνοντας τα 6-7 fps.

Το natural feature tracking, τόσο στην πλατφόρμα των κινητών τηλεφώνων, όπως και γενικότερα, συγκεντρώνει έντονο ερευνητικό ενδιαφέρον. Παρόλα αυτά, εξαιτίας των μεγάλων απαιτήσεων σε υπολογιστικούς πόρους, παραμένει πολύ δύσκολη η εφαρμογή του σε πρακτικές εφαρμογές AR, ειδικά από τη στιγμή που ακόμα και τώρα χρησιμοποιείται αποκλειστικά και μόνο η CPU του κινητού τηλεφώνου για το υπολογιστικό φορτίο.

Μέχρι την αύξηση της υπολογιστικής ισχύος των κινητών²⁸ ή, φυσικά, σε περιπτώσεις που είναι εκ των πραγμάτων δεν ενδείκνυται η εφαρμογή της (βλ υποσημείωση/footnote 52), χρησιμοποιείται η εναλλακτική που παρουσιάζεται στην επόμενη ενότητα.



Εικόνα 14 - Αντιστοίχιση των features σε δύο εικόνες-όψεις του ίδιου αντικειμένου

²⁸ τόσο με την ανάπτυξη των CPU όσο κυρίως με την εμφάνιση πολυπύρηνων CPU ή τη δυνατότητα προγραμματισμού της GPU και των DSP (που σιγά σιγά αποκτάνε περισσότερα τηλέφωνα πέρα των high end) από 3rd party προγραμματιστές

Fiducial Marker Tracking

Σε αντίθεση με το Natural Feature Tracking, με τη χρήση του οποίου το περιβάλλον παραμένει αναλλοίωτο, το Marker Tracking έγκειται στην εκ των προτέρων παρέμβαση σε αυτό, και στην τοποθέτηση ειδικών “σημαδιών”, εύκολα ανιχνεύσιμων και αναγνωρίσιμων από τον υπολογιστή ή το κινητό τηλέφωνο.

Έτσι, ούτε με αυτό το είδος tracking υπάρχει η ανάγκη εξειδικευμένου εξοπλισμού (πέρα από έναν υπολογιστή με κάμερα ή ένα camera phone) ή ιδιαίτερων υποδομών (πχ Wifi κεραιών). Το αντίβαρο, όμως, του κέρδους σε υπολογιστικό φορτίο, είναι η παρέμβαση στο περιβάλλον με την τοποθέτηση markers και η οπτική μόλυνση που αυτά προκαλούν.

Εάν αυτή μπορεί να παραβλεφθεί²⁹, τα markers προσφέρουν επιπλέον πλεονεκτήματα, όπως πολύ αποτελεσματική αναγνώριση ακόμα και σε δύσκολες συνθήκες ή σε δύσκολα περιβάλλοντα (πχ διάδρομος ενός δημόσιου κτηρίου, όπου το natural feature tracking δεν είναι και τόσο αποτελεσματικό), πολύ μικρό κόστος ανάπτυξης, αφού ένας συνηθισμένος εκτυπωτής αρκεί για την εκτύπωση των markers, ενώ προσφέρουν δυνατότητα πολύ γρήγορου prototyping, ακόμα και για εφαρμογές που σχεδιάζονται για άλλου είδους tracking, αφού η δημιουργία και ανάρτηση των markers απαιτεί ελάχιστο χρόνο.

Εκτυπώνοντας ένα marker και αναρτώντας το σε ένα αντικείμενο, ουσιαστικά αποδίδεται σε αυτό μία μοναδική ταυτότητα. Και παρόλο που αυτή η ταυτότητα είναι “ηλεκτρονική”, αυτό επιτυγχάνεται με ένα απτό (“tangible”) αντικείμενο (marker), χωρίς κάποια χρήση ειδικού ηλεκτρονικού εξοπλισμού.

Έτσι, κατά την εκτύπωση ενός marker, ουσιαστικά έχουμε τη δημιουργία “Painted bits”³⁰, που περιέχουν τα bits της ηλεκτρονικής ταυτότητας ενός αντικειμένου, στο οποίο και αναρτώνται. Η ανάγνωσή των “painted bits” γίνεται με έναν υπολογιστή και μία κάμερα ή απλούστερα με ένα camera phone.

²⁹ Κάτι τέτοιο μπορεί να είναι αποδεκτό σε αρκετές περιπτώσεις, η χρήση του όμως, για παράδειγμα, σε ένα μουσείο αρχαίων μνημείων είναι εκ των πραγμάτων απαγορευτική

³⁰ Οι Ishii και Ullmer ονομάζουν [1] “tangible bits” τον τρόπο διαχείρισης της πληροφορίας μέσα από απτά μέσα, ενώ την απεικόνιση της εικονικής πληροφορίας στην οθόνη, την ονομάζουν “painted bits”. Στα πλαίσια της παρούσας εργασίας, με την εκτύπωση των bits της πληροφορίας στα marker, η έννοια “painted bits” αποκτάει πιο κυριολεκτική σημασία.

Έτσι, παρόλο που τα markers εμπίπτουν στην κατηγορία των “tangible bits”, στην παρούσα εργασία ορίζεται η παρακάτω ονομαστική σύμβαση:

- Τα **“painted bits”** ορίζουν τη χρήση markers για το visual tracking, και συνήθως “ζωγραφίζονται” σε απλό χαρτί με τη χρήση συνηθισμένου εκτυπωτή. Ο όρος “painted” τονίζει τον τεχνητό τρόπο αποθήκευσης ή και μετάδοσης της πληροφορίας.
- Τα **“visual bits”**, ορίζουν τη χρήση natural features για το tracking. Ο όρος “visual” καθορίζει (κυρίως σε αντιδιαστολή με τον όρο “painted”) τη χρήση φυσικής οπτικής πληροφορίας, όπως περίπου την εκλαμβάνει ο άνθρωπος όταν παρατηρεί το περιβάλλον με τα μάτια του.

Ουσιαστικά, το όλο σύστημα αποτελεί ένα τηλεπικοινωνιακό σύστημα. Η πληροφορία, δηλαδή οι ηλεκτρονικές ταυτότητες που δημιουργούνται στον υπολογιστή για τη σήμανση των διαφόρων αντικειμένων, κωδικοποιούνται και μετασχηματίζονται σε “painted bits”, αλλάζοντας έτσι μορφή, από ηλεκτρονική σε απτή. Κατόπιν αναρτώνται στα διάφορα αντικείμενα. Όταν ο χρήστης πλησιάζει ένα marker με ένα camera phone, το marker αποτελεί τον πομπό της πληροφορίας, που στέλνει τα “painted bits” στο δέκτη, δηλαδή στον οπτικό αισθητήρα του τηλεφώνου. Μέσω του αισθητήρα, λαμβάνονται τα “painted bits” και μετατρέπονται σε ψηφιακή πληροφορία, η οποία αποκωδικοποιείται και ανακτάται έτσι η αρχική ηλεκτρονική ταυτότητα.

Η ταυτότητα που αποδίδουν τα “painted bits” στα διάφορα αντικείμενα, αποτελούν ένα μέσο γεφύρωσης του πραγματικού και του εικονικού κόσμου, γεγονός μεγάλης σημασίας για την AR. Με την αποκωδικοποίηση των “painted bits” και την ανάκτηση της ταυτότητας ενός αντικειμένου, δίνεται η δυνατότητα στο χρήστη

- είτε άντλησης πληροφορίας σχετικά με αυτό, από κάποια βάση δεδομένων ή από το internet
- είτε αυτόματης έναρξης κάποιου service, όπως multimedia playback (βλ εικόνα 15) ή άνοιγμα κάποιου site στον browser
- είτε προσθήκης πληροφοριών σχετικά με αυτό το αντικείμενο, με upload κειμένου, φωνής ή εικόνας/video

Τα “painted bits” μετατρέπουν λοιπόν τα markers σε “rhicons”³¹ [1], “φέρνοντας” κατά κάποιο τρόπο το desktop στον πραγματικό κόσμο. Κάθε “σημαδεμένο” αντικείμενο αποτελεί και ένα “εικονίδιο”, ενώ η επιλογή του και η έναρξη των αντίστοιχων services γίνεται με τη βοήθεια της κάμερας και του software αποκωδικοποίησης, με μία “point ‘n shoot” λογική. Τα χέρια του χρήστη αποδεσμεύονται από το πληκτρολόγιο, και ο όλος χειρισμός γίνεται είτε με το ένα χέρι, στην περίπτωση των camera phones, είτε handsfree, στην περίπτωση του wearable computing.



Εικόνα 15 - Παράδειγμα marker-based AR εφαρμογής

³¹ PHYsical iCON

Visual Markers

Όπως ειπώθηκε προηγουμένως, στην προσπάθειά μας να γεφυρώσουμε τον πραγματικό με τον ψηφιακό κόσμο, μπορούμε να “ζωγραφίσουμε bits” πάνω σε αντικείμενα του πραγματικού κόσμου, με την μορφή των markers. Έτσι, δημιουργούμε ένα είδος “οπτικού link”³², διαδικασία γνωστή και ως object hyperlinking ή ως mobile tagging, επεκτείνοντας το internet πέρα από τον ηλεκτρονικό κόσμο, στον κόσμο των πραγμάτων και των πραγματικών τοποθεσιών³³.

Τα markers είναι λοιπόν ένας βολικός τρόπος διακριτοποίησης των αντικειμένων μεταξύ τους, με μικρό οικονομικό και υπολογιστικό κόστος και σχετικά μικρή παρεμβατικότητα στον περιβάλλοντα χώρο μας. “Τυπώνοντας” ένα marker σε ένα αντικείμενο, έχουμε πλέον μία μοναδική ταυτότητα γι’ αυτό, καθώς και δυνατότητα ευκολότερης ανάπτυξης πρακτικών εφαρμογών που εκμεταλλεύονται αυτή την ταυτοποίηση.

Ειδικότερα με τη διάδοση των camera phones, το τηλέφωνο μπορεί να αποτελέσει έναν κινητό browser, και να αλλάξει τον τρόπο που αλληλεπιδρούμε με τα αντικείμενα του περιβάλλοντός μας.



Εικόνα 16 - Παράδειγμα χρήση marker σε εφημερίδα για πρόσβαση σε ψηφιακή πληροφορία

Ο χρήστης μπορεί να αλληλεπιδράσει όχι μόνο με markers τυπωμένα στο χαρτί (πχ σε αφίσες ή εφημερίδες - βλ εικόνα 16), αλλά και με markers στην οθόνη του υπολογιστή, που ενσωματώνονται σε websites ή online videos³⁴ ή και σε ειδικές οθόνες (πχ σε διαφημιστικές βιτρίνες) που απεικονίζουν εναλλασσόμενα markers (έτσι τα markers γίνονται πλέον 3D, με την επιπλέον διάσταση του χρόνου, ή μπορούμε να θεωρήσουμε ότι αποστέλλεται πληροφορία πολυπλεγμένη στο χρόνο)

³² Εκτός από τα visual tags (markers) υπάρχουν και άλλοι τρόποι απόδοσης μίας ταυτότητας σε αντικείμενα ή τοποθεσίες [103]:

- Με RFID tags Τοποθετώντας παθητικά tags στα αντικείμενα και “διαβάζοντάς” τα με τη χρήση του κατάλληλου RFID reader, που συνδέεται με κινητό τηλέφωνο.
- Με SMS tags Σε κάθε αντικείμενο/τοποθεσία αναγράφεται ένας κωδικός, τον οποίο ο χρήστης στέλνει με SMS σε μία ειδική υπηρεσία, η οποία και αποστέλλει τις επιθυμητές πληροφορίες (πχ www.yellowarrow.net)
- Με Virtual tags Δε χρησιμοποιούνται πραγματικά tags, αλλά εικονικά, καθώς οι πληροφορίες συνδέονται με γεωγραφικές συντεταγμένες τοποθεσιών, ενώ γίνεται χρήση συστημάτων όπως το GPS, το Galileo ή και Wifi δίκτυα.
- Με Hardlink Σε κάθε αντικείμενο/τοποθεσία αναγράφεται ένας κωδικός, ο οποίος εισάγεται σε έναν mobile browser κινητού τηλεφώνου, στο οποίο και κατεβαίνουν οι αντίστοιχες ψηφιακές πληροφορίες.

³³ πχ www.semopedia.org

³⁴ Για παράδειγμα στο video clip των “Pet Shop Boys” για το τραγούδι “Integral”, στο οποίο ενσωματώθηκαν QR-Codes που παραπέμπουν σε websites για τα πολιτικά δικαιώματα.

Η εμφάνιση του mobile tagging έγινε με την κυκλοφορία του J-SH09 (το 2002) από τη Sharp και την J-Phone, ένα camera phone με ενσωματωμένο reader για το γιαπωνέζικο 1D barcode system και το 2D QR-Code system. Το τελευταίο, μάλιστα, εξελίχθηκε στην Ιαπωνία στο de facto standard για τις επιχειρήσεις κινητής τηλεφωνίας και διαφημιστικών υπηρεσιών. Επίσης, τα περισσότερα κινητά τηλέφωνα εκεί πωλούνται πλέον με προεγκατεστημένες εφαρμογές ανάγνωσης διαφόρων τύπων barcodes, και το mobile tagging τυγχάνει ιδιαίτερα μεγάλης κλίμακας εφαρμογής [67]. Έτσι, ο χρήστης, βλέποντας για παράδειγμα στο δρόμο την αφίσα μίας συναυλίας, βγάζει μία φωτογραφία το QR code της αφίσας, χρησιμοποιώντας το κινητό του, και μπαίνει αυτόματα στο site του διοργανωτή κλείνοντας εν κινήσει τα εισιτήριά του.

Οι πραγματικές δυνατότητες του mobile tagging είναι, όμως, μάλλον ανεξερεύνητες ακόμα, ενώ η διάδοση του εκτός Ιαπωνίας προχωράει με πιο αργούς ρυθμούς. Παρόλα αυτά το σκηνικό αλλάζει σιγά σιγά, καθώς εμφανίζονται περισσότερες εφαρμογές ανάγνωσης διάφορων barcodes για smartphones, ενώ κερδίζει έδαφος το virtual tagging για outdoor AR εφαρμογές σε smartphones [68].

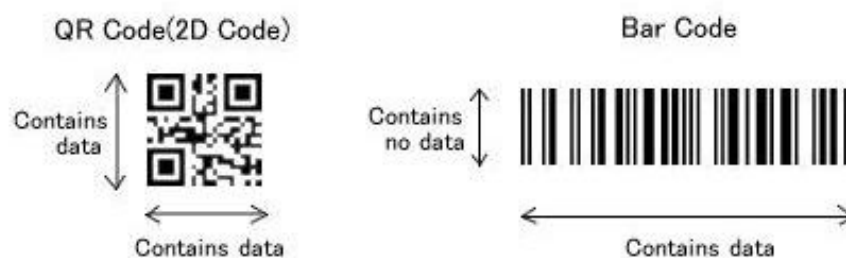
Είδη marker

Κατά καιρούς έχουν προταθεί διάφορα είδη marker, για διάφορες εφαρμογές.

Η πρώτη εισβολή των markers στην καθημερινότητά μας έγινε με το γνωστό 1D barcode που συνοδεύει κάθε προϊόν. Μάλιστα είναι το μόνο είδος marker που έχει επικρατήσει για μια συγκεκριμένη εφαρμογή (στη σήμανση προϊόντων στα super marker) σχεδόν παντού τόσα πολλά χρόνια.

Παρόλο που η διείσδυση του είναι μεγάλη, οι εφαρμογές που μπορεί να βρει είναι σχετικά περιορισμένες, μιας και:

- Για επιτυχή ανάγνωση το barcode πρέπει να βρίσκεται σε παράλληλο επίπεδο με το φακό της κάμερας, και σε γνωστό rotation
- Η κωδικοποίηση στα 1D barcodes γίνεται σε μία μόνο διεύθυνση, όπως φαίνεται και στην παρακάτω εικόνα



Εικόνα 17 - Σύγκριση 1D και 2D barcode συστημάτων

- Απαιτείται μεγάλη ανάλυση της κάμερας, λόγω των πυκνών γραμμών κωδικοποίησης και της κωδικοποίησης σε μία μόνο διάσταση

Έτσι, στρεφόμαστε στις λύσεις των δυσδιάστατων (2D) barcodes, τα οποία είναι και αυτά αρκετά διαδεδομένα, κυρίως για ειδικές εφαρμογές (πχ ταυτοποίηση προϊόντων, marketing κτλ).

Αυτά, μπορούμε να πούμε ότι χωρίζονται σε δύο κατηγορίες [67]:

- **Στα database 2D barcodes**

Η λειτουργία αυτών παρουσιάζεται στην εικόνα 18 (a) και μπορεί να χαρακτηριστεί και “offline”. Η ιδιότητά τους να απεικονίζουν μεγάλο αριθμό bits³⁵, καθιστά ικανή την κωδικοποίηση όχι μόνο αριθμών, αλλά και αλφαριθμητικών χαρακτήρων, είτε με τη μορφή κειμένου είτε με τη μορφή URL³⁶. Έτσι, δεν υπάρχει η ανάγκη για μία κεντρική βάση δεδομένων σε κάποιο server που θα έχει τον απόλυτο ρόλο στο σύστημα, είτε στέλνοντας περιεχόμενο είτε κάνοντας τη δρομολόγηση σε εξωτερικά services. Κατά κάποιο τρόπο, η βάση δεδομένων της εφαρμογής είναι (έστω μερικώς) κατανομημένη στον πραγματικό κόσμο μέσω των “painted bits”. Με αυτά ο χρήστης μπορεί να λάβει πληροφορίες στο κινητό του οπουδήποτε, οποτεδήποτε, και ανεξάρτητα από τη συνδεσιμότητα της συσκευής στο δίκτυο κινητής ή στο internet.

- **Στα index-based 2D barcodes**

Η λειτουργία αυτών παρουσιάζεται στην εικόνα 18 (b) και μπορεί να χαρακτηριστεί και “online”. Συνήθως απεικονίζουν αρκετά μικρότερο αριθμό bits από τα database (offline) barcodes, λειτουργώντας, έτσι, απλά ως κωδικοποίηση ενός μοναδικού αριθμού “ταυτότητας”, που παίζει το ρόλο του κλειδιού για την πρόσβαση στην κεντρική βάση δεδομένων της εφαρμογής. Αυτή είτε δίνει τα κατάλληλα δεδομένα στο χρήστη είτε βοηθάει στο routing προς εξωτερικά services.

Έτσι, προϋπόθεση για τη χρήση των index-based 2D barcodes είναι είτε κάποια database προαποθηκευμένη στο τηλέφωνο, είτε, συνηθέστερα, η συνδεσιμότητα του κινητού με οποιονδήποτε τρόπο σε κάποιον server³⁷. Η λειτουργία τους μπορεί

³⁵ Παρακάτω παρουσιάζεται ως παράδειγμα η μέγιστη χωρητικότητα για δύο συστήματα marker (από <http://www.denso-wave.com/qrcode/aboutqr-e.html>)

	Numeric	Alphanumeric	Binary	Kanji
QR Code	7,089	4,296	2,953	1,817
DataMatrix	3,116	2,355	1,556	778

³⁶ Για κωδικοποίηση πολύ μεγάλων URL μπορεί να συνδυαστεί η χρήση των marker με αυτή του tinyURL, εφόσον υπάρχει δυνατότητα σύνδεσης στο internet, με την απώλεια φυσικά της μη εξάρτησης από κάποιο server.

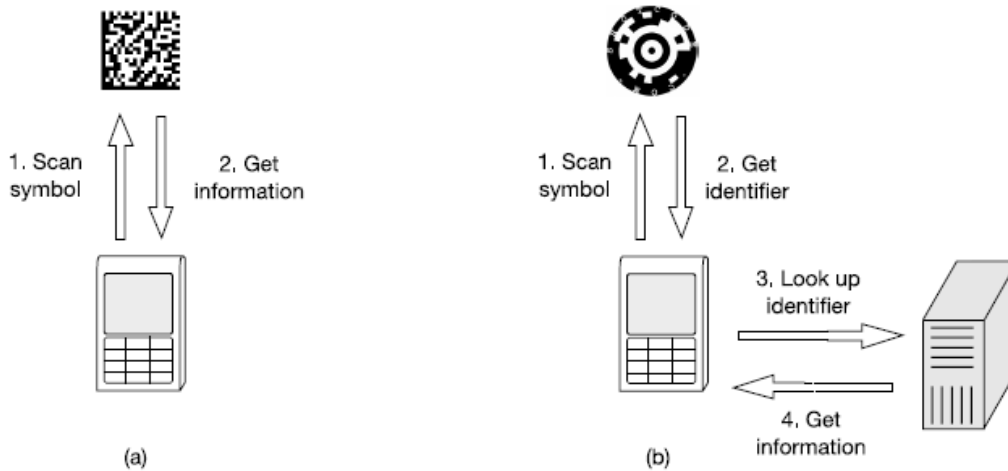
³⁷ Με τη χρήση αυτών των marker, υιοθετείται σχεδόν αυτόματα το σύστημα client-server για την εφαρμογή, γεγονός που απαιτεί τη συνδεσιμότητα της συσκευής.

Η συνδεσιμότητα του client μπορεί, όμως, να μην εξασφαλίζεται 100% ανά πάσα χρονική στιγμή. Παράλληλα, ο χρήστης μπορεί να κάνει συχνά ερωτήματα στη βάση δεδομένων του server, με αποτέλεσμα η εφαρμογή να επηρεάζεται αρνητικά από το χρόνο απόκρισης του συστήματος client-server.

Τα παραπάνω μπορούν να λυθούν με clustering των δεδομένων της database σε μικρές γεωγραφικές περιοχές (“Ioxels” [54]). Έτσι, ο server στέλνει στον client ένα κομμάτι της database (ανάλογα με το πώς αυτός κινείται στο χώρο ή κάθε φορά που αλλάζει το περιεχόμενο του συγκεκριμένου τμήματος της DB), που του εξασφαλίζει την αποτελεσματική λειτουργία της εφαρμογής και την αυτονομία σε αυτό το “Ioxel”.

Με αυτό τον τρόπο επιτυγχάνεται τόσο η δραστική μείωση της απόκρισης του συστήματος, όσο και η ανοσοποίησή του απέναντι στη διάλειψη της επικοινωνίας με το server. [54] [104]

να παρομοιωθεί με το γνωστό tinyURL³⁸, ειδικά στην περίπτωση του client-server μοντέλου).



Εικόνα 18 - Σύγκριση “database” και “index-based” 2D marker συστημάτων

Θα πρέπει να τονιστεί ότι, σε αντίθεση με τα database 2D barcodes, τα index-based 2D barcodes αναπτύχθηκαν ειδικά για τη χρήση με κινητό τηλέφωνο, με τους περιορισμούς της built in κάμερας που έχουν (χειρότερης ποιότητας οπτικά, μεγάλη βαρελοειδή παραμόρφωση, υψηλή χρωματική απόκλιση, και χαμηλή ανάλυση), καθώς και τους περιορισμένους υπολογιστικούς πόρους.

Παραδείγματα μερικών 2D Barcode συστημάτων παρατίθενται στον παρακάτω:

database 2D barcodes	index-based 2D barcodes
QR Code ³⁹	Visual Code ⁴⁰ [69]
VSCode	ShotCode ⁴¹
Data Matrix ⁴² (Semacode ⁴³)	ColorCode
MaxiCode ^{44 45}	
High Capacity Color Barcode	

Πίνακας 1 - Παραδείγματα μερικών 2D Barcode συστημάτων

³⁸ Στην database του server της εφαρμογής (<http://tinyurl.com>), στην κάθε εγγραφή γίνεται αντιστοίχιση ενός μακροσκελούς URL σε ένα μικρότερο (“tinyURL”). Όταν στον browser εισαχθεί το tinyURL, ο server της εφαρμογής βρίσκει στην database το μακροσκελές URL που του αντιστοιχεί, και κάνει αναδρομολόγηση προς αυτό [105]

³⁹ <http://www.denso-wave.com/qrcode/qstandard-e.html>

⁴⁰ Πέρα από τη λειτουργία σαν marker system όπως παρουσιάζεται στο κείμενο, οι δημιουργοί του στοχεύουν σε ένα νέο mobile HCI σύστημα, με συνεχές tracking του marker και “ανάγνωση” των κινήσεων του τηλεφώνου ως προς το marker (με movement-detection μέσω της κάμερας), το οποίο λειτουργεί έτσι σαν “οπτικό ποντίκι”

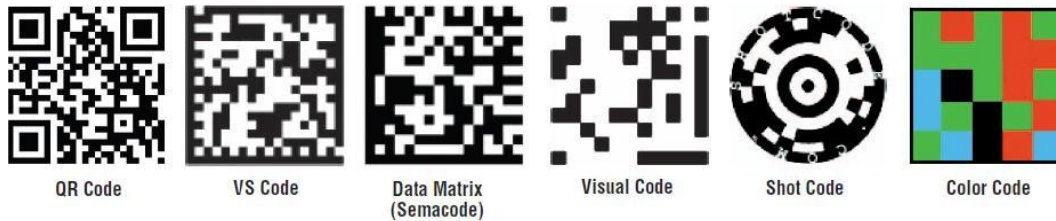
⁴¹ <http://www.shotcode.com/>

⁴² ISO/IEC 16022:2006, 15415, 15418:2009, 15424:2008, 15434:2009, 15459

⁴³ <http://semacode.com/>

⁴⁴ ISO/IEC 16023:2000

⁴⁵ Χρησιμοποιείται από το [United Parcel Service](http://www.ups.com) (UPS)



Εικόνα 19 - Μερικά παραδείγματα διάφορων 2D barcode συστημάτων

Αναλυτική παρουσίαση και σύγκριση των marker systems των τριών πρώτων σειρών του πίνακα 34, μπορεί να βρει κανείς στην εργασία των Kato και Tan [67].

Γενικότερα, όμως, μπορούμε να πούμε ότι τα συστήματα αυτά έχουν διάφορα χαρακτηριστικά, όπως

- Υψηλή αντίθεση για αποτελεσματικότερη ανάγνωση (όλα είναι bitonal, με εξαίρεση το ColorCode και το High Capacity Color Barcode)
- Ανοσία στη rotation της κάμερας
- Ανοσία στο θόρυβο με τη προσθήκη error correction codes κατά την κωδικοποίηση του marker (ωφελεί και στην αύξηση της αποτελεσματικής της ανάγνωσης από μεγαλύτερη απόσταση)
- Αποτελεσματική ανάγνωση με απόσταση κάμερας-marker περίπου από 5cm έως 25 cm κατά μέσο όρο

Παρόλα τα παραπάνω πλεονεκτήματα, παρατηρούνται στο κάθε marker system και (τουλάχιστο ένα από) τα παρακάτω μειονεκτήματα:

- Πρέπει να βρίσκονται σε επίπεδο σχετικά παράλληλο με το φακό της κάμερας (αν και δεν υπάρχει περιορισμός ως προς τη σχετική περιστροφή)
- Για επιτυχή αναγνώριση πρέπει να καταλαμβάνουν μεγάλο μέρος της εικόνας, άρα έχουμε μικρό αποδεκτό εύρος απόστασης της κάμερας από το marker
- Δεν μπορούμε να εξάγουμε 6 βαθμούς ελευθερίας (6 DoF)⁴⁶
(με εξαίρεση το ColorCode)

Όπως θα φανεί στην επόμενη ενότητα, αυτά τα μειονεκτήματα αντιβαίνουν στις απαιτήσεις των marker-based AR εφαρμογών. Σαν αποτέλεσμα, δημιουργείται η ανάγκη για ένα νέο marker system στα πλαίσια της AR.

⁴⁶ Όπως θα δειχτεί παρακάτω, για tracking 6 βαθμών ελευθερίας χρειάζονται 4 σημεία σε κάθε marker, που συνήθως δίνονται μέσω ενός σταθερού μαύρου περιγράμματος

Οι επιθυμητές ιδιότητες ενός AR marker system

Ένα ιδανικό marker system για Augmented Reality, απαιτείται να έχει τις παρακάτω ιδιότητες:

- Θα πρέπει να γίνεται ανάγνωση των markers ακόμα και υπό μη ελεγχόμενες συνθήκες⁴⁷. Έτσι, είναι ιδανικότερη η χρήση bitonal markers για υψηλή αντίθεση, και η αποφυγή χρήσης χρώματος, μιας και οι χρωματικές αποκλίσεις είναι μεγάλες από συσκευή σε συσκευή, ενώ και η επίδραση των φωτιστικών συνθηκών επηρεάζει έντονα το αποτέλεσμα.
- Θα πρέπει να γίνεται ανάγνωση του marker από σχετικά μεγάλες αποστάσεις (έως και μερικά μέτρα) και από (σχετικά) ευρυγώνιους φακούς. Αν και αυτό ρυθμίζεται εν μέρει και από το μέγεθος των marker, πρέπει να χρησιμοποιούνται marker με κωδικοποίηση λίγων bits, ώστε να μην περιέχουν στην επιφάνειά τους πυκνή πληροφορία που τα κάνει δυσανάγνωστα. Έτσι, υιοθετούνται τα index-based 2D barcodes, σε συνδυασμό με κάποιο error correction code⁴⁸, ώστε να αυξηθεί η αποτελεσματικότητα και η αξιοπιστία της ανάγνωσης από απόσταση.
- Θα πρέπει να υπάρχει η δυνατότητα ανίχνευσης πολλών markers σε ένα frame, που μπορεί να καταλαμβάνουν μικρό χώρο σε αυτό. Έτσι, οι μέθοδοι ανάγνωσης πρέπει να προσαρμόζονται σε αυτή την απαίτηση, και να μην απαιτούν (για ευκολία) να καλύπτεται το μεγαλύτερο μέρος του frame από ένα και μοναδικό ενός μοναδικού marker κάθε φορά (όπως στα marker systems που παρουσιάστηκαν παραπάνω).
- Θα πρέπει να μπορούμε να διεξάγουμε με αυτό pose estimation της κάμερας 6 βαθμούς ελευθερίας, για να είναι εφικτή η εισαγωγή του εικονικού στοιχείου πάνω στο πραγματικό. Αυτός ο περιορισμός μας ωθεί στη χρήση τετράγωνων markers.

Ενδιαφέροντα συμπεράσματα μπορούμε να εξάγουμε και από της εργασία των Kato και Tan⁴⁹ [67]. Όπως αναφέρουν:

- Κατά μέσο όρο τα index-based 2D Barcodes είχαν στις δοκιμές τους καλύτερη απόδοση από τα database barcodes
- Στα markers με μεγαλύτερο μέγεθος cell, δηλαδή στα marker systems στα οποία έχουμε
 - μικρότερη πυκνότητα δεδομένων στην επιφάνεια του marker, και άρα
 - μικρότερο πλήθος διαφορετικών συνδυασμών-marker παρατηρήθηκε
 - πιο αξιόπιστη ανάγνωση και

⁴⁷ έντονες μεταβολές του φωτισμού, ανομοιόμορφος φωτισμός της επιφάνειας του marker λόγω σκιών ή ανακλάσεων φωτός, ανάγνωση των marker υπό μεγάλη γωνία ή υπό μερική επικάλυψη κτλ

⁴⁸ πχ crc, bch, reed-solomon κτλ

⁴⁹ Αν και η εργασία τους δεν αφορούσε εφαρμογές σε AR, αλλά γενικά σε pervasive εφαρμογές, τα συμπεράσματα μπορούν να επεκταθούν με ασφάλεια

- από μεγαλύτερη απόσταση
- Η ανάλυση της κάμερας δεν παίζει κυρίαρχο ρόλο στην απόδοση του όλου συστήματος, σε αντίθεση με άλλα χαρακτηριστικά, όπως η ποιότητα των οπτικών, το autofocus, η ευαισθησία στον περιβάλλοντα φωτισμό
- Με διπλασιασμό του μεγέθους του marker, διπλασιάζεται περίπου και η απόσταση ασφαλούς επιτυχούς ανάγνωσης⁵⁰.

Επίσης, συμπεραίνουν ότι υπάρχουν τρεις κύριοι τρόποι βελτίωσης της αξιοπιστίας ενός marker system:

- Το μέγεθος των cells
- Οι μέθοδοι αποκωδικοποίησης
- Το hardware του reader

Συνοψίζοντας τα παραπάνω για ευκολότερη αναφορά, ένα marker system ιδανικό για AR εφαρμογές, θα πρέπει:

- Να περιβάλλεται από τετράγωνο περίγραμμα
- Να είναι bitonal
- Να είναι index-based 2D barcode
- Να έχει όσο το δυνατόν μικρότερη πυκνότητα “οπτικών bits” (μικρότερο μέγεθος cell), με το ανάλογο trade off για το διαφορετικό πλήθος διαφορετικών markers⁵¹
- Να επιτρέπει την pose estimation 6 βαθμών ελευθερίας
- Να περιέχει error correction
- Να ανιχνεύεται από μακρινή απόσταση
- Το μέγεθος του marker να καθορίζεται από το μέγεθος της απόστασης κάμερας-marker, στην οποία απαιτείται αξιόπιστη ανάγνωση και αποκωδικοποίηση.

⁵⁰ Βέβαια, αυτό παρατηρείται για αποστάσεις της τάξης των μερικών cm (βλ σελ 34). Σε ένα AR σύστημα όπου έχουμε αποστάσεις της τάξης των μερικών μέτρων, η αύξηση του μεγέθους αναμένεται να αυξάνει σίγουρα την απόσταση επιτυχούς ανάγνωσης, όχι όμως σε αυτό το ποσοστό (διπλασιασμό)

⁵¹ Το trade off διαφέρει για κάθε εφαρμογή. Σε ένα indoor navigation AR σύστημα, για παράδειγμα, θέλουμε πολύ μεγάλο πλήθος διαφορετικών markers (χρήση σε κάθε πόρτα, στους τοίχους, στους διαδρόμους κτλ). Σε κάθε όροφο όμως, μπορούν να ανακυκλώνονται τα ίδια markers, με τη βοήθεια κάποιου tag που πληροφορεί το σύστημα για τη μετάβαση σε διαφορετικό όροφο. Έτσι διαμορφώνονται ανάλογα και τα (κλειδιά στα) ερωτήματα του client προς το server.

3. Βιβλιογραφική επισκόπηση

Το πρώτο 2D barcode σύστημα που πληρούσε τις παραπάνω προϋποθέσεις για την AR ήταν το Matrix [70], που παρουσιάστηκε το 1996 από τον Rekimoto [history], και αργότερα (2000) εξελίχθηκε στο Cybercode [71], που στηρίχτηκε από τη Sony.

Το γεγονός αυτό, μαζί με την κυκλοφορία του J-SH09, συνετέλεσε στην εξέλιξη και εμπορική χρήση των pervasive 2D barcodes (γενικότερα, ανεξάρτητα από τα AR συστήματα) που παρουσιάστηκαν παραπάνω.

Μετά τα συστήματα του Rekimoto, η marker-based AR συγκέντρωσε το ενδιαφέρον μεγάλου μέρους της ερευνητικής AR κοινότητας, καθώς οι περιορισμοί της εποχής στην υπολογιστική ισχύ έκαναν αδύνατο το natural feature tracking⁵², ενώ αργότερα το ενδιαφέρον αναβίωσε με την εφαρμογή της marker-based AR στα κινητά τηλέφωνα.

Έτσι, παρουσιάστηκαν αρκετά συστήματα marker για AR εφαρμογές, στην ενότητα όμως που ακολουθεί θα παρουσιαστούν οι κυριότεροι αντιπρόσωποι των “τάσεων” που παρατηρήθηκαν, και οι οποίοι κυριαρχούν στις αναφορές στη σχετική βιβλιογραφία.

Αν και παρουσιάζονται ορισμένα πλεονεκτήματα και μειονεκτήματα κάθε μεθόδου, ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στις εργασίες των Zhang et al [72] και Fiala [73] για πληρέστερη συγκριτική μελέτη.

⁵² Αυτό, όμως, δεν ήταν φυσικά ο μοναδικός λόγος εξέλιξης της marker-based AR, αλλά έδωσε μία περαιτέρω αρχική ώθηση.

Υπάρχουν εφαρμογές όπου το natural feature tracking δεν ενδείκνυται, όπως σε ένα indoor tracking AR σύστημα, καθώς οι διάδρομοι ενός δημόσιου κτηρίου, για παράδειγμα, είναι πανομοιότυποι και το natural feature tracking είτε θα αποτυγχάνει είτε δε θα δίνει αποδεκτά αποτελέσματα, ειδικά αν εφαρμοστεί σαν μοναδική μέθοδος και όχι σε συνδυασμό με κάποια άλλη ενδεικτικότερη για αυτή την εφαρμογή.

Ένα marker-based σύστημα (ή κάποιο σύστημα διαφορετικού tagging) θα ήταν αποτελεσματικότερο για μία τέτοια εφαρμογή, με το trade-off φυσικά της οπτικής μόλυνσης. (αυτή θα μπορούσε να μειωθεί είτε με το συνδυασμό μεθόδων tagging, είτε με χρήση designable markers [84]).

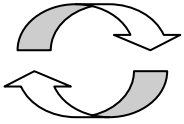
Με ανάλογη οπτική παρέμβαση (και όχι μόλυνση) θα μπορούσε ίσως να βελτιωθεί και η απόδοση του natural feature tracking, με τη χρήση πινάκων/αφισών/graffiti στο κτίριο, χωρίς όμως να φτάνει την αποτελεσματικότητα του marker tracking.

ARToolkit

Το ARToolkit παρουσιάστηκε από τους Kato και Billinghurst το 1999 [44], ενώ ο κώδικάς του δόθηκε στη διαθεσιμότητα του κοινού [74] σαν δωρεάν λογισμικό ανοιχτού κώδικα, με άδεια GNU GPL⁵³ (για μη εμπορικούς σκοπούς). Αυτός ήταν και ο λόγος που έγινε γρήγορα δημοφιλές⁵⁴ ανάμεσα στους ενδιαφερόμενους στους τομείς της AR και του Human Computer Interaction, οι οποίοι είχαν πλέον τη δυνατότητα να αναπτύξουν γρήγορα εφαρμογές με ένα έτοιμο εργαλείο λογισμικού, χωρίς να ασχοληθούν με το χρονοβόρο κομμάτι της επεξεργασίας εικόνας.

Το ARToolkit κάνει χρήση marker σε άσπρο χαρτί και με μαύρο τετράγωνο περίγραμμα. Στο εσωτερικό του μπορεί να τοποθετηθεί οποιοδήποτε ασπρόμαυρο (διαβαθμίσεις του γκρι) pattern είτε αυτό είναι καταληπτό από τον άνθρωπο είτε όχι (βλ εικόνα 20).

Στον παρακάτω πίνακα φαίνονται τα διάφορα στάδια της λειτουργίας του ARToolkit [74]

Αρχικοποίηση	1. Έναρξη της λειτουργίας της κάμερας, ανάγνωση των αρχείων με τα προαποθηκευμένα marker patterns και των παραμέτρων της κάμερας
Κυρίως loop 	2. Ανάγνωση του τρέχοντος video frame
	3. Εντοπισμός των marker και αναγνώριση των pattern μέσα στο frame
	4. Υπολογισμός της θέσης της κάμερας σε σχέση με τα αναγνωρισμένα markers
	5. Εμφάνιση εικονικών αντικειμένων πάνω στα αναγνωρισμένα markers
Τερματισμός	6. Τερματισμός της λειτουργίας της κάμερας

Πίνακας 2 - Τα βασικά βήματα της λειτουργίας του ARToolkit

Όπως φαίνεται στο στάδιο της αρχικοποίησης, προϋποθέτεται η εκ των προτέρων calibration της κάμερας, ενώ βρίσκονται αποθηκευμένα σε ένα αρχείο όλα τα marker patterns που χρησιμοποιούνται (το γιατί θα γίνει κατανοητό παρακάτω)



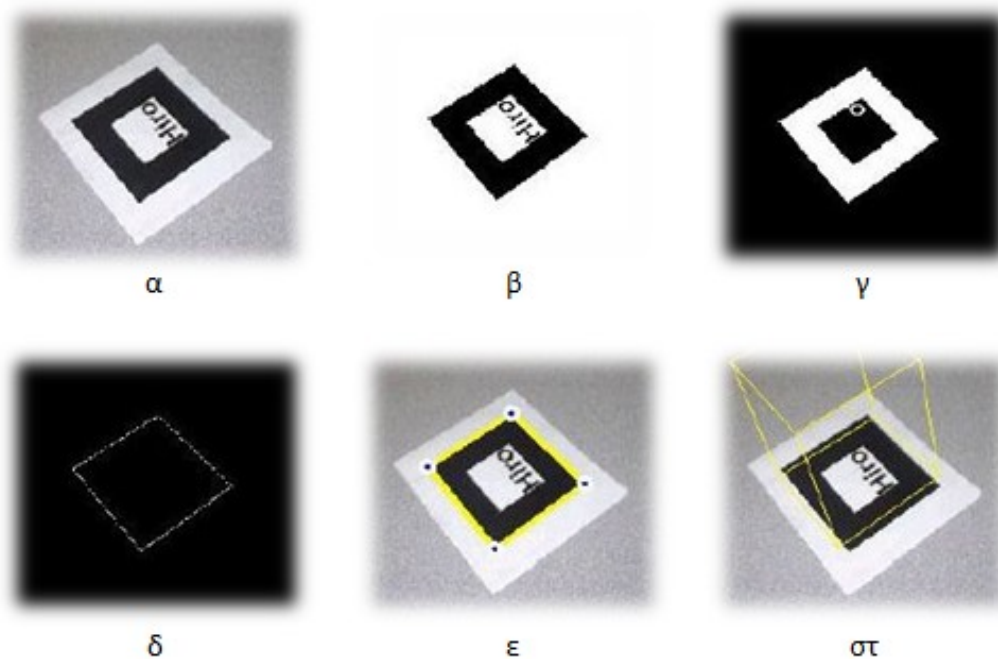
Εικόνα 20 - Μερικά παραδείγματα markers του ARToolkit

Στην επόμενη σελίδα απεικονίζονται τα διάφορα επαναλαμβανόμενα στάδια στο main loop της εφαρμογής. Αρχικά **(α)** διαβάζεται το τρέχον frame από την κάμερα. Έπειτα **(β)**, με ένα κατώφλι, το frame μετατρέπεται σε δυαδική εικόνα, εκμεταλλευόμενο το γεγονός ότι το περίγραμμα των markers παρουσιάζει έντονη αντίθεση (μαύρο “κάδρο” σε άσπρο χαρτί). Στη δυαδική αυτή εικόνα αναζητούνται ομάδες “ενωμένων” pixels που ορίζουν διαφορετικές περιοχές στο marker **(γ)** (περιοχές είτε με πολύ μικρό είτε με πολύ μεγάλο μέγεθος απορρίπτονται αμέσως), και με βάση αυτές εξάγεται το περίγραμμα του marker **(δ)**. Έτσι, μπορούν πλέον να βρεθούν οι 4 ακμές του marker, καθώς και οι 4 γωνίες του **(ε)**.

⁵³ <http://www.gnu.org/licenses/gpl-3.0.txt>

⁵⁴ Σήμερα είναι πολύ δημοφιλές και για διαδικτυακές εφαρμογές AR, όντας η μόνη υλοποίηση AR μεθόδων με Flash, με την πρόσφατη εμφάνιση του FLARToolkit [110]

Οι 4 αυτές γωνίες χρησιμοποιούνται μετέπειτα για την αναγνώριση της ταυτότητας του marker, την εύρεση της θέσης του ως προς την κάμερα, και τελικά την εμφάνιση εικονικών αντικειμένων πάνω στο marker, σαν να ήταν προσκολλημένα πάνω σε αυτό (στ).



Εικόνα 21 - Απεικόνιση των βασικών βημάτων της λειτουργίας του ARToolkit

Πιο συγκεκριμένα, αφού βρεθούν οι 4 γωνίες του marker, χρησιμοποιούνται για τον υπολογισμό της homography και την αφαίρεση της “προοπτικής παραμόρφωσης” εξαιτίας της σχετικής θέσης της κάμερας ως προς το marker. Φέρνοντας με αυτό τον τρόπο το marker σε “κανονική” θέση (canonical view), δειγματοληπτούμε το εσωτερικό του, που αποτελείται από διάφορες διαβαθμίσεις του γκρι, με ένα πλέγμα $N \times N$ από scanlines⁵⁵. Δημιουργείται έτσι το αντίστοιχο διάνυσμα (feature vector - 256 στοιχεία για $N=16$, ενώ 1024 για $N=32$), το οποίο θα συγκριθεί με όλα τα προαποθηκευμένα feature vectors, για κάθε marker που χρησιμοποιεί η εφαρμογή. Μάλιστα σε κάθε marker αντιστοιχούν 12 (εξηγείται παρακάτω) προαποθηκευμένα feature vectors, ένα για κάθε πιθανή περιστροφή. Αποτέλεσμα της σύγκρισης των feature vectors με correlation είναι η εξαγωγή ενός συντελεστή εμπιστοσύνης (confidence factor). Ανάλογα με την τιμή αυτού σε σχέση με ένα προκαθορισμένο κατώφλι, έχουμε την τελική αναγνώριση ή μη του marker.

Πλεονεκτήματα

Όπως φαίνεται, η λειτουργία του συστήματος είναι αρκετά απλή, με αποτέλεσμα αυτό να είναι ιδανικό για εφαρμογές πραγματικού χρόνου.

Πέρα από αυτό η λειτουργία του προϋποθέτει, από πλευράς εξοπλισμού, εκτός από τον υπολογιστή και την κάμερα μόνο τη δημιουργία των markers, κάτι που μπορεί να γίνει πολύ εύκολα με έναν συνηθισμένο εκτυπωτή.

Τέλος, κατά τη δημιουργία ενός marker έχουμε τη δυνατότητα να τοποθετήσουμε στο εσωτερικό του οποιοδήποτε ασπρόμαυρο σχήμα ή εικόνα (διαβαθμίσεις του γκρι),

⁵⁵ στην κανονική έκδοση είναι $N=16$, αλλά πολλοί χρησιμοποιούν $N=32$

“αναγνώσιμη” από τον άνθρωπο, κάτι που καθιστά το σύστημα χρήσιμο και για διαφημιστικές εφαρμογές, με τη χρήση του λογότυπου κάποιας εταιρείας ή προϊόντος, ενώ κατά κάποιο τρόπο μειώνεται και η οπτική μόλυνση που προκαλούν τα markers στο χώρο της εφαρμογής.

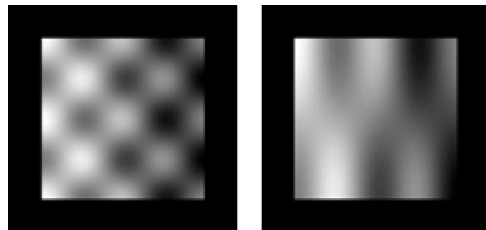
Μειονεκτήματα

Η απλή λειτουργία του ARToolkit μπορεί να έχει πολλά πλεονεκτήματα, έχει όμως και αρκετά σοβαρά μειονεκτήματα [75] [35].

Το γεγονός ότι γίνεται έντονη χρήση προκαθορισμένων κατωφλίων, καθιστά το σύστημα χρήσιμο κυρίως υπό συγκεκριμένες ελεγχόμενες συνθήκες. Υπό μη ελεγχόμενες συνθήκες όμως, όπως για παράδειγμα με μεταβολές του φωτισμού, η μέθοδος αποτυγχάνει εύκολα. Επίσης, στην περίπτωση ύπαρξης πολλών marker σε ένα frame υπό ανομοιόμορφο φωτισμό, η αναγνώριση μπορεί είναι επιτυχής μόνο για μερικά από αυτά (βλ εικόνα 25). Κι αυτό γιατί υπό ανομοιόμορφο φωτισμό ή λόγω αντανάκλασεων παρατηρείται συχνά το φαινόμενο ένα “άσπρο” pixel στη μία μεριά του marker να είναι πιο χαμηλότερη τιμή φωτεινότητας από ένα “μαύρο” pixel στην άλλη μεριά. Για να ξεπεραστεί αυτό το εμπόδιο χρησιμοποιείται από πολλούς η αντικατάσταση των σταθερών (fixed) με τοπικά ή προσαρμόσιμα (local/adaptive) κατώφλια (όπως το ARToolkitplus που παρουσιάζεται στην επόμενη παράγραφο).

Εξαιτίας της μεθόδου μετατροπής του κάθε frame σε δυαδική εικόνα, και της εύρεσης του περιγράμματος των marker με ανίχνευση ομάδων ενωμένων pixels, η μέθοδος δεν έχει ανοχή απέναντι στην μερική επικάλυψη του marker. Έτσι, η παραμικρή επικάλυψη έστω και ενός μικρού μέρους του περιγράμματος, οδηγεί την ανίχνευση του marker σε αποτυχία.

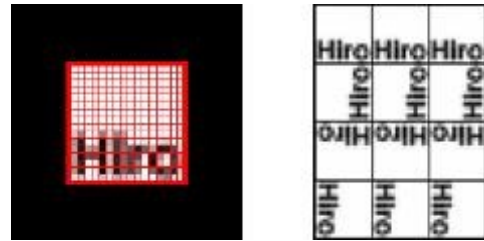
Επίσης, η σύγκριση των προαποθηκευμένων feature vectors με αυτά των αναγνώσιμων markers γίνεται με correlation, που δεν δίνει αρκετά αξιόπιστα αποτελέσματα. Έτσι, είναι δυνατόν δύο φαινομενικά διαφορετικά markers να έχουν σχεδόν όμοια feature vectors, κάτι που τα κάνει δυσδιάκριτα μεταξύ τους. Σαν αποτέλεσμα είτε αποτυγχάνει η αναγνώριση των markers είτε γίνεται λανθασμένη ταυτοποίησή τους (intermarker confusion rate). Μία μέθοδος αντιμετώπισης του προβλήματος αυτού αναπτύχθηκε από τους Xiao et al [76] και έγκειται στη δημιουργία marker patterns (βλ εικόνα 22) με τη χρήση του μετασχηματισμού DCT (Discrete Cosine Transform). Με τον τρόπο αυτό επιτυγχάνεται καλύτερη διακριτοποίηση μεταξύ των markers Εικόνα 22 - Σχεδιασμός markers με χρήση DCT (με αποτέλεσμα μικρότερο δείκτη σύγχυσης ανάμεσα στα markers), καθώς και μεγαλύτερη ανοσία στο θόρυβο και στη μερική επικάλυψη του pattern.



Όπως ειπώθηκε προηγουμένως, υπάρχουν σε ένα αρχείο προαποθηκευμένα τα feature vectors όλων των marker που χρησιμοποιεί η εφαρμογή. Για την ακρίβεια προαποθηκεύονται 12 (=4x3) vectors (βλ εικόνα 23) για κάθε marker, καθώς έχουμε 4 δυνατές περιστροφές, ενώ για κάθε περίπτωση περιστροφής εφαρμόζουμε 3 διαφορετικά κατώφλια, ώστε να έχουμε μία σχετική ανοχή στην αλλαγή των φωτιστικών συνθηκών (ή εφαρμόζουμε διαφορετικές αποστάσεις ανάμεσα στο marker και στην κάμερα). Με αυτό τον τρόπο, γλιτώνουμε το υπολογιστικό κόστος της online περιστροφής στον κυρίως κώδικα

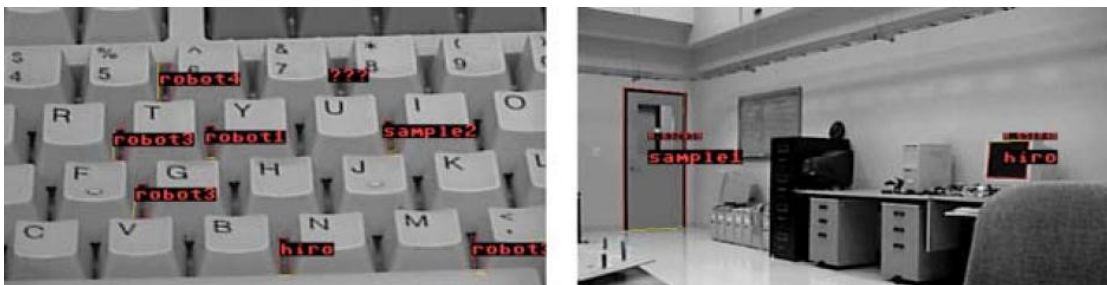
της εφαρμογής και κερδίζουμε σε frame rate. Δημιουργείται όμως η απαίτηση από το χρήστη κατά την ανάπτυξη της εφαρμογής να δημιουργήσει το αρχείο των patterns για όλα τα markers, με συγκεκριμένη κάμερα και συγκεκριμένες φωτιστικές συνθήκες, παίρνοντας ταυτόχρονα αποφάσεις για τα 3 κατώφλια “ανοχής” στη μεταβολή του φωτισμού, και προσπαθώντας να βρει το κατάλληλο trade off.

Επιπρόσθετα, ενώ σε εφαρμογές μικρής κλίμακας δεν έχουμε ιδιαίτερα προβλήματα, σε εφαρμογές μεγάλης κλίμακας δημιουργούνται αρκετά προβλήματα. Με αυξανόμενο πλήθος διαφορετικών marker δε μεγαλώνει απλά το μέγεθος του αρχείου των pattern, αλλά αυτά γίνονται και περισσότερο δυσδιάκριτα μεταξύ τους. Ταυτόχρονα, η διαδικασία ταυτοποίησης του κάθε marker γίνεται χρονοβόρα, μιας και το feature vector του κάθε εξεταζόμενου marker συγκρίνεται με correlation με κάθε vector του αρχείου των patterns και εν συνεχεία βρίσκεται ο καλύτερος συντελεστής εμπιστοσύνης μεταξύ όλων των αποτελεσμάτων. Ο αντίκτυπος, όπως είναι φυσικό, είναι φανερός στο frame rate της εφαρμογής.

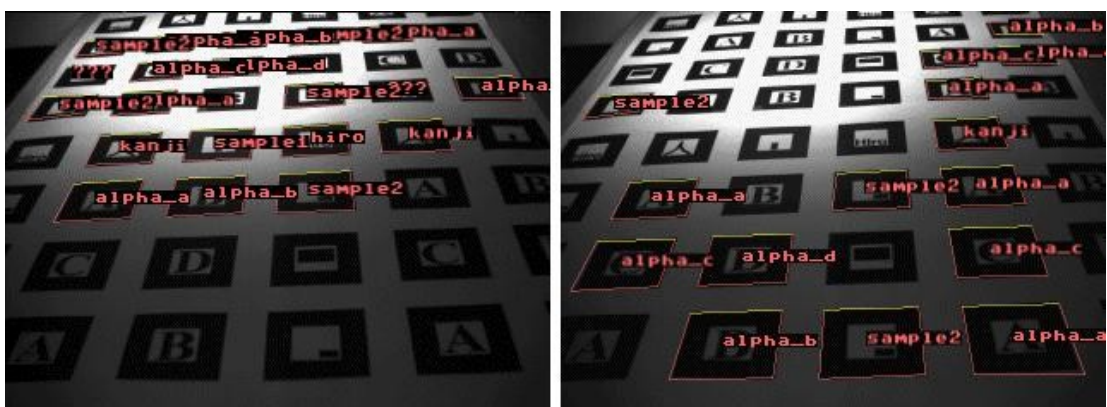


Εικόνα 23 - Τα scanlines για την ανάγνωση και τα 12 προαποθηκευμένα patterns για κάθε marker (ARToolkit)

Τέλος, η απλότητα των μεθόδων του ARToolkit, και κυρίως της μεθόδου της pose estimation, οδηγεί σε ευδιάκριτο registration error και jittering⁵⁶, καθώς και σε υψηλό δείκτη εσφαλμένης θετικής ανίχνευσης marker (false positive rate - βλ εικόνα 24)



Εικόνα 24 - Παραδείγματα false-positive ανίχνευσης markers με το ARToolkit



Εικόνα 25 - Η επίδοση του ARToolkit υπό ανομοιόμορφο φωτισμό

⁵⁶ γίνεται αντιληπτό με το μάτι όταν το εικονικό αντικείμενο δεν είναι “κολλημένο” στο marker, αλλά είτε τρεμοπαίζει (jitter) είτε η θέση του δεν είναι σωστή σε σχέση με το marker (registration error)

ARTag

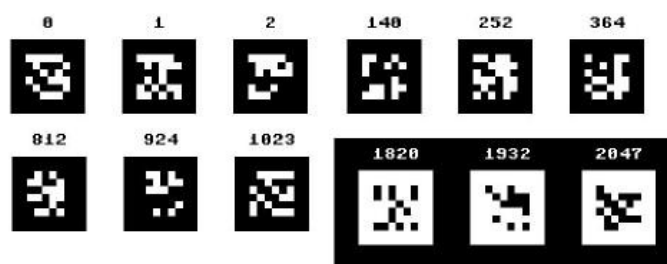
Σε μία προσπάθεια να διατηρηθούν τα θετικά στοιχεία του ARToolkit και να ξεπεραστούν τα αρνητικά σημεία στην απόδοσή του, παρουσιάστηκε από τον Mark Fiala το ARTag (2004), [75]. Όπως θα φανεί παρακάτω, οι διαφοροποιήσεις του σε σχέση με το ARToolkit επέδρασαν θετικά στο τελικό αποτέλεσμα του tracking και της αναγνώρισης των marker (με κάποιο επιπλέον υπολογιστικό κόστος). Παρόλα αυτά η διάδοση του δεν ήταν όμοια με αυτή του ARToolkit, καθώς, αν και free για μη εμπορικούς σκοπούς, ήταν κλειστού κώδικα. Σήμερα δε διατίθεται ελεύθερα, λόγω προβλημάτων σχετικά με τα πνευματικά δικαιώματα.

Οι διαφοροποιήσεις του ARTag έχουν σχέση τόσο με τα marker αυτά καθαυτά, όσο και με τις μεθόδους για τον εντοπισμό και την αποκωδικοποίησή τους:

Τα markers του ARTag

Τα markers του ARTag έχουν τετράγωνο πλαίσιο, οι 4 γωνίες του οποίου βοηθούν στην αποκωδικοποίηση του εσωτερικού του, αλλά και στην pose estimation (6DoF), ακριβώς όπως και στο ARToolkit. Μπορούμε να έχουμε (βλ εικόνα 26):

- είτε μαύρο πλαίσιο σε άσπρο φόντο
- είτε άσπρο πλαίσιο σε μαύρο φόντο.



Εικόνα 26 - Παραδείγματα markers του ARTag

Με αυτό τον τρόπο διπλασιάζεται ο αριθμός των διαφορετικών συνδυασμών-markers που μπορούν να προκύψουν, δίνοντας τη δυνατότητα ανάπτυξης μεγάλης κλίμακας AR εφαρμογών, χωρίς άμεση ανάγκη επαναχρησιμοποίησης των ίδιων markers σε διαφορετικά μέρη της εφαρμογής (πχ [77])

Επίσης, όπως φαίνεται και στην παραπάνω εικόνα, το εσωτερικό των markers δεν αποτελείται από κάποια εικόνα σε διαβαθμίσεις του γκρι, βάσει της οποίας η αναγνώριση γίνεται με correlation μεταξύ εικόνων. Τα markers αυτά είναι “index-based 2D barcodes”, και το εσωτερικό τους αποτελείται από 36 “printed bits”, διατεταγμένα σε μορφή 6x6 πίνακα (ενώ το περίγραμμα έχει πάχος 2 pixels). Από τα 36 “printed bits”

- τα 11 bits δίνουν το ID του marker. Το MSB ορίζει το χρώμα του κάδρου του marker και του αντίστοιχου φόντου (όπως παρουσιάστηκε παραπάνω), ενώ τα υπόλοιπα 10 δίνουν το sub-ID του marker, με δυνατότητα δημιουργίας 1024 διαφορετικών συνδυασμών. Το MSB διπλασιάζει, λοιπόν, το πλήθος των συνδυασμών του sub-ID σε 2048⁵⁷

⁵⁷ Αν και, από αυτά, δε γίνεται χρήση των IDs 682 και 1706, που αντιστοιχούν σε εντελώς μαύρο ή άσπρο marker, γιατί συναντώνται συχνά στο περιβάλλοντα χώρο κατά το false positive detection

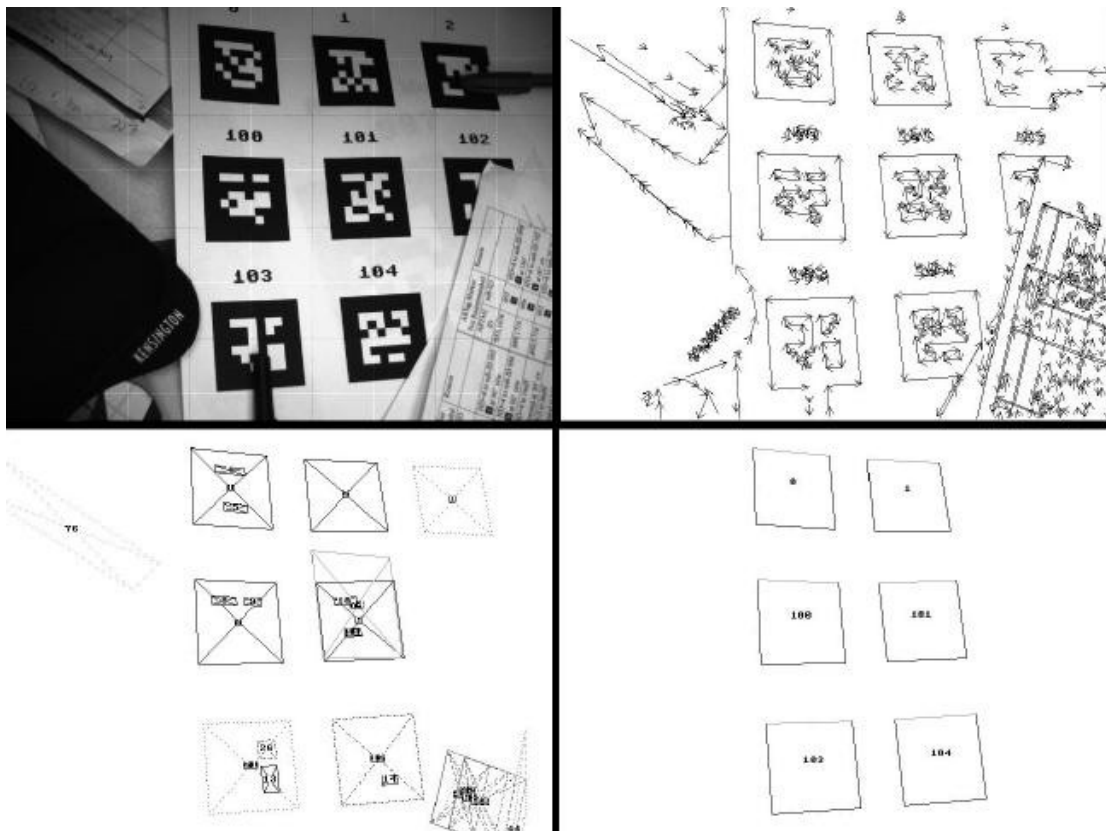
- τα υπόλοιπα 26 bits χρησιμοποιούνται για μείωση της πιθανότητας λανθασμένης ανίχνευσης ή λανθασμένης ταυτοποίησης. Η χρησιμότητά τους παρουσιάζεται αναλυτικότερα στην παράγραφο περί encoding/decoding των markers.

Τόσο η δημιουργία τους, όσο και η αποκωδικοποίησή των markers, γίνονται με τεχνικές ψηφιακού σήματος, που παρουσιάζονται παρακάτω.

Εντοπισμός των Markers

Αντί της χρήσης κάποιου (μοναδικού) σταθερού κατωφλίου, τη μετατροπή του frame σε δυαδική εικόνα και την εύρεση των τετράγωνων πλαισίων με blob detection, γίνεται ανίχνευση των τετράγωνων πλαισίων με χρήση edge-detection^{58 59} (βλ εικόνα 27).

Με τη χρήση edge detection, εκμεταλλευόμαστε και πάλι την υψηλή αντίθεση ανάμεσα στο τετράγωνο “κάδρο” του κάθε marker και στην περιοχή που το περιβάλλει, χωρίς την ανάγκη χρήση κάποιου αυστηρού κατωφλίου φωτεινότητας. Απεναντίας, γίνεται χρήση ενός (όχι τόσο αυστηρού) κατωφλίου διαφοράς φωτεινότητας, ώστε να ληφθούν υπόψη μόνο οι edges που διαχωρίζουν περιοχές με πολύ υψηλή διαφορά φωτεινότητας (όπως ανάμεσα στο κάδρο των marker και στο φόντο).



Εικόνα 27 - Απεικόνιση των βασικών βημάτων της λειτουργίας του ARTag

⁵⁸ Για την ακρίβεια, edge-detection γίνεται στην αναθεωρημένη (revised) έκδοση του ARTag [75]. Στην αρχική του έκδοση [106], ισχύουν όλα όσα αναφέρονται σε αυτή την ενότητα, με τη διαφορά ότι αντί για edge-detection χρησιμοποιείται blob-detection.

⁵⁹ στις δημοσιεύσεις δε δίνονται περαιτέρω πληροφορίες σχετικά με την ακριβή μέθοδο edge detection που χρησιμοποιείται, μιας και το σύστημα είναι κλειστού κώδικα

Έτσι (βλ εικόνα 27), αρχικά βρίσκονται τα pixels (edgels) που ανήκουν σε κάποια edge, και από αυτά κρατούνται μόνο όσα ανήκουν σε edges με υψηλή διαφορά φωτεινότητας στις δύο πλευρές της. Αυτά ενώνονται σε γραμμικά τμήματα, τα οποία εν συνεχεία ομαδοποιούνται σε τετράπλευρα. Οι 4 γωνίες κάθε τετραπλεύρου χρησιμοποιούνται για τη δημιουργία ενός homography mapping για τη δειγματοληψία του εσωτερικού τους, και εν τέλει την αναγνώριση του ID τους.

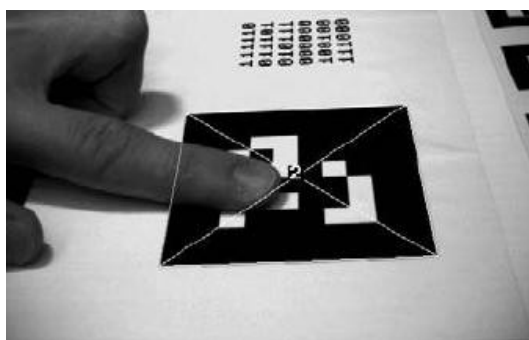
Έτσι, εξαλείφεται όχι μόνο η ανάγκη χρήσης κάποιου αυστηρού κατωφλίου φωτεινότητας, αλλά και η επιβάρυνση του χρήστη να διεξάγει πειράματα με το φωτισμό, ώστε να ορίσει ποιο θα είναι αυτό το κατώφλι.

Η ανίχνευση μπορεί πλέον να είναι επιτυχής και υπό λιγότερο ελεγχόμενες φωτιστικές συνθήκες, δηλαδή σε λιγότερο ελεγχόμενα περιβάλλοντα που πλησιάζουν περισσότερο στα πραγματικά (πχ ολόκληρη η έκταση ενός κτηρίου, με συνδυασμό φυσικού και τεχνηκού φωτισμού). Αυτό φαίνεται παραστατικότητα, με την αντιπαράθεση των εικόνων 25 και 28.



Εικόνα 28 - Η επίδοση του ARTag υπό ανομοιόμορφο φωτισμό

Επιπρόσθετα, η edge detection καθιστά πιθανή την αναγνώριση του marker ακόμα και υπό μερική επικάλυψη, όπως φαίνεται στην εικόνα 29 (σε συνδυασμό με τη χρήση error detection για το εσωτερικό του marker που θα παρουσιαστεί παρακάτω).



Εικόνα 29 - Ανίχνευση με το ARTag ενός marker υπό μερική επικάλυψη

Τέλος, προκειμένου να αποκτήσει η ανίχνευση κάποια σχετική ανοσία στο blurring, γίνεται προσπάθεια ανίχνευσης των markers σε 3 εικόνες:

- Στο κανονικό frame, με κανονική ανάλυση
- Σε μία εικόνα που προκύπτει με υποδειγματοληψία⁶⁰ του αρχικού frame κατά 2

⁶⁰ και στις δύο διευθύνσεις

- Σε μία εικόνα που προκύπτει με υποδειγματοληψία του αρχικού frame κατά 4

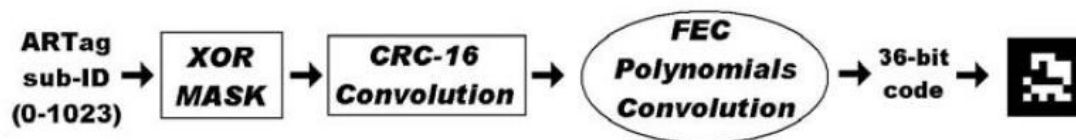
Έτσι, με τη χρήση υποδειγματοληψίας μπορεί να γίνει επιτυχής ανίχνευση ακόμα και όταν στην αρχική εικόνα (πλήρους ανάλυσης), λόγω του blurring, η διαφορά φωτεινότητας στις 2 μεριές μιας edge του marker δεν ξεπερνά την τιμή του κατωφλίου διαφοράς φωτεινότητας. Σε περιπτώσεις που απαιτείται υψηλό frame rate μπορεί να απενεργοποιηθεί η χρήση υποδειγματοληψίας, με επιτάχυνση του όλου αλγορίθμου κατά 3-4 φορές.

Τεχνικές Encoding / Decoding των Markers

Όπως προαναφέρθηκε, λόγω της φύσης των markers, η κωδικοποίησή και αποκωδικοποίησή τους γίνεται με τεχνικές ψηφιακού σήματος.

Encoding

Κατά τη διαδικασία της κωδικοποίησης έχουμε τη μετατροπή ενός δεκαδικού ID στα “printed bits” ενός marker, μέσα από τρία διαδοχικά στάδια, όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 30 - Η διαδικασία κωδικοποίησης της πληροφορίας από το ARTag

Η XOR μάσκα χρησιμοποιείται για να κάνει το ID 0 χρησιμοποιήσιμο και για να “μπερδέψει” τα “printed bits”, μιας και αναμένεται οι χρήστες να χρησιμοποιούν κυρίως τα μικρότερα IDs (0,1,2,...). Με αυτόν τον τρόπο αυτά αποκτούν μεγαλύτερη απόσταση Hamming.

Το CRC-16 πολυώνυμο που χρησιμοποιείται είναι το $x^{16} + x^{12} + x^5 + 1$, και εφαρμόζεται στο αποτέλεσμα του XOR masking με συνέλιξη με τη δυαδική ακολουθία 10001000000100000

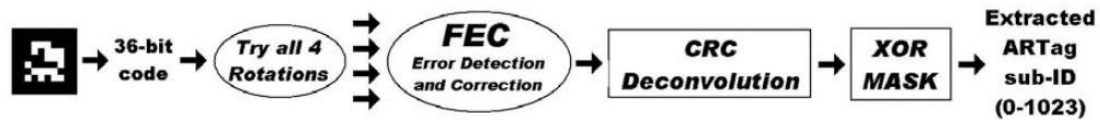
Decoding

Κατά τη διάρκεια “ανάγνωσης” του χώρου από την κάμερα, εφόσον έχουμε επιτυχή ανίχνευση ενός marker και εύρεση των 4 γωνιών του, το εσωτερικό του marker δειγματοληπτείται από ένα πλέγμα 6x6, οπότε και γίνεται “ανάγνωση” των “printed bits”. Για την “ανάγνωσή” τους γίνεται χρήση ενός κατωφλίου, που προκύπτει από τις φωτεινότητες των pixels που περιβάλλουν το περίγραμμα του marker.

Έτσι, από τα “printed bits” προκύπτουν 4 ομάδες των 36 bits, μία για κάθε πιθανή περιστροφή του τετράγωνου marker. Από αυτές, μόνο μία μπορεί να είναι έγκυρη, κάτι που βρίσκεται με τη χρήση του CRC (έπειτα από το FEC⁶¹ για διόρθωση πιθανών λαθών).

Κατά τη διάρκεια της αποκωδικοποίησης, έχουμε τη μετατροπή των “printed bits” ενός marker σε ένα δεκαδικό ID, μέσα από 4 διαδοχικά στάδια, όπως φαίνεται και στην εικόνα που ακολουθεί.

⁶¹ Forward Error Correction



Εικόνα 31 - Η διαδικασία αποκωδικοποίησης της πληροφορίας από το ARTAg

Μετά τη διόρθωση σφαλμάτων στο στάδιο του FEC, έχουμε το στάδιο της CRC αποσυνέλιξης (deconvolution), το οποίο μπορούν να περάσουν επιτυχώς μόνο $\frac{1}{2^{16}}$ δυαδικές ακολουθίες, οι οποίες αντιστοιχούν σε κάποιο έγκυρο ARTAg marker.

Βρίσκουμε έτσι ποια από τις 4 πιθανές ακολουθίες για κάθε marker είναι η ορθή, ενώ απορρίπτονται οι ακολουθίες που προέρχονται από εσφαλμένη θετική ανίχνευση⁶² (από τετράπλευρα σχήματα που δεν είναι στην πραγματικότητα markers, όπως πχ κάδρα στον τοίχο).

Με τις παραπάνω μεθόδους, έχουμε υψηλότερη επίδοση στην ανίχνευση των markers⁶³ και πολύ χαμηλό δείκτη σύγχυσης μεταξύ των markers⁶⁴.

⁶² Τα σφάλματα που συναντώνται κατά την ανίχνευση των markers

<i>false positive detection</i>	όταν η εφαρμογή καταδεικνύει την ύπαρξη marker στην εικόνα σε μία θέση όπου στην πραγματικότητα δεν υπάρχει κάποιο marker
<i>false negative detection</i>	όταν η εφαρμογή καταδεικνύει την μη ύπαρξη marker στην εικόνα σε μία θέση όπου στην πραγματικότητα υπάρχει κάποιο marker
<i>inter-marker confusion rate</i>	Ο βαθμός στον οποίο αποδίδεται σε ανιχνευμένα markers εσφαλμένη ταυτότητα

⁶³ με χαμηλότερο false negative rate και χαμηλότερο false positive error rate (<0.0078%)

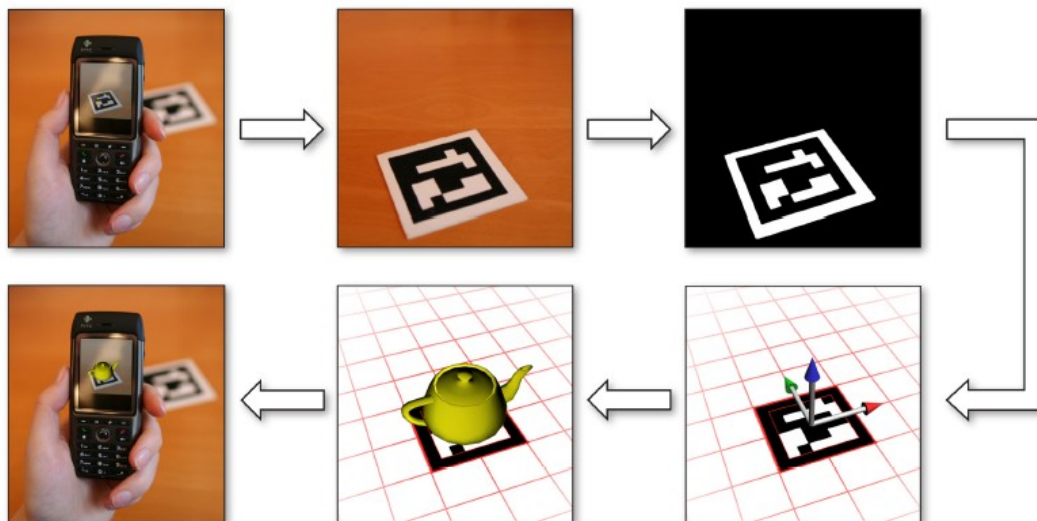
⁶⁴ inter-marker confusion rate

ARToolkit+

Η ανάπτυξη του mobile computing οδήγησε κάποια στιγμή το ενδιαφέρον των AR ερευνητών στη νέα υπολογιστική πλατφόρμα των κινητών τηλεφώνων.

Η πρώτη AR εφαρμογή που έτρεχε αυτόνομα σε κινητό τηλέφωνο (χωρίς οποιαδήποτε μετακύληση του υπολογιστικού φόρτου σε κάποιον server) αναπτύχθηκε το 2003, με τη χρήση του ARToolkit σε κινητό τηλέφωνο με λειτουργικό Windows CE [78]. Βάσει των αποτελεσμάτων αυτής της εργασίας, έγινε φανερή η ανάγκη δημιουργίας ενός νέου framework, προσαρμοσμένου στους περιορισμούς που εισάγουν τα κινητά τηλέφωνα.

Έτσι, αργότερα, παρουσιάστηκε το ARToolkitplus [79], που ήταν βασισμένο στο ARToolkit, αλλά με κύριο (όχι αποκλειστικό) προορισμό τη χρήση σε κινητά τηλέφωνα. Όπως φαίνεται και στην παρακάτω εικόνα, τα κύρια μέρη της λειτουργίας του είναι όμοια με αυτά του ARToolkit (βλ εικόνα 21)



Εικόνα 32 - Απεικόνιση των βασικών βημάτων της λειτουργίας του ARToolkitplus

Υπάρχουν όμως και αρκετές διαφοροποιήσεις⁶⁵:

Χρήση ID-based Markers

Γίνεται χρήση marker με τη μορφή 2D Barcode, το εσωτερικό των οποίων αποτελείται από “printed bits” (“δυναμικά” cells). Με αυτό τον τρόπο αποφεύγεται η διαδικασία όχι μόνο της δημιουργίας templates (εσωτερικών εικόνων των markers) και της “εκπαίδευσης” του συστήματος πάνω σε αυτά από το χρήστη, αλλά και του template matching κατά τη διάρκεια του tracking.

Επιταχύνεται λοιπόν κατά πολύ η όλη διαδικασία, ειδικά σε εφαρμογές στις οποίες γίνεται χρήση μεγάλου αριθμού διαφορετικών marker, καθώς η κωδικοποίηση/αποκωδικοποίηση του marker γίνεται με τεχνικές ψηφιακού σήματος, ενώ στο tracking δεν διεξάγεται σύγκριση εικόνων. Επιπρόσθετα, γίνεται χρήση forward error correction (CRC) για μείωση των όποιων σφαλμάτων.

⁶⁵ Μια πιο πλήρης λίστα των διαφοροποιήσεων μπορεί να βρεθεί στο [107]

Pixel Formats

Η μετατροπή του pixel format των διάφορων camera phones σε ένα format κοινής αναφοράς, το οποίο χρησιμοποιεί η εφαρμογή, επηρεάζει αρνητικά την επίδοση τους συνολικού αλγορίθμου, τόσο λόγω υπολογιστικού κόστους, όσο και λόγω της έντονης χρήσης της μνήμης κατά τη μετατροπή.

Έτσι, λήφθηκε πρόνοια για την υποστήριξη του (native) pixel format κάθε τηλεφώνου, ενώ κάποια από αυτά τα format που χρησιμοποιούνται συχνά, όπως το YUV12, ευνοούν από τη φύση τους⁶⁶ τη χρήση τους σε αλγόριθμους για γρήγορο tracking (παρόλο που στο τέλος το frame μετατρέπεται σε RGB για απεικόνισή του στην οθόνη). Κάποια άλλα format, απεναντίας, όπως το RGB565 χρειάζονται μετατροπή για ανάλογη χρήση (κάτι που μπορεί να επιταχυνθεί με κατάλληλα look up tables).

Αυτόματη μεταβολή κατωφλίου

Όπως είδαμε, στο ARToolkit ο χρήστης έχει την ευθύνη διεξαγωγής δοκιμών και ορισμού ενός απλού threshold, που χρησιμοποιείται για τη μετατροπή τους εκάστοτε frame σε δυαδική εικόνα. Η απλή κατωφλίωση του ARToolkit αποτυγχάνει, όμως, εύκολα, κυρίως λόγω των μεταβολών των συνθηκών φωτισμού.

Επίσης, παρόλο που πλέον πολλά τηλέφωνα έχουν autogain για προσαρμογή της εικόνας στο φωτισμό του περιβάλλοντος, η τακτική αυτή του απλού και μοναδικού κατωφλίου εξακολουθεί να προκαλεί αρκετά προβλήματα.

Έτσι, επειδή η κλασική λύση για αυτό το πρόβλημα, το global thresholding (από ολόκληρη την εικόνα), είναι υπολογιστικά κοστοβόρα για χρήση σε κινητά τηλέφωνα, υιοθετήθηκε η παρακάτω, αυτόματη και πολύ οικονομικότερη, μέθοδος:

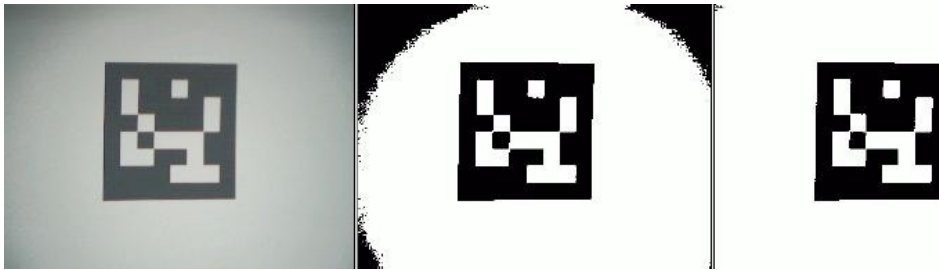
Αντί να εξετάζεται ολόκληρη η εικόνα, εξετάζονται μόνο οι περιοχές του τελευταίου frame, στις οποίες εντοπίστηκε κάποιο marker. Από αυτές τις περιοχές βρίσκεται ο μέσος όρος φωτεινότητας των pixels, που είναι και το κατώφλι που θα εφαρμοστεί για το επόμενο frame. Εάν δε βρεθεί κάποιο marker (είτε γιατί δεν υπάρχει κανένα είτε γιατί αυτή η μέθοδος εύρεσης κατωφλίου απέτυχε στη συγκεκριμένη περίπτωση), το κατώφλι παίρνει τυχαίες τιμές για κάθε καινούριο frame, μέχρι να έχουμε κάποιο επιτυχή εντοπισμό marker. Οι δοκιμές έδειξαν ότι σε περίπτωση αποτυχίας, αυτός ο μηχανισμός τυχαίου κατωφλίου οδηγεί σε επιτυχή αναγνώριση έπειτα από πολύ λίγα frames.

Αντιμετώπιση του Vignetting

Πέρα όμως από τα παραπάνω, το σταθερό κατώφλι προκαλεί και προβλήματα λόγω του έντονου vignetting⁶⁷ που εισάγουν τα (χαμηλής ποιότητας) οπτικά των τηλεφώνων, όπως φαίνεται στην εικόνα 33 που ακολουθεί. Για να λυθεί και αυτό το πρόβλημα, ο χρήστης ορίζει το μέγεθος του vignetting αριθμητικά (και όχι με τη χρήση image masking, για οικονομία στη μνήμη), με αποτέλεσμα να παίρνουμε μία δυαδική εικόνα απαλλαγμένη από τις συνέπειες του vignetting (βλ εικόνα 33 - δεξιά), με πολύ μικρό υπολογιστικό κόστος.

⁶⁶ Το YUV12 περιέχει full resolution - 8 bit - κανάλι φωτεινότητας

⁶⁷ Η παρατηρούμενη μεταβολή της φωτεινότητας και του κορεσμού των χρωμάτων σε μία εικόνα κατά την ακτινική κατεύθυνση (από το κέντρο της εικόνας προς την περιφέρεια), λόγω της παραμόρφωσης που εισάγει ο φακός ενός οπτικού συστήματος [108]



Εικόνα 33 - Η επίδραση του vignetting κατά το binarization της εικόνας και η αντιμετώπισή του από το ARToolkitplus

Προγραμματιστικές Τεχνικές για χρήση σε ARM επεξεργαστές

Δεδομένου ότι τα περισσότερα κινητά τηλέφωνα⁶⁸ έχουν ARM⁶⁹ επεξεργαστές χωρίς FPU⁷⁰ (Floating Point Unit), καθώς και περιορισμένη μνήμη χαμηλής ταχύτητας, για οικονομία στην ενεργειακή κατανάλωση, οι προγραμματιστικές τεχνικές που χρησιμοποιούνται πρέπει να λαμβάνουν υπόψη τους περιορισμένους υπολογιστικούς πόρους. Έτσι:

- Πολλά κομμάτια κώδικα που έκαναν εκτεταμένη χρήση floating points (πχ το κομμάτι του pose estimation) ξαναγράφηκαν, με αποφυγή των floating points και την προτίμηση των fixed points.
- Όπου ήταν δυνατό χρησιμοποιήθηκαν look-up tables για την επιτάχυνση της όλης διαδικασίας (παρόλο που εισάγουν κάποιο υπολογιστικό σφάλμα), όπως για παράδειγμα:
 - Για τις τριγωνομετρικές συναρτήσεις ημιτόνου και συνημιτόνου που χρησιμοποιούνται ευρέως κυρίως για το pose estimation
 - Για τα υψηλής τάξης πολυώνυμα που χρησιμοποιούνται για τη διόρθωση της παραμόρφωσης που προκαλούν τα οπτικά του φακού
 - Για τη διαίρεση fixed point αριθμών

Camera Calibration

Στη μέθοδο του ARToolkit για camera calibration προστέθηκε η υποστήριξη για το Matlab Camera Calibration Toolbox [80]. Επίσης, για την επιτάχυνση του Camera Calibration, που είναι συνήθως κοστοβόρα υπολογιστική διαδικασία, χρησιμοποιείται look up table, το οποίο είτε δημιουργείται υπολογιστικά (χρειάζεται περίπου 10 δευτερόλεπτα) είτε αποθηκεύεται σε ένα αρχείο στη μνήμη του τηλεφώνου.

Pose estimation

Ειδικά για τη χρήση του ARToolkitPlus σε desktops, προστέθηκε η μέθοδος “Robust Planar Pose Tracking” [RPP], για μεγαλύτερη αξιοπιστία και λιγότερο jitter. Η χρήση του δεν είναι δυνατή σε κινητά τηλέφωνα, λόγω της έντονης χρήσης floating point υπολογισμών.

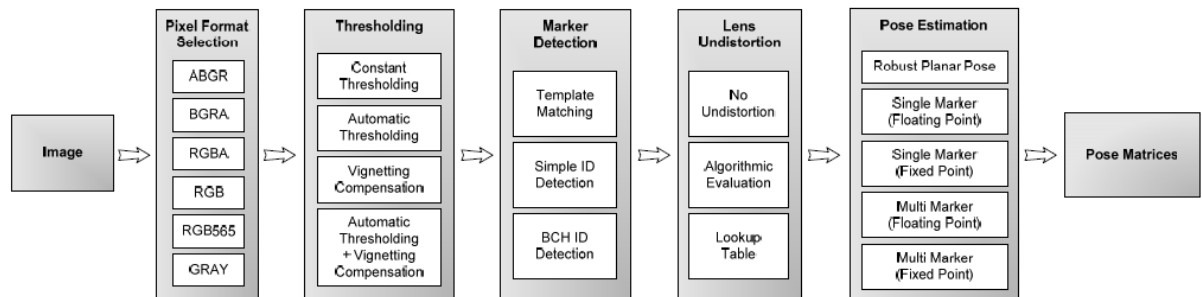
⁶⁸ Το 2007 το 98% των (περισσότερων του 1 δισεκατομμυρίων) κινητών τηλεφώνων είχε ARM επεξεργαστή, ενώ το 2009 το 90% όλων των embedded 32-bit RISC επεξεργαστών ήταν ARM αρχιτεκτονικής [109]

⁶⁹ Ένα RISC (Reduced Instruction Set Computer) ISA (Instruction Set Computer) 32 bit [109]

⁷⁰ Έτσι δεν υπάρχει επιτάχυνση των πράξεων ανάμεσα σε float αριθμούς από το υλικό. Η εκτέλεσή τους (όπως και η διαίρεση ακέραιων αριθμών) γίνεται αποκλειστικά από το software, με να είναι έως και 50 φορές πιο αργή σε σχέση με τις πράξεις ανάμεσα σε ακέραιους [79]

Συνολική λειτουργία

Στην παρακάτω εικόνα φαίνεται σε block διάγραμμα η λειτουργία του ARToolkitPlus. Σαν είσοδος στο σύστημα, θεωρείται ένα frame από την κάμερα (βλ εικόνα 32), ενώ μετά τη λειτουργία διαδοχικών μονάδων με σαφώς διακριτό ρόλο, παίρνουμε σαν έξοδο για κάθε marker που ανιχνεύτηκε επιτυχώς έναν πίνακα 4x4 με float αριθμούς, κατά το πρότυπο του OpenGL matrix format (βλ εικόνα 34).



Εικόνα 34 - Παρουσίαση των δομικών-λειτουργικών μονάδων του ARToolkitplus

Αυτός ο πίνακας μπορεί να χρησιμοποιηθεί από την εφαρμογή για την εισαγωγή εικονικών αντικειμένων στη θέση των marker, δημιουργώντας στο χρήστη την ψευδαίσθηση ότι συνυπάρχουν μαζί με τα πραγματικά στοιχεία (βλ εικόνα 32).

Προκειμένου η απόδοση της AR στα κινητά τηλέφωνα να γίνει καλύτερη, το ARToolkitPlus εγκαταλείφθηκε, και έγινε η δημιουργία του Studierstube ES [27], που ξεκίνησε από μηδενική βάση (και όχι σαν μετατροπή του ARToolkit, όπως το ARToolkitPlus) με γνώμονα τη χρήση του αποκλειστικά σε κινητά τηλέφωνα [81] [82]

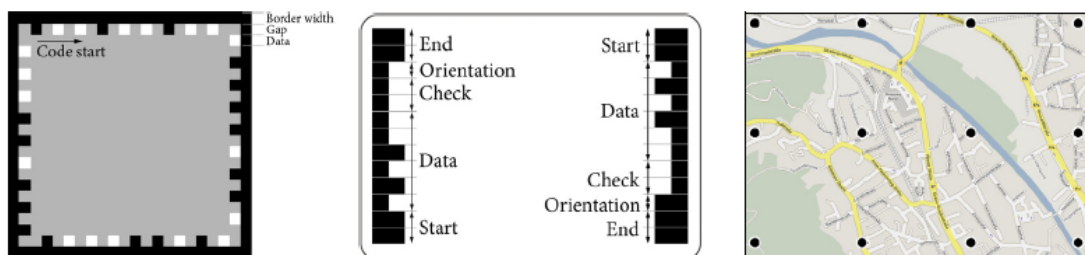
Μερικές επιπλέον αναφορές

Στις προηγούμενες παραγράφους παρουσιάστηκαν οι αντιπροσωπευτικότερες μέθοδοι για ανίχνευση markers σε AR εφαρμογές. Λόγω αυτής της αντιπροσωπευτικότητας συναντώνται σχεδόν αποκλειστικά στις αναφορές των σχετικών εργασιών.

Είναι όμως άξιες αναφοράς και ορισμένες άλλες, μεταγενέστερες, μέθοδοι:

Unobtrusive Markers

Στην εργασία τους [83] οι Wagner et al παρουσίασαν μερικά είδη markers με τέτοια μορφή, ώστε να ελαχιστοποιούν όσο το δυνατόν περισσότερο την “οπτική μόλυνση” που προκαλεί ένα visual marker. Στην εικόνα που ακολουθεί φαίνονται τα τρία είδη marker που παρουσιάζονται:



Εικόνα 35 - Μερικά είδη marker που ελαχιστοποιούν την προκαλούμενη “οπτική μόλυνση”

- Το frame marker, αριστερά στην εικόνα, είναι το νέο είδος marker που προτείνεται. Όπως φαίνεται, το εξωτερικό μαύρο κάδρο διατηρείται κατά το πρότυπο των προηγούμενων μεθόδων, ενώ η πληροφορία (τόσο τα bit δεδομένων όσο και τα parity bits) είναι κωδικοποιημένη σε ένα είδος “πληροφοριακού κάδρου” στην περιφέρεια του marker.

Το εσωτερικό του marker δε συμμετέχει στην όλη διαδικασία κωδικοποίησης/αποκωδικοποίησης της πληροφορίας, με αποτέλεσμα να μπορεί να εισαχθεί οποιαδήποτε εικόνα καταληπτή από τον άνθρωπο. Με αυτό τον τρόπο όχι μόνο μειώνεται η οπτική μόλυνση, αλλά ευνοείται και η ανάπτυξη διαφημιστικών εφαρμογών με τη χρήση λογοτύπων εταιρειών.

Η όλη κωδικοποίηση/αποκωδικοποίηση της πληροφορίας γίνεται με τεχνικές ψηφιακού σήματος, ενώ το μέγεθος των “printed bits” μπορεί να μεταβάλλεται κατά βούληση, ρυθμίζοντας έτσι το βαθμό στον οποίο κάνουν “ενοχλητική” την παρουσία τους.

- Το split marker, στη μέση της εικόνας, είναι ένα είδος marker που εμπνεύστηκε τα markers του Rekimoto για το Eye of Judgment [23]. Η πληροφορία είναι κωδικοποιημένη μόνο σε δύο από τις τέσσερις πλευρές του marker, ενώ παρόλο που δεν υπάρχει τετράγωνο μαύρο πλέγμα, χρησιμοποιούνται δύο μαύρες γραμμές-οδηγοί που καθοδηγούν τη διαδικασία του tracking και της δειγματοληψίας για την αναγνώριση του ID.
- Τα dot markers, δεξιά στην εικόνα, είναι ιδανικά για τη “σήμανση” μεγάλων επιφανειών, σε αντίθεση με τα markers που παρουσιάστηκαν ως εδώ. Με τη χρήση

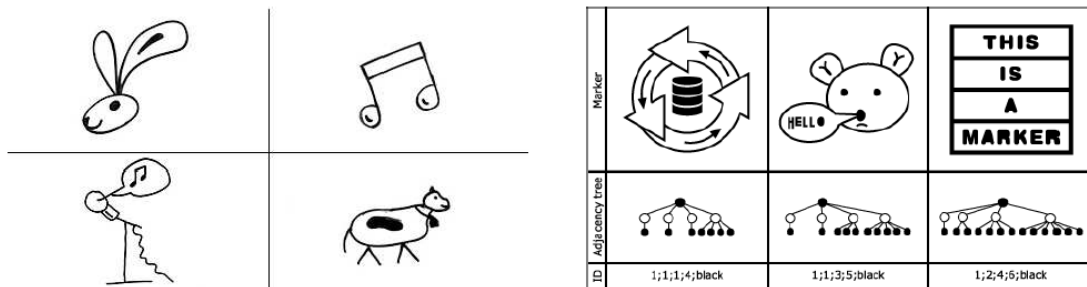
τους τυπώνεται σε μία επιφάνεια (πχ σε ένα μεγάλο χάρτη) ένα πλέγμα από μαύρες κουκίδες με άσπρη περιφέρεια-δαχτυλίδι. Αυτές οι κουκίδες “τεμαχίζουν” (ανά γειτονικές τετράδες) την όλη επιφάνεια σε μικρά τετραγωνάκια, κάτι που επιταχύνει σημαντικά τη χρήση οποιασδήποτε μεθόδου natural feature tracking που αλλιώς θα ήταν πιο χρονοβόρα.

Ουσιαστικά πρόκειται για μία υβριδική μέθοδο συνδυασμού natural feature και marker tracking

Designable Markers

Μία άλλη μέθοδος ελαχιστοποίησης της οπτικής μόλυνσης που προκαλούν τα markers προτάθηκε από τους Costnza et al [84], και έγκειται στη δημιουργία ασπρόμαυρων (bitonal) markers πλήρως καταληπτών από τον άνθρωπο.

Όπως φαίνεται και στην εικόνα που ακολουθεί, τα markers είναι απλά σχέδια που μπορούν να γίνουν και με το χέρι, θα πρέπει όμως να ακολουθηθούν κάποιιοι κανόνες⁷¹.



Εικόνα 36 - Designable markers, πλήρως "αναγνώσιμα" από τον άνθρωπο

Αυτό γίνεται δυνατό καθώς κατά την ανίχνευση των markers δεν παίζει ρόλο η γεωμετρία του marker, αλλά η τοπολογία του. Το κάθε frame της κάμερας μετατρέπεται σε δυαδική εικόνα και έπειτα ανιχνεύονται οι διάφορες άσπρες ή μαύρες περιοχές. Ανάλογα με το φωλιάσμα αυτών των περιοχών προκύπτει ένα δέντρο (βλ εικόνα 36, δεξιά) που περιγράφει τη γειτνίαση των περιοχών⁷² (region adjacency tree). Έτσι, η αναζήτηση της ύπαρξης των markers (και μετέπειτα της ταυτοποίησης) γίνεται με αναζήτηση (των υπόδεντρων) σε αυτό το δέντρο, ένα πρόβλημα γνωστό σαν “subtree isomorphism problem”.

Στη διαδικασία που περιγράφεται παραπάνω δεν εμπλέκονται floating point υπολογισμοί, με αποτέλεσμα αυτή η μέθοδος να είναι κατάλληλη για χρήση και σε κινητά τηλέφωνα [85].

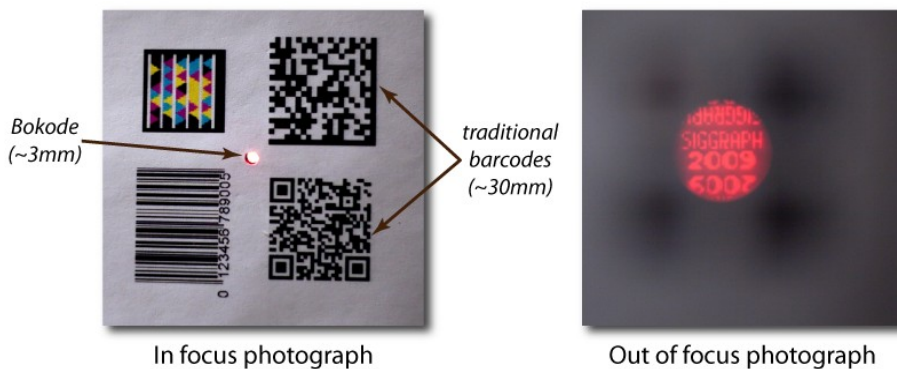
⁷¹ Ένα έγκυρο marker αποτελείται από μία μαύρη περιοχή που περικλείει 3 ή περισσότερες άσπρες περιοχές, οι μισές ή περισσότερες από τις οποίες πρέπει να περικλείουν μία ή περισσότερες μαύρες περιοχές. Τα παραπάνω χρώματα μπορούν να αναστραφούν. Επιτρέπονται δηλαδή 3 επίπεδα φωλιάσματος, για απλοποίηση της όλης διαδικασίας.

⁷² Κάθε περιοχή είναι ένας κόμβος. Η περιοχή-φόντο είναι η ρίζα του δένδρου. Κάθε φορά που παρουσιάζεται φωλιασμένη περιοχή, αυτή αντιστοιχεί σε κόμβο (child) χαμηλότερου επιπέδου.

Bokode

Τέλος παρουσιάζεται ένα marker system που ακολουθεί μία διαφορετική προσέγγιση σε σχέση με τα υπόλοιπα.

Όπως φαίνεται και στην εικόνα που ακολουθεί τα Bokode⁷³ markers είναι σημαντικά μικρότερα σε σχέση με τα υπόλοιπα markers, ενώ η ανάγνωσή τους μπορεί να γίνει με μία συνηθισμένη κάμερα ακόμα και από μερικά μέτρα απόσταση.



Εικόνα 37 - Η χρήση του φαινομένου bokeh στην ανίχνευση των markers

Αυτό γίνεται δυνατό με την εκμετάλλευση του φαινομένου bokeh που συναντάται στις φωτογραφικές μηχανές ή κάμερες λόγω των οπτικών στοιχείων του φακού.

Με τη χρήση ενός επιπλέον φακού (lenslet) πάνω στο marker και με μία συνηθισμένη κάμερα, ρυθμισμένη ώστε να βγάζει out of focus φωτογραφίες (βλ παραπάνω εικόνα), μπορούμε να έχουμε αποτελεσματική ανίχνευση και pose estimation ακόμα και από σχετικά μεγάλη απόσταση.

Το πλεονέκτημα αυτό έχει φυσικά το αυξημένο κόστος δημιουργίας των markers, καθώς (λόγω του απαιτούμενου lenslet) δεν αρκεί ένας απλός εκτυπωτής.

Επίσης, το γεγονός ότι η κάμερα πρέπει να βγάζει out of focus φωτογραφίες/frames δημιουργεί την ανάγκη χρήσης μίας επιπλέον κάμερας για AR εφαρμογές, εκτός και αν η εφαρμογή στοχεύει στην επαύξηση της πραγματικότητας με άλλο τρόπο (πχ φωνητικές οδηγίες).

⁷³ bokeh code

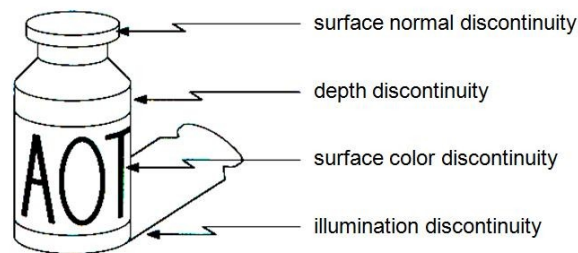
4. Μαθηματικό υπόβαθρο

Εντοπίζοντας τις Edges

Όπως αναφέρθηκε σε προηγούμενη ενότητα, ένας τρόπος για να εξάγουμε ενδιαφέρουσα πληροφορία από μία εικόνα, και να απορρίψουμε την περιττή, είναι κάνοντας edge detection (ένας από τους διάφορους τρόπους feature extraction).

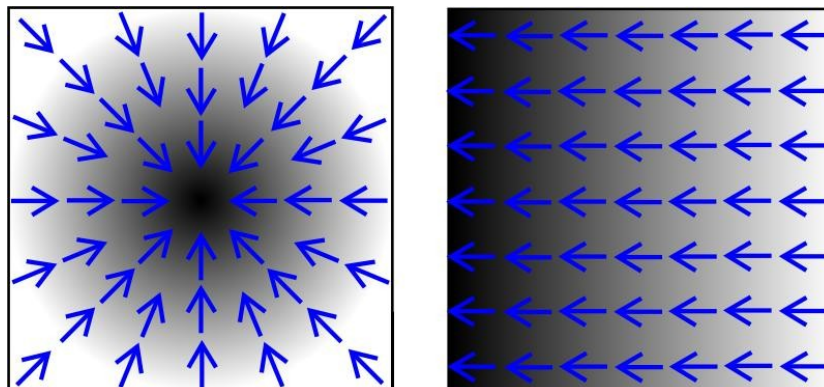
Η εύρεση των edges είναι μία συνηθισμένη διαδικασία, καθώς οι edges αντιστοιχούν σε ενδιαφέροντα σημεία σε μία εικόνα (βλ εικόνα 38):

- στις ακμές κάποιου αντικειμένου
- όπου έχουμε εναλλαγή του βάθος πεδίου
- όπου έχουμε έντονη χρωματική μεταβολή
- όπου έχουμε απότομη μεταβολή της φωτεινότητας



Εικόνα 38 - Διάφορες περιπτώσεις εμφάνισης edges σε μία εικόνα

Όταν, όμως, δουλεύουμε με ασπρόμαυρες εικόνες (με grayscale values της φωτεινότητας, από 0 έως 255) όλες οι παραπάνω περιπτώσεις παρουσίας edge μεταφράζονται σε απότομες μεταβολές (discontinuities) της φωτεινότητας.



Εικόνα 39 - Παραδείγματα της intensity gradient σε μία εικόνα

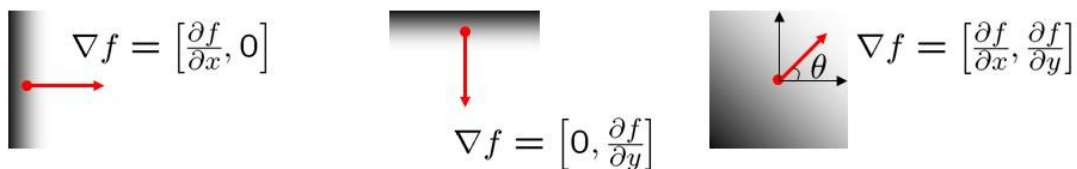
Ορισμός της Intensity Gradient

Το μαθηματικό εργαλείο που εκφράζει τις μεταβολές ενός σήματος ή μιας συνάρτησης, είναι η παράγωγος, την οποία και χρησιμοποιούμε για το edge detection.

Οι ασπρόμαυρες εικόνες, όμως, παριστάνονται με έναν δυσδιάστατο πίνακα ακέραιων τιμών, αποτελώντας έτσι ένα δυσδιάστατο διακριτό σήμα φωτεινότητας.

Προκειμένου να ορίζεται η παράγωγος της φωτεινότητας, κάνουμε την υπόθεση ότι η εικόνα δεν αποτελεί μία δυσδιάστατη διακριτή συνάρτηση φωτεινότητας, αλλά παριστάνεται από τη δειγματοληψία, σε δυσδιάστατο πλέγμα, μίας συνεχούς δυσδιάστατης συνάρτησης φωτεινότητας [86].

Η παράγωγος μίας δυσδιάστατης συνάρτησης (εδώ της φωτεινότητας) δίνει για κάθε σημείο ένα διάνυσμα, τα στοιχεία του οποίου είναι η οριζόντια και η κάθετη συνιστώσα της παραγωγού. Το διάνυσμα αυτό δείχνει προς την κατεύθυνση της μέγιστης μεταβολής φωτεινότητας, ενώ το μέτρο του εκφράζει το πόσο απότομη είναι αυτή η μεταβολή (βλ εικόνες 39 και 40).



Εικόνα 40 - Οι διάφορες περιπτώσεις της intensity gradient σε μία εικόνα

Έτσι, σε κάθε σημείο η παράγωγος της φωτεινότητας παριστάνεται ως

$$\nabla f = [G_x \quad G_y]^T = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T \quad (1.1)$$

Ενώ το μέτρο της και η γωνία της δίνονται από τους τύπους

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \quad (1.2)$$

$$ang(\nabla f) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (1.3)$$

όπου η παραπάνω γωνία έχει σύνολο τιμών το $(-90, +90)$ και ουσιαστικά δίνει απλά τη διεύθυνση του διανύσματος.

Με τη χρήση, όμως, αυτή της διεύθυνσης και των συνιστωσών της παραγωγού, μπορεί να βρεθεί εύκολα η κλίση της παραγωγού, με μία αυτοσχέδια συνάρτηση στο Matlab.

Εκτιμώντας την image Gradient

Όταν χρειαζόμαστε την παράγωγο της φωτεινότητας, αρκούμαστε σε μία εκτίμηση αυτής, μέσω των συνιστωσών της. Το σφάλμα που εισάγεται από την εκτίμηση δεν έχει πάντα ιδιαίτερες επιπτώσεις, καθώς, ανάλογα με την εφαρμογή, ακόμα και μία χοντρική αλλά γρήγορη εκτίμηση μπορεί να έχει πολύ μεγάλη πρακτική αξία.

Έτσι, ανάγουμε το πρόβλημα της εκτίμησης της παραγώγου ενός 2D σήματος, στην εκτίμηση των μερικών παραγώγων στην κάθετη και οριζόντια κατεύθυνση (1D πρόβλημα)

Μερικοί απλοί τρόποι προσέγγισης της μερικής παραγώγου μίας 1D εικόνας είναι οι παρακάτω

Προσέγγιση μερικής παραγώγου	Kernel
$\frac{\partial f}{\partial x}(x, y) \approx f[x+1, y] - f[x, y]$	$[0 \quad -1 \quad 1]$
$\frac{\partial f}{\partial x}(x, y) \approx f[x, y] - f[x-1, y]$	$[-1 \quad 1 \quad 0]$
$\frac{\partial f}{\partial x}(x, y) \approx \frac{1}{2}f[x+1, y] - \frac{1}{2}f[x-1, y]$	$\left[-\frac{1}{2} \quad 0 \quad \frac{1}{2} \right]$

Πίνακας 3 - Διάφορες περιπτώσεις προσέγγισης της μερικής παραγώγου

Ο καλύτερος, ίσως, τρόπος από τους παραπάνω, ώστε να λαμβάνονται υπόψη και οι 2 άμεσοι γείτονες του υπό εξέταση pixel, είναι ο τρίτος, δηλαδή η χρήση central differences

$$\begin{cases} G_x(x, y) = -a \cdot f(x-1, y) + 0 \cdot f(x, y) + b \cdot f(x+1, y) \\ G_y(x, y) = -a \cdot f(x, y-1) + 0 \cdot f(x, y) + b \cdot f(x, y+1) \end{cases} \quad (1.4)$$

ή, σε μορφή συνέλιξης⁷⁴

$$\begin{cases} G_x(x, y) = [-a \quad 0 \quad b] * f \\ G_y(x, y) = [-a \quad 0 \quad b]^T * f \end{cases} \quad (1.5)$$

όπου το άθροισμα των απολύτων τιμών των στοιχείων του kernel ισούται με τη μονάδα

$$a + b = 1 \quad (1.6)$$

⁷⁴ Η συνέλιξη αποτελεί ένα είδος “weighted average sum” της τοπικής περιοχής του σήματος εισόδου (συνάρτησης φωτεινότητας), και για διακριτά σήματα ορίζεται ως [113]

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m] \cdot g[m]$$

Βλέπουμε λοιπόν, ότι σε κάθε pixel που εξετάζεται, χρησιμοποιείται ένα kernel (παράθυρο) μήκους τριών pixels κεντραρισμένο σε αυτό. Το αποτέλεσμα λαμβάνει (“συμμετρικά”) υπόψη τους δύο άμεσους γείτονες του κάθε pixel (στην κάθετη και οριζόντια διεύθυνση)⁷⁵.

Προκειμένου να γίνει μια πιο αξιόπιστη εκτίμηση, με μεγαλύτερη ανοσία στο θόρυβο⁷⁶, θα μπορούσαν να ληφθούν υπόψη τα 4 άμεσα γειτονικά pixels (2 σε κάθε μεριά), δηλαδή να γίνει χρήση κάποιου kernel μήκους 5 pixels (κεντραρισμένο στο υπό εξέταση pixel). Σε αυτή την περίπτωση, οι παραπάνω εξισώσεις παίρνουν τη μορφή

$$\begin{cases} G_x(x,y) = [-a & -b & 0 & c & d] * f \\ G_y(x,y) = [-a & -b & 0 & c & d]^T * f \end{cases} \quad (1.7)$$

ενώ, αν οι δύο γείτονες κάθε πλευράς που ισαπέχουν από το υπό εξέταση pixel έχουν τον ίδιο συντελεστή “βαρύτητας”, έχουμε

$$\begin{cases} G_x(x,y) = [-a & -b & 0 & b & a] * G \\ G_y(x,y) = [-a & -b & 0 & b & a]^T * G \end{cases} \quad (1.8)$$

Με

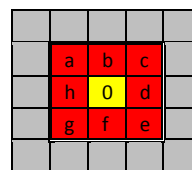
$$2a + 2b = 1 \quad (1.9)$$

⁷⁵ Ο γνωστός edge detector “Sobel”, βασίζεται στα παρακάτω φίλτρα

$$G_x(x,y) = \frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * f \quad G_y(x,y) = \frac{1}{8} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * f$$

Για κάθε συνιστώσα λαμβάνονται, δηλαδή, υπόψη και τα 8 γειτονικά pixels που περιτριγυρίζουν σε κάθε κατεύθυνση το υπό εξέταση pixel, με τη χρήση ενός kernel 3x3 [111]

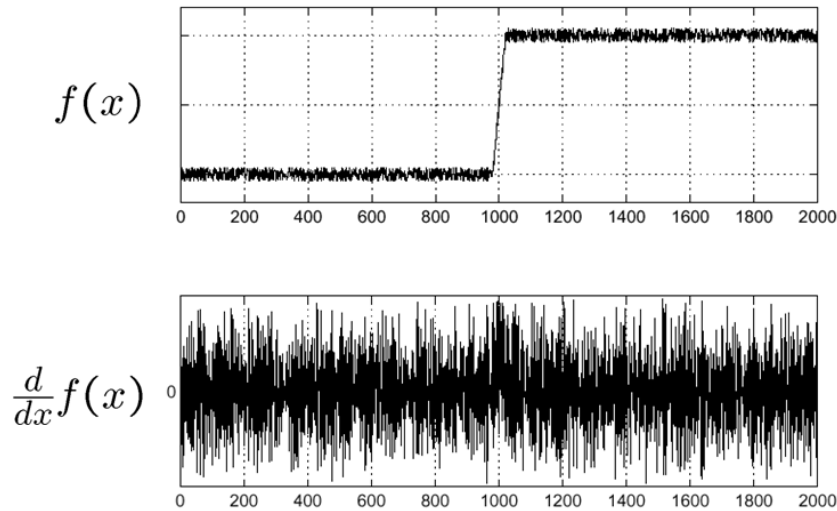
$$L(x,y) = \begin{bmatrix} a & b & c \\ h & 0 & d \\ g & f & e \end{bmatrix} * f$$



⁷⁶ αν και όσο μεγαλώνει το kernel αυξάνει και το σφάλμα σχετικά με την ακριβή θέση του edge (λόγω του smoothing της εικόνας)

Derivative of Gaussian

Οι ιδανικές edges μοντελοποιούνται με μία βηματική συνάρτηση. Όπως είναι φυσικό, όμως, τέτοιες ιδανικές edges δεν παρατηρούνται στην πραγματικότητα. Απεναντίας, οι πραγματικές edges έχουν τη μορφή που παρουσιάζεται στην παρακάτω εικόνα (περίπτωση 1D εικόνας), όπου η επίδραση του θορύβου είναι εμφανής.



Εικόνα 41 - Η επίδραση του θορύβου σε ένα 1D σήμα κατά την παραγωγή του

Εάν διαφορίσουμε απευθείας αυτό το σήμα, για να εκτιμήσουμε την κλίση της συνάρτησης φωτεινότητας (intensity gradient), βλέπουμε ότι η παραγωγή λειτουργεί σαν ανωδιαβατό φίλτρο, εις βάρος της “ωφέλιμης” εικόνας και ενισχύοντας την επίδραση του θορύβου, καθώς:

- Η εικόνα αποτελείται κυρίως από χαμηλότερες συχνότητες, με εξαίρεση ορισμένες περιοχές, όπως οι edges, ενώ
- Ο θόρυβος αποτελείται κυρίως από υψηλότερες συχνότητες⁷⁷

Έτσι, πάντα το πρώτο βήμα στην όλη διαδικασία είναι το smoothing⁷⁸ της εικόνας, με κάποιο χαμηλοπερατό φίλτρο, ώστε να αποκοπούν οι υψηλές συχνότητες, που οφείλονται κυρίως στο θόρυβο (με την παράπλευρη μερική απώλεια πληροφορίας σε περιοχές όπως οι edges), ενώ να κρατηθούν οι χαμηλότερες.

Μία συνηθισμένη μέθοδος αποθορυβοποίησης, είναι η συνέλιξη με κάποιο πυρήνα, που προκύπτει με δειγματοληψία μιας Gaussian συνάρτησης.

Η Gaussian συνάρτηση δίνεται από τον τύπο

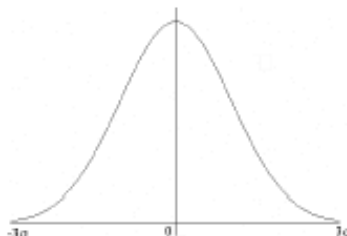
$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1.10)$$

⁷⁷ στην περίπτωση του λευκού θορύβου, έχει ομοιόμορφο φασματικό περιεχόμενο

⁷⁸ Για κάθε pixel παίρνουμε το σταθμευμένο μέσο όρο της φωτεινότητας αυτού και των γειτονικών του pixels

όπου μ είναι η μέση τιμή⁷⁹, και σ είναι η τυπική απόκλιση⁸⁰.

Από τη γραφική παράσταση της Gaussian συνάρτησης (βλ εικόνα 42) φαίνεται ότι αυτή μπορεί να χρησιμεύσει για την εξαγωγή ενός είδους σταθμευμένου μέσου όρου.



Εικόνα 42 - Μορφή μιας Gaussian συνάρτησης

Όταν αυτή χρησιμοποιείται για διαδικασία αποθρομβοποίησης⁸¹, οπότε και θέλουμε τη μεγαλύτερη επίδραση στη σταθμευμένο μέσο όρο να την έχει το υπό εξέταση pixel και όχι τα γειτονικά του, παίρνουμε, για $\mu = 0$

$$G_0(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (1.11)$$

που λειτουργεί σαν χαμηλοπερατό φίλτρο.

Με τη χρήση της Gaussian συνάρτησης, μόνο η επιρροή των πιο κοντινών γειτονικών pixels είναι σημαντική, ενώ καθώς αυξάνεται η απόσταση των γειτόνων από το υπό εξέταση pixel, η επιρροή τους μειώνεται με πολύ γρήγορο ρυθμό, τείνοντας στο μηδέν.

Το πόσο μεγάλη είναι η περιοχή των γειτόνων που επηρεάζουν αρκετά το αποτέλεσμα, και κατ' επέκταση την ένταση του smoothing, ρυθμίζεται από το σ , που ρυθμίζει το “άνοιγμα” της “καμπάνας” της Gaussian συνάρτησης. (όσο μεγαλύτερο είναι το σ , τόσο πιο έντονο είναι και το smoothing της εικόνας).

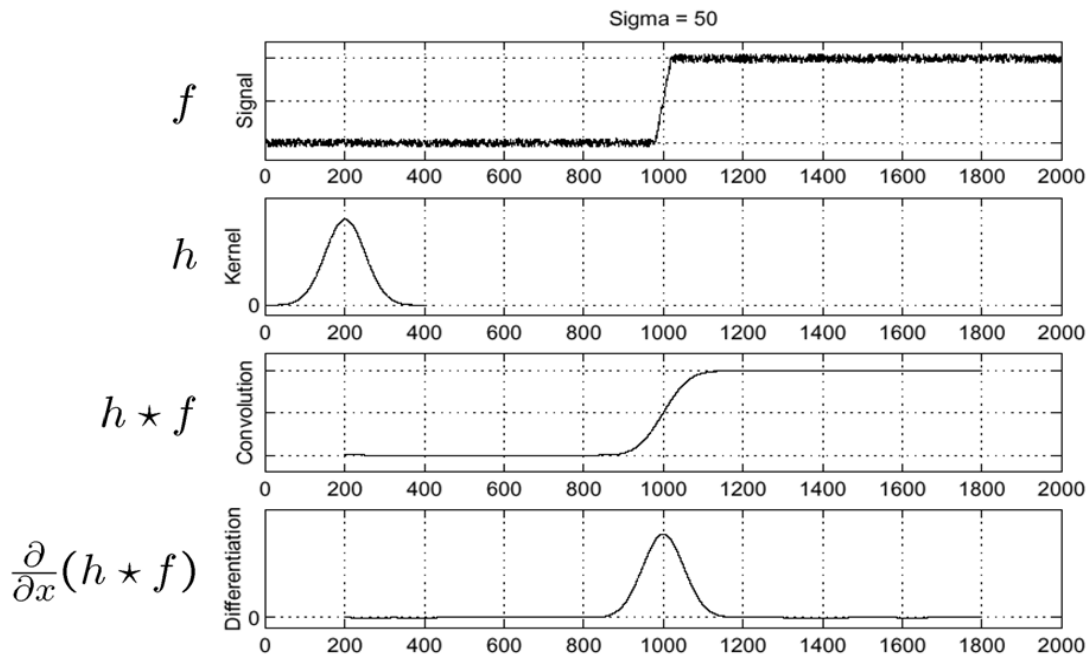
Η επιρροή, όμως, ακόμα και των πιο μακρινών γειτόνων, αν και ασήμαντη, είναι μη μηδενική, δηλαδή από τη Gaussian συνάρτηση προκύπτει φίλτρο άπειρης διάστασης.

Έτσι, λαμβάνουμε συνήθως υπόψη μόνο την άμεση γειτονιά του κάθε pixel, αγνοώντας αυθαίρετα τα υπόλοιπα, και θεωρώντας το σφάλμα που εισάγεται ανεπαίσθητο.

⁷⁹ Δίνει τη θέση της μέγιστη τιμής της Gaussian συνάρτησης στον οριζόντιο άξονα (περίπτωση 1D)

⁸⁰ Ορίζει το άνοιγμα του “καμπανοειδούς” γραφήματος της Gaussian συνάρτησης. Όσο μεγαλύτερη τιμή έχει, τόσο πιο πολύ “απλώνεται” η καμπανοειδής γραφική παράσταση.

⁸¹ Σύμφωνα με τον Canny [51], η ιδανική συνάρτηση smoothing είναι προσεγγιστικά όμοια με την Gaussian συνάρτηση



Εικόνα 43 - Συνέλιξη ενός σήματος με Gaussian kernel και παραγωγή του αποτελέσματος

Έτσι, έχουμε πλέον την διαδικασία που απεικονίζεται παραπάνω. Όπως φαίνεται,

- Στην πρώτη γραφική παράσταση έχουμε τη συνάρτηση φωτεινότητας μιας 1D εικόνας (1D σήμα)
- Στη δεύτερη έχουμε το kernel που προκύπτει με χρήση Gaussian συνάρτησης
- Στην τρίτη έχουμε το αποτέλεσμα της αποθρομβοποίησης (smoothing της εικόνας), μετά τη συνέλιξη των δύο παραπάνω σημάτων
- Στο τέλος έχουμε την παραγωγή του αποθρομβοποιημένου σήματος. Όπως φαίνεται, η θέση στην οποία παρουσιάζεται η μέγιστη τιμή, αντιστοιχεί στην edge του 1D σήματος

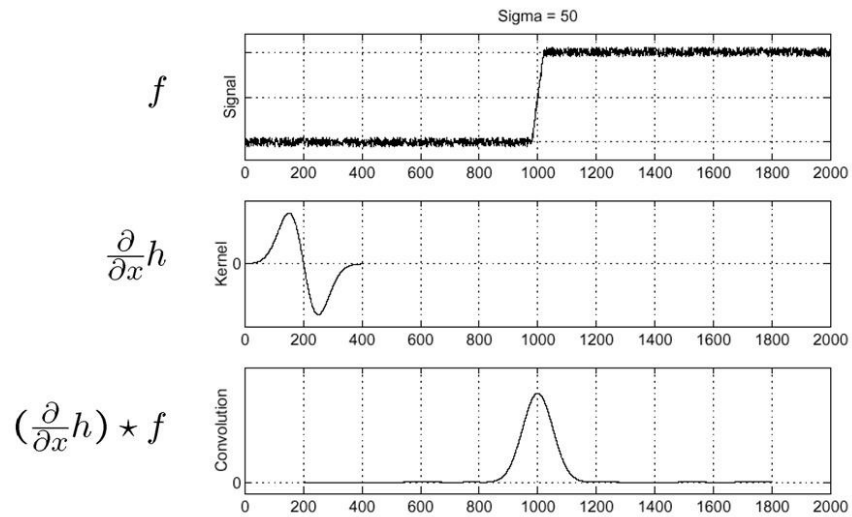
Τόσο όμως ο τελεστής της διαφόρισης, όπως και ο τελεστής της συνέλιξης, είναι γραμμικοί.

Έτσι έχουμε:

$$\frac{d}{dx}(f * h) = \frac{df}{dx} * h = f * \frac{dh}{dx} \quad (1.12)$$

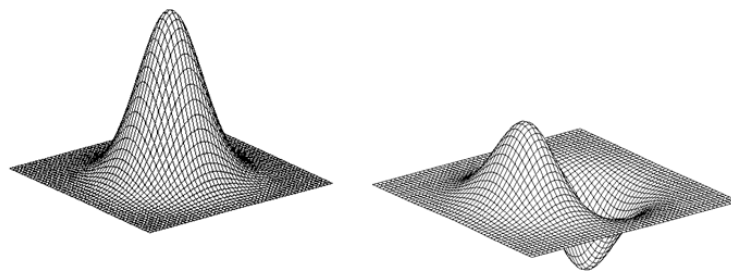
μπορούμε, δηλαδή, να συνέλιξουμε το 1D σήμα f με ένα kernel $\frac{dh}{dx}$ που προκύπτει με

δειγματοληψία μίας derivative of Gaussian συνάρτησης (η μορφή της φαίνεται παραστατικότερα στην εικόνα 45), όπως φαίνεται στην ακόλουθη εικόνα



Εικόνα 44 - Συνέλιξη ενός σήματος με Derivative of Gaussian kernel

Όπως βλέπουμε, εάν έχει προϋπολογιστεί η derivative of Gaussian (όπως πριν είχε προϋπολογιστεί η απλή Gaussian), φτάνουμε στο ίδιο αποτέλεσμα, με ένα βήμα λιγότερο.

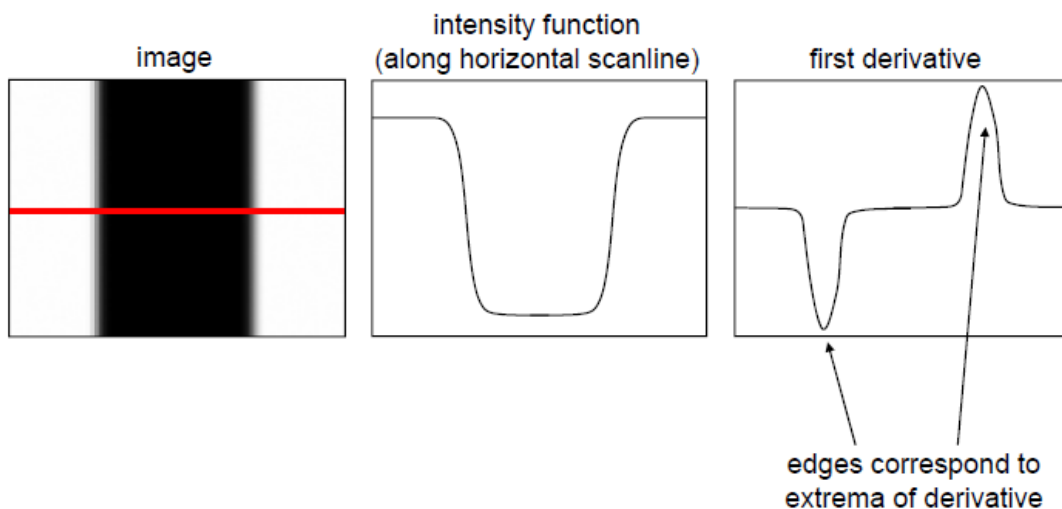


Εικόνα 45 - Δυσδιάστατη γραφική παράσταση μιας Gaussian και μιας Derivative of Gaussian συνάρτησης

Εντοπισμός της θέσης των edgels

Μία απλή μέθοδος για την εύρεση των pixels που ανήκουν σε κάποιο edge (edgels) έχει ως εξής:

1. Αρχικά υπολογίζεται σε κάθε pixel (ή στα pixels που μας ενδιαφέρουν) μία εκτίμηση του μέτρου της παραγώγου της φωτεινότητας (gradient magnitude), σαν μέτρο της “έντασης” των edges, καθώς και η διεύθυνση και η κλίση της παραγώγου αυτής.
2. Έπειτα εντοπίζονται τα τοπικά ακρότατα του μέτρου της παραγώγου (βλ εικόνα 46), που εκτιμήθηκε στο προηγούμενο βήμα. Για την εύρεση της θέσης τους με sub-pixel ακρίβεια μπορεί να γίνει χρήση της μεθόδου maxima of quadratics, που παρουσιάζεται παρακάτω.
3. Τέλος, εφαρμόζεται ένα κατώφλι, έτσι ώστε να κρατηθούν μόνο οι πιο “έντονες” και, άρα, σημαντικότερες edges, ενώ να απορριφτούν οι πιο αδύνατες, που είτε δεν παρουσιάζουν ενδιαφέρον, είτε οφείλονται σε θόρυβο. Η επιλογή αυτού του κατωφλίου αποτελεί ένα trade off:
 - ένα χαμηλό threshold οδηγεί σε
 - αποτέλεσμα με πολλή περιττή πληροφορία προς επεξεργασία (άρα και πτώση της ταχύτητας εκτέλεσης του αλγορίθμου)
 - επιρρέπεια στο θόρυβο
 - ένα υψηλό threshold οδηγεί σε
 - συγκράτηση μόνο λίγων πολύ σημαντικών πληροφοριών (άρα επιταχύνεται η εκτέλεση του αλγορίθμου),
 - απώλεια χρήσιμης πληροφορίας
 - edges που παρουσιάζουν μεγάλο βαθμό τεμαχισμού



Εικόνα 46 - Εντοπισμός των edgels μέσω της intensity gradient

Sub-Pixel Ακρίβεια

Έχοντας βρει τη (“χονδρική”) θέση ενός edgel, με ακρίβεια της τάξης του pixel, είναι σημαντικό να προχωρήσουμε στην εύρεση της ακριβούς θέσης με sub-pixel ακρίβεια.

Εν γένει, αυτό είναι δυνατό με τη χρήση του εκάστοτε pixel-edgel και των δύο άμεσων γειτόνων του. Γνωρίζοντας τη θέση των παραπάνω pixels και την τιμή της φωτεινότητας σε αυτά, προσεγγίζουμε μία συνάρτηση δευτέρου βαθμού που δείχνει ακριβέστερα τη μεταβολή της φωτεινότητας σε αυτά τα τρία pixels. Η θέση του τοπικού ακρότατου αυτής της συνάρτησης είναι και η ακριβέστερη (sub-pixel) θέση του edgel.

Maxima of Quadratics

Για να υπολογίσουμε τους συντελεστές του πολυωνύμου n -οστού βαθμού

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (1.13)$$

που προσεγγίζει τη συνάρτηση $f(x)$ στα σημεία $[x_0, f(x_0)], \dots, [x_{m-1}, f(x_{m-1})]$ σχηματίζουμε τις m γραμμικές εξισώσεις

$$p_n(x_i) = a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n = f(x_i) \quad (1.14)$$

για $i = 0, \dots, m-1$

Αυτές ορίζουν ένα γραμμικό σύστημα m εξισώσεων με n αγνώστους

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m-1} & x_{m-1}^2 & \dots & x_{m-1}^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{m-1}) \end{bmatrix} \quad (1.15)$$

Θέτοντας

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m-1} & x_{m-1}^2 & \dots & x_{m-1}^n \end{bmatrix} \quad a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} \quad y = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{m-1}) \end{bmatrix} \quad (1.16)$$

έχουμε

$$Aa = y \quad (1.17)$$

όπου ο πίνακας A είναι εν γένει μη τετράγωνος, και άρα μη αντιστρέψιμος.

Προκειμένου να γίνει δυνατή η αντιστροφή πολλαπλασιάζουμε με τον πίνακα A^T

$$A^T Aa = A^T y \quad (1.18)$$

Και παίρνουμε τη λύση

$$a = (A^T A)^{-1} A^T y \quad (1.19)$$

Αντικαθιστώντας τους συντελεστές a_i προκύπτει

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (1.20)$$

Ειδική περίπτωση 3 σημείων και πολυωνύμου 2^{ου} βαθμού

Για την περίπτωση όμως που έχουμε 3 σημεία και ψάχνουμε πολυώνυμο 2^{ου} βαθμού, έχουμε

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{bmatrix} \quad (1.21)$$

$$Aa = y \quad (1.22)$$

Βλέπουμε ότι σε αυτή την περίπτωση ο πίνακας $Aa = y$ είναι ήδη τετράγωνος, και άρα αντιστρέψιμος. Έτσι, μπορούμε να πάρουμε απευθείας την παρακάτω λύση

$$a = (A)^{-1} y \quad (1.23)$$

$$p_2(x) = a_0 + a_1x + a_2x^2 \quad (1.24)$$

Ακρότατα πολυωνύμου 2ου βαθμού

Θεωρώντας ένα πολυώνυμο 2^{ου} βαθμού της μορφής

$$f(x) = ax^2 + bx + c \quad (1.25)$$

Με παραγωγή και εξίσωση του αποτελέσματος με το μηδέν παίρνουμε

$$f'(x) = 2ax + b \quad (1.26)$$

$$f'(x) = 0 \Rightarrow 2ax + b = 0 \Rightarrow x = -\frac{b}{2a} \quad (1.27)$$

Η τιμή αυτή αντιστοιχεί στη θέση του τοπικού ακρότατου που αναζητείται.

Ταυτοποιώντας το marker

Μέχρι τώρα είδαμε πως αναγνωρίζουμε την ύπαρξη του κάθε marker σε ένα frame, και πως βρίσκουμε τις ακμές του και τις 4 γωνίες του με ακρίβεια. Με τη βοήθεια αυτών των 4 γωνιών έχουμε τα εφόδια για να ταυτοποιήσουμε το marker. Για να γίνει αυτό, όμως, θα πρέπει να βρούμε έναν τρόπο να αφαιρεθεί η “προοπτική παραμόρφωση” που εισάγεται λόγω της σχετικής θέσης της κάμερας ως προς το marker.

Homogenous Coordinates

Προκειμένου να διευκολυνθούμε, θα χρησιμοποιήσουμε ομογενείς συντεταγμένες [87]. Αυτές απλοποιούν τις πράξεις στο projective space, ακριβώς όπως οι Καρτεσιανές συντεταγμένες στον Ευκλείδειο χώρο, και γι' αυτό χρησιμοποιούνται ευρέως στη γραφική με υπολογιστές. Με τη χρήση τους οι affine ή οι προοπτικοί μετασχηματισμοί μπορούν εύκολα να παρασταθούν απλά με έναν πίνακα.

Έτσι, έχουμε την παρακάτω αντιστοίχιση στην αναπαράσταση σημείων, όπου w ένας αυθαίρετος μη μηδενικός αριθμός⁸².

	Euclidean Geometry	Projective Geometry
Σημείο στον 2D χώρο	$\begin{bmatrix} x \\ y \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$ με $w \neq 0$
Σημείο στον 3D χώρο	$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix}$ με $w \neq 0$

Πίνακας 4 - Παράσταση σημείων με Ομογενείς συντεταγμένες

Στον 2D χώρο, που μας ενδιαφέρει, λοιπόν, εάν w είναι ένας αυθαίρετος μη μηδενικός αριθμός, για να πάρουμε τις ομογενείς συντεταγμένες ενός σημείου από τις αντίστοιχες ευκλείδειες, έχουμε

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} xw \\ yw \\ w \end{bmatrix} \quad (1.28)$$

Συμπεραίνουμε λοιπόν, ότι όλα τα διανύσματα που είναι ανάλογα μεταξύ τους, λόγω διαφορετικής τιμής της μεταβλητής w (με $w \neq 0$), αντιστοιχούν στο ίδιο σημείο.

⁸² Για $w=0$ παίρνουμε τα σημεία $[x' \ y' \ 0]$, τα λεγόμενα points at infinity, που βρίσκονται πάνω στην line at infinity (με την εξαίρεση του σημείου $[0 \ 0 \ 0]$)

Αντίστροφα, για να πάρουμε τις ευκλείδειες συντεταγμένες από τις ομογενείς, έχουμε

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'/w \\ y'/w \end{bmatrix} \quad (1.29)$$

Εξαιτίας αυτής της διαίρεσης που παρατηρείται, παρόλο που μπορούμε να χρησιμοποιήσουμε έναν οποιοδήποτε μη μηδενικό αριθμό w , συνήθως παίρνουμε

$$w = 1 \quad (1.30)$$

ώστε η αλλαγή συστήματος συντεταγμένων να γίνεται πολύ εύκολα.

Έτσι συνήθως χρησιμοποιούμε την παρακάτω αντιστοιχία

	Euclidean Geometry	Projective Geometry
Σημείο στον 2D χώρο	$\begin{bmatrix} x \\ y \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$

Πίνακας 5 - Συνήθης παράσταση 2D σημείου με ομογενείς συντεταγμένες

Projective Mapping

Όπως προαναφέρθηκε, ένας από τους λόγους που χρησιμοποιούνται οι ομογενείς συντεταγμένες είναι γιατί διευκολύνουν την προοπτική προβολή (projective mapping⁸³), δηλαδή την προβολή ενός επιπέδου σε ένα άλλο, μέσω ενός σημείου.

Παρακάτω θα εξετάσουμε την 2D περίπτωση αυτών [87].

	Ομογενείς Συντεταγμένες	Καρτεσιανές Συντεταγμένες
Source	$\hat{p}_s = \begin{bmatrix} u' \\ v' \\ q \end{bmatrix}$	$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u'/q \\ v'/q \end{bmatrix}$ για $q \neq 0$
Destination	$\hat{p}_d = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'/w \\ y'/w \end{bmatrix}$ για $w \neq 0$
$\hat{p}_s = \begin{bmatrix} u' \\ v' \\ q \end{bmatrix} \xrightarrow{\text{mapping}} \hat{p}_d = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$		

Πίνακας 6 - Αλλαγή από Ομογενείς σε Καρτεσιανές συντεταγμένες

⁸³ γνωστή και ως perspective ή homogeneous transformation

“Ευθύ” projective mapping

Οι γενικοί τύποι της προβολής (σε καρτεσιανές συντεταγμένες) είναι οι παρακάτω

$$x = \frac{au + bv + c}{gu + hv + i} \quad y = \frac{du + ev + f}{gu + hv + i} \quad (1.31)$$

Ή πιο απλά σε ομογενείς συντεταγμένες

$$\hat{p}_d = \hat{M}_{sd} \hat{p}_s \quad (1.32)$$

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \cdot \begin{bmatrix} u' \\ v' \\ q \end{bmatrix} \quad (1.33)$$

όπου

$$\left. \begin{array}{l} (1.32) \\ (1.33) \end{array} \right\} \hat{M}_{sd} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (1.34)$$

Από τον τύπο (1.33) παίρνουμε

$$(1.33) \Rightarrow \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} xw \\ yw \\ w \end{bmatrix} = \begin{bmatrix} au'+bv'+cq \\ du'+ev'+fq \\ gu'+hv'+iq \end{bmatrix} \quad (1.35)$$

Όπως σημειώθηκε παραπάνω όμως, όλα τα διανύσματα που είναι μεταξύ τους ανάλογα, λόγω διαφορετικής τιμής της μεταβλητής w (με $w \neq 0$), αντιστοιχούν στο ίδιο σημείο. Έτσι, στον τελευταίο τύπο, χωρίς απώλεια της γενικότητας μπορούμε να θέσουμε $i=1$ αυθαίρετα ⁸⁴

Έτσι, παρόλο που ο πίνακας \hat{M}_{sd} έχει αρχικά 9 μεταβλητές, με τη συνθήκη $i=1$, έχουμε

$$(1.34) \Rightarrow \hat{M}_{sd} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \quad (1.36)$$

δηλαδή ο πίνακας \hat{M}_{sd} έχει πλέον 8 άγνωστες μεταβλητές που καθορίζουν το mapping.

Το 2D mapping χαρακτηρίζεται λοιπόν από 8 βαθμούς ελευθερίας

⁸⁴ εκτός απ' την περίπτωση που το σημείο $[0 \ 0]$ maps to a point at infinity

Εάν επιπλέον $g = 0$ και $h = 0$, τότε η projective transformation είναι affine⁸⁵ (βλ εικόνα 48)

Η προοπτική προβολή έχει τις παρακάτω ιδιότητες:

- Παρόλο που δε διατηρεί σταθερές τις αποστάσεις μεταξύ δύο σημείων, διατηρεί σταθερή την αναλογία των αποστάσεων ανάμεσα σε 2 σημεία (the cross ratio of points)
- Διατηρεί τις γραμμές σε όλες τις διευθύνσεις (όπως και οι affine μετασχηματισμοί)
(Με βάση τις δύο παραπάνω ιδιότητες, αναμένουμε το αποτέλεσμα του mapping ενός τετραπλεύρου να είναι κι αυτό τετράπλευρο - βλ εικόνα 47)
- Η αντίστροφη διαδικασία μιας προοπτικής προβολής είναι και αυτή μία προοπτική προβολή.

⁸⁵ Για την περίπτωση των 2D affine μετασχηματισμών, οι σχέσεις

$$x' = x + t_x \quad \text{και} \quad y' = y + t_y$$

Εκφράζονται ως

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

“Αντίστροφο” projective mapping

Για το αντίστροφο projective mapping (διατηρούμε τους συμβολισμούς που θέσαμε πριν)

$$\hat{p}_s = \begin{bmatrix} u' \\ v' \\ q \end{bmatrix} \xleftarrow{\text{mapping}} \hat{p}_d = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}$$

χρησιμοποιούμε είτε το αντίστροφο είτε το adjoint matrix του M_{sd} . Κι αυτό γιατί χρησιμοποιούμε ομογενείς συντεταγμένες, ενώ ο αντίστροφος του M_{sd} συνδέεται με το adjoint με τη σχέση

$$\hat{M}_{sd}^{-1} = \frac{\text{adj}(\hat{M}_{sd})}{\det(\hat{M}_{sd})} = \frac{1}{\det(\hat{M}_{sd})} \text{adj}(\hat{M}_{sd}) \quad (1.37)$$

συνδέονται δηλαδή με μία σχέση αναλογίας.

Ο αντίστροφος όμως δεν ορίζεται για μοναδιαίο πίνακα, ενώ ο adjoint ορίζεται πάντα. Έτσι, ο adjoint προτιμάται στους υπολογισμούς όπου γίνεται χρήση ομογενών συντεταγμένων.

Για το αντίστροφο mapping έχουμε λοιπόν

$$(1.32) \Rightarrow \hat{p}_s = \hat{M}_{ds} \hat{p}_d \quad (1.38)$$

$$(1.33) \Rightarrow \begin{bmatrix} u' \\ v' \\ q \end{bmatrix} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} \quad (1.39)$$

όπου

$$\hat{M}_{ds} = \text{adj}(\hat{M}_{sd}) \quad (1.40)$$

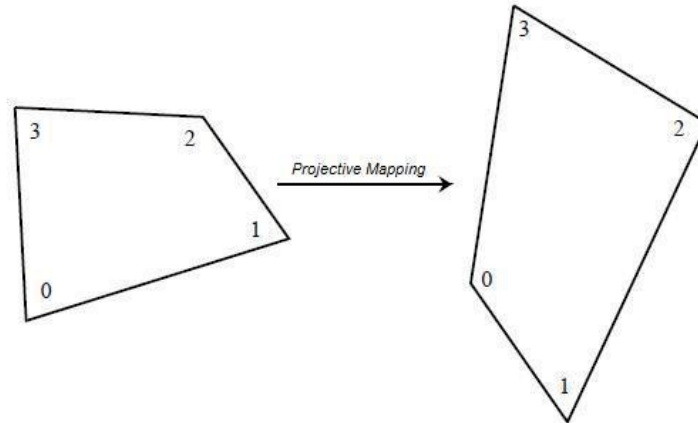
$$\left. \begin{array}{l} (1.38) \\ (1.39) \end{array} \right\} \hat{M}_{ds} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} = \begin{bmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{bmatrix} \quad (1.41)$$

Σε καρτεσιανές συντεταγμένες έχουμε

$$u = \frac{Ax + By + C}{Gx + Hy + I} \quad v = \frac{Dx + Ey + F}{Gx + Hy + I} \quad (1.42)$$

Γενική μέθοδος 2D Projective Mapping

Στην παρακάτω εικόνα φαίνεται παραστατικά η γενική περίπτωση του 2D projective mapping, όσον αφορά στο mapping τετραπλεύρων. Όπως προαναφέρθηκε, με mapping ενός τετραπλεύρου καταλήγουμε και πάλι σε τετράπλευρο (βλ εικόνα 47).



Εικόνα 47 - Projective mapping

Για την αντιστοίχιση

$$[u_k \quad v_k] \xrightarrow{\text{mapping}} [x_k \quad y_k] \quad (1.43)$$

όπου το k δεικτοδοτεί την αντιστοίχιση του ζεύγους σημείων, έχουμε τις ακόλουθες σχέσεις (σε καρτεσιανές συντεταγμένες) [87]

$$(1.31) \Rightarrow x_k = \frac{au_k + bv_k + c}{gu_k + hv_k + 1} \Rightarrow \quad (1.44)$$

$$x_k = au_k + bv_k + c - gu_k x_k - hv_k x_k$$

$$(1.31) \Rightarrow y_k = \frac{du_k + ev_k + f}{gu_k + hv_k + 1} \Rightarrow \quad (1.45)$$

$$y_k = du_k + ev_k + f - gu_k y_k - hv_k y_k$$

Δηλαδή κάθε ζεύγος αντιστοίχισης σημείων δίνει 2 εξισώσεις.

Σε ομογενείς συντεταγμένες, θα χρησιμοποιήσουμε την παρακάτω εξίσωση

$$(1.32) \Rightarrow \hat{p}_d = \hat{M}_{sd} \hat{p}_s \quad (1.46)$$

$$(1.33) \Rightarrow \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \cdot \begin{bmatrix} u' \\ v' \\ q \end{bmatrix} \quad (1.47)$$

όπου θέσαμε $i=1$, όπως εξηγήθηκε παραπάνω. Έτσι, το projective mapping χαρακτηρίζεται από 8 βαθμούς ελευθερίας, και ο πίνακας \hat{M}_{sd} έχει 8 άγνωστες μεταβλητές, που μπορούν να βρεθούν από ένα σύστημα 8 εξισώσεων.

Τις εξισώσεις αυτές τις παίρνουμε από 4 αντιστοιχίσεις σημείων (αφού το κάθε ζεύγος δίνει 2 εξισώσεις), δηλαδή από τις αντιστοιχίσεις των 4 γωνιών του τετραπλεύρου.

Έτσι, από τις 4 αντιστοιχίες γωνιών του τετραπλεύρου

$$\begin{array}{l} [u_0 \quad v_0] \xrightarrow{\text{mapping}} [x_0 \quad y_0] \\ [u_1 \quad v_1] \xrightarrow{\text{mapping}} [x_1 \quad y_1] \\ [u_2 \quad v_2] \xrightarrow{\text{mapping}} [x_2 \quad y_2] \\ [u_3 \quad v_3] \xrightarrow{\text{mapping}} [x_3 \quad y_3] \end{array}$$

Πίνακας 7 - Αντιστοίχιση σημείων για projective mapping

έχουμε (από τις σχέσεις 1.44, 1.45) τις παρακάτω 8 εξισώσεις (2 για κάθε αντιστοιχία γωνίας)

	x	y
k=0	$x_0 = au_0 + bv_0 + c - gu_0x_0 - hv_0y_0$	$y_0 = du_0 + ev_0 + f - gu_0y_0 - hv_0x_0$
k=1	$x_1 = au_1 + bv_1 + c - gu_1x_1 - hv_1y_1$	$y_1 = du_1 + ev_1 + f - gu_1y_1 - hv_1x_1$
k=2	$x_2 = au_2 + bv_2 + c - gu_2x_2 - hv_2y_2$	$y_2 = du_2 + ev_2 + f - gu_2y_2 - hv_2x_2$
k=3	$x_3 = au_3 + bv_3 + c - gu_3x_3 - hv_3y_3$	$y_3 = du_3 + ev_3 + f - gu_3y_3 - hv_3x_3$

Πίνακας 8 - Οι εξισώσεις του projective mapping

Και σε μορφή πινάκων

$$\begin{bmatrix} u_0 & v_0 & 1 & 0 & 0 & 0 & -u_0x_0 & -v_0x_0 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1x_1 \\ u_2 & v_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -v_2x_2 \\ u_3 & v_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -v_3x_3 \\ 0 & 0 & 0 & u_0 & v_0 & 1 & -u_0y_0 & -v_0y_0 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1y_1 & -v_1y_1 \\ 0 & 0 & 0 & u_2 & v_2 & 1 & -u_2y_2 & -v_2y_2 \\ 0 & 0 & 0 & u_3 & v_3 & 1 & -u_3y_3 & -v_3y_3 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (1.48)$$

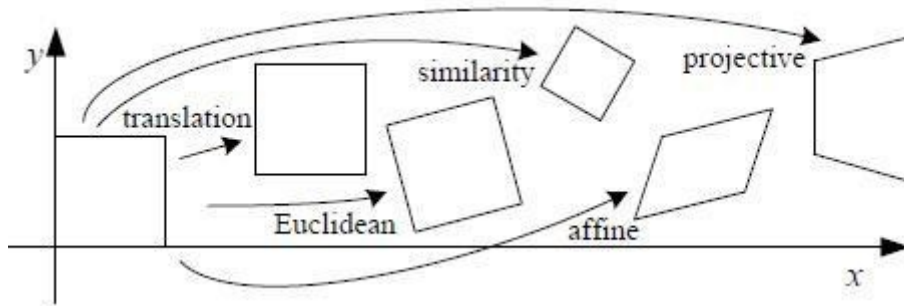
Λύνοντας το παραπάνω σύστημα, βρίσκουμε τους 8 άγνωστους συντελεστές του πίνακα \hat{M}_{sd} (επίσης $i=1$), που ορίζει το forward mapping.

Αφού βρεθεί ο \hat{M}_{sd} , μπορούμε να βρούμε τον πίνακα \hat{M}_{ds} του inverse mapping είτε ακολουθώντας ανάλογη διαδικασία και λύνοντας και πάλι ένα 8x8 σύστημα είτε απλούστερα από τη σχέση

$$\hat{M}_{ds} = adj(\hat{M}_{sd}) \quad (1.49)$$

Όταν δεν υπάρχει ανάγκη γρήγορης εκτέλεσης των υπολογισμών, το σύστημα μπορεί να λυθεί με κάποια κλασική μέθοδο, πχ Gaussian Elimination.

Όταν όμως υπάρχει ανάγκη γρήγορων υπολογισμών, όπως σε εφαρμογές πραγματικού χρόνου (πχ μία εφαρμογή AR), μπορούμε να λύσουμε το σύστημα όπως παρουσιάζεται στην επόμενη ενότητα.



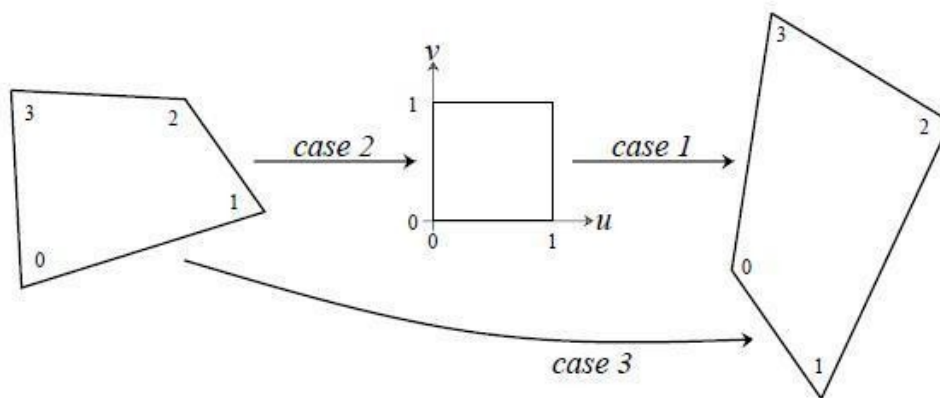
Εικόνα 48 - Διάφοροι τύποι planar mapping

Ειδικές μέθοδοι 2D Projective Mapping

Στην παρακάτω εικόνα φαίνονται παραστατικά οι ειδικές περιπτώσεις του 2D projective mapping, όσον αφορά στο mapping τετραπλεύρων.

Διακρίνονται 3 περιπτώσεις [87]:

- Περίπτωση 1: Mapping από ένα μοναδιαίο τετράγωνο⁸⁶ σε ένα τετράπλευρο
- Περίπτωση 2: Mapping από ένα τετράπλευρο σε ένα μοναδιαίο τετράγωνο
- Περίπτωση 3: Mapping από ένα τετράπλευρο σε ένα άλλο τετράπλευρο



Εικόνα 49 - Οι 3 κύριες περιπτώσεις κατά το projective mapping

Σε real time εφαρμογές η γρήγορη εύρεση του πίνακα του mapping είναι επιτακτικής σημασίας, γι αυτό για κάθε ειδική περίπτωση ακολουθούμε την κατάλληλη μέθοδο, όπως παρουσιάζεται παρακάτω.

Περίπτωση 1 - Από μοναδιαίο τετράγωνο σε τετράπλευρο

Θεωρώντας όμως σαν “αρχή” ένα μοναδιαίο τετράγωνο (με το ένα άκρο στην αρχή των αξόνων), οι εξισώσεις της γενική λύσης (εξίσωση 1.48) απλοποιούνται όπως φαίνονται παρακάτω

	x	y	u	v
Σημείο 0	x_0	y_0	0	0
Σημείο 1	x_1	y_1	1	0
Σημείο 2	x_2	y_2	1	1
Σημείο 3	x_3	y_3	0	1

Πίνακας 9 - Αντιστοίχιση σημείων: από μοναδιαίο τετράγωνο σε τετράπλευρο

$$\begin{aligned}
 c &= x_0 \\
 a + c - gx_1 &= x_1 \\
 a + b + c - gx_2 - hx_2 &= x_2 \\
 b + c - hx_3 &= x_3
 \end{aligned}
 \tag{1.50}$$

⁸⁶ Με το ένα άκρο του στην αρχή των αξόνων

$$\begin{aligned}
 f &= y_0 \\
 d + f - gy_1 &= y_1 \\
 d + e + f - gy_2 - hy_2 &= y_2 \\
 e + f - hy_3 &= y_3
 \end{aligned}$$

Θεωρώντας επιπλέον

$$\begin{aligned}
 \Delta x_1 &= x_1 - x_2 \\
 \Delta x_2 &= x_3 - x_2 \\
 \Sigma x &= x_0 - x_1 + x_2 - x_3 \\
 \Delta y_1 &= y_1 - y_2 \\
 \Delta y_2 &= y_3 - y_2 \\
 \Sigma y &= y_0 - y_1 + y_2 - y_3
 \end{aligned} \tag{1.51}$$

έχουμε τις παρακάτω υποπεριπτώσεις

- Εάν $\Sigma x = 0$ και $\Sigma y = 0$, τότε το mapping είναι affine (το τετράπλευρο-“πηγή” είναι παραλληλόγραμμο⁸⁷), και έχουμε

$$\begin{aligned}
 a &= x_1 - x_0 & d &= y_1 - y_0 \\
 b &= x_2 - x_1 & e &= y_2 - y_1 \\
 c &= x_0 & f &= y_0 \\
 g &= 0 & h &= 0
 \end{aligned} \tag{1.52}$$

- Εάν $\Sigma x \neq 0$ ή $\Sigma y \neq 0$, τότε

$$\begin{aligned}
 g &= \frac{\begin{vmatrix} \Sigma x & \Delta x_2 \\ \Sigma y & \Delta y_2 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} & h &= \frac{\begin{vmatrix} \Delta x_1 & \Sigma x \\ \Delta y_1 & \Sigma y \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} \\
 a &= x_1 - x_0 + gx_1 & d &= y_1 - y_0 + gy_1 \\
 b &= x_3 - x_0 + hx_3 & e &= y_3 - y_0 + hy_3 \\
 c &= x_0 & f &= y_0
 \end{aligned} \tag{1.53}$$

⁸⁷

$\Sigma x = x_0 - x_1 + x_2 - x_3 = (x_2 - x_3) - (x_1 - x_0) = x_{23} - x_{10}$

$\Sigma y = y_0 - y_1 + y_2 - y_3 = (y_2 - y_3) - (y_1 - y_0) = y_{23} - y_{10}$

$\Sigma x = 0 \Rightarrow x_{23} = x_{10}$

$\Sigma y = 0 \Rightarrow y_{23} = y_{10}$

Περίπτωση 2 - Από τετράπλευρο σε μοναδιαίο τετράγωνο

Σε αυτή την περίπτωση, βρίσκουμε το ζητούμενο πίνακα με έμμεσο τρόπο. Αρχικά βρίσκουμε τον πίνακα του inverse mapping, δηλαδή από μοναδιαίο τετράγωνο σε τετράπλευρο (βλ περίπτωση 1), και έπειτα παίρνουμε το ζητούμενο πίνακα (του forward mapping) από τη σχέση

$$\hat{M}_{forward} = adj(\hat{M}_{reverse}) \quad (1.54)$$

Ή με άλλα λόγια

$$\hat{M}_{περίπτωση2} = adj(\hat{M}_{περίπτωση1}) \quad (1.55)$$

Περίπτωση 3 - Από τετράπλευρο σε τετράπλευρο

Σε αυτή την περίπτωση, μπορούμε να ακολουθήσουμε τα δύο ενδιάμεσα βήματα των περιπτώσεων 1 και 2

Η λογική των δύο βημάτων φαίνεται και παραστατικά στην εικόνα 49.

Έτσι, από την παρακάτω σχέση⁸⁸ παίρνουμε τον πίνακα της μελετώμενης περίπτωσης.

⁸⁸ Από την εικόνα 49 έχουμε τις σχέσεις

$$\hat{P}_{κατίστασηΔεξιά} = \hat{M}_{περίπτωση3} \cdot \hat{P}_{κατίστασηΑριστερά} \quad a$$

$$\hat{P}_{κατίστασηΔεξιά} = \hat{M}_{περίπτωση1} \cdot \hat{P}_{κατίστασηΜεσαία} \quad b$$

$$\hat{P}_{κατίστασηΜεσαία} = \hat{M}_{περίπτωση2} \cdot \hat{P}_{κατίστασηΑριστερά} \quad c$$

$$b \Rightarrow \hat{P}_{κατίστασηΔεξιά} = \hat{M}_{περίπτωση1} \cdot \hat{P}_{κατίστασηΜεσαία}$$

$$\stackrel{c}{\Rightarrow} \hat{P}_{κατίστασηΔεξιά} = \hat{M}_{περίπτωση1} \cdot \hat{M}_{περίπτωση2} \cdot \hat{P}_{κατίστασηΑριστερά} \quad d$$

Και τελικά

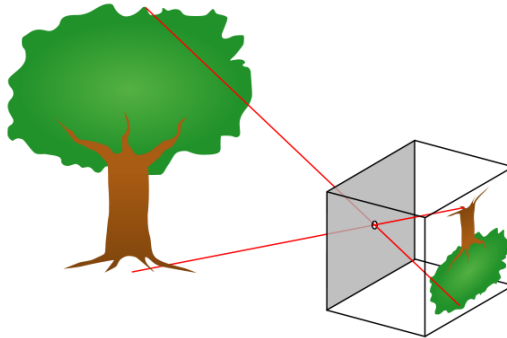
$$\left. \begin{matrix} a \\ d \end{matrix} \right\} \Rightarrow \hat{M}_{περίπτωση3} = \hat{M}_{περίπτωση1} \cdot \hat{M}_{περίπτωση2}$$

Μοντελοποίηση της κάμερας

Σαν μοντέλο κάμερας στην παρούσα εργασία επιλέγεται το pinhole μοντέλο, επαυξημένο με μερικές παραμέτρους που μοντελοποιούν ορισμένες παραμορφώσεις.

Απλό Pinhole μοντέλο

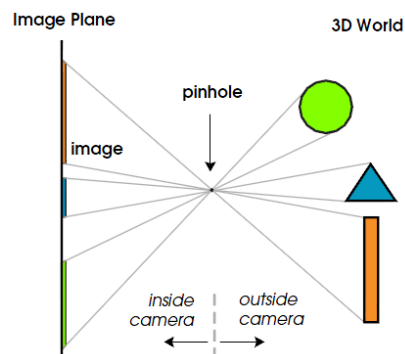
Το pinhole μοντέλο είναι το απλούστερο μοντέλο κάμερας, το οποίο γίνεται μάλιστα και εύκολα κατανοητό μέσω του παραδείγματος της pinhole φωτογραφικής μηχανής⁸⁹ [88] που μπορεί πολύ εύκολα να κατασκευάσει κάποιος [89].



Εικόνα 50 - Pinhole φωτογραφική μηχανή

Όπως βλέπουμε στην παραπάνω εικόνα, θεωρώντας την κάμερα σαν ένα κουτί με μία μικρή τρύπα στην “μπροστινή” της επιφάνεια (χωρίς την ύπαρξη φακού), έχουμε τη δημιουργία μίας πολύ απλής φωτογραφικής μηχανής. Οι ακτίνες φωτός, αφού προσπέσουν πάνω σε ένα αντικείμενο, περνάνε μέσα από την τρύπα της pinhole κάμερας και καταλήγουν στην πίσω επιφάνεια του κουτιού, όπου και απεικονίζεται η τελική φωτογραφία, η οποία απεικονίζει τον πραγματικό κόσμο ανεστραμμένο.

Κατά τη φωτογράφιση έχουμε μία αποτύπωση του τρισδιάστατου περιβάλλοντος χώρου σε μία δυσδιάστατη εικόνα μέσω του φωτός, χάνοντας, μάλιστα, με τον τρόπο αυτό πληροφορία από την πραγματική σκηνή (πχ δε διατηρούνται τα σχήματα, ενώ διατηρούνται οι γραμμές). Μία αναπαράσταση αυτής της αποτύπωσης-προβολής έχουμε στην εικόνα που ακολουθεί [90].

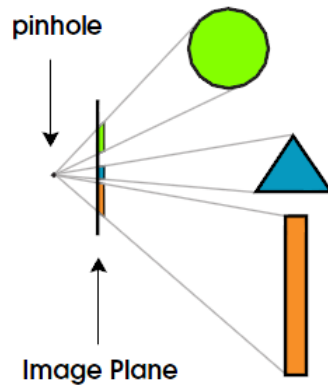


Εικόνα 51 - Λειτουργία του pinhole μοντέλου

⁸⁹ γνωστή και σαν “camera obscura”

Όπως βλέπουμε (εικόνα 51) δεξιά έχουμε τον πραγματικό τρισδιάστατο κόσμο, αριστερά έχουμε το επίπεδο της δυσδιάστατης εικόνας-αποτύπωσης, ενώ στη μέση έχουμε την pinhole σαν “διαχωριστικό σημείο”. Το επίπεδο της εικόνας-αποτύπωσης βρίσκεται λοιπόν στην πραγματικότητα πίσω από την pinhole, σε μία συγκεκριμένη απόσταση από αυτό.

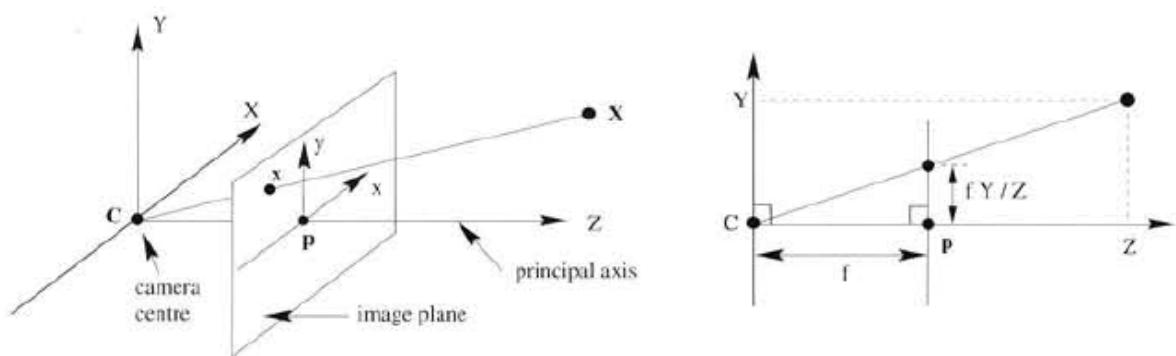
Μετατοπίζοντας, όμως, νοητά το επίπεδο της εικόνας μπροστά από την pinhole [90], διατηρώντας την ίδια απόσταση μεταξύ τους, έχουμε ένα αποτέλεσμα που μοιάζει με την παρακάτω εικόνα



Εικόνα 52 - Απλούστευση του pinhole μοντέλου

Όπως βλέπουμε, με αυτή τη νοητή μεταφορά έχουμε μία απλούστερη και πιο κατανοητή αναπαράσταση της διαδικασίας της φωτογράφισης, καθώς πλέον δεν έχουμε ανεστραμμένη αποτύπωση των πραγματικών αντικειμένων στην εικόνα. Για λόγους απλούστευσης, χωρίς απώλεια της γενικότητας, θα διατηρηθεί αυτή η σύμβαση στο υπόλοιπο της εργασίας.

Μία πιο αυστηρή αναπαράσταση του pinhole μοντέλου απεικονίζεται στην παρακάτω εικόνα [91].



Εικόνα 53 - Αναλυτικό διάγραμμα του pinhole μοντέλου

Το σημείο pinhole, που παίζει και τον κεντρικό ρόλο στην όλη διαδικασία, ονομάζεται και camera centre (ή optical centre), ενώ το επίπεδο που περνάει από αυτό και είναι παράλληλο στο επίπεδο της εικόνας-αποτύπωσης, δηλαδή το image plane, ονομάζεται principal plane(επίπεδο XCY). Ο άξονας που ξεκινάει από το camera centre και διαπερνά

κάθετα το image plane ονομάζεται principal axis (ή principal ray), ενώ το σημείο τομής με το image plane ονομάζεται principal point.

Όπως φαίνεται [91], το image plane παριστάνεται με το επίπεδο

$$Z = f \quad (1.56)$$

όπου f είναι η εστιακή απόσταση, ενώ η αποτύπωση-προβολή ενός πραγματικού σημείου σε αυτό γίνεται μέσω της τομής του ευθύγραμμου τμήματος XC με το επίπεδο του image plane.

Εάν τα σημεία του τρισδιάστατου πραγματικού χώρου παριστάνονται ως

$$\hat{X} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1.57)$$

και τα σημεία του Image plane παριστάνονται ως

$$\hat{x} = \begin{bmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{bmatrix} \quad (1.58)$$

τότε κατά την προβολή του τρισδιάστατου χώρου στο image plane έχουμε την αντιστοίχιση

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} f \frac{X}{Z} \\ f \frac{Y}{Z} \end{bmatrix} \quad (1.59)$$

Χρησιμοποιώντας ομογενείς συντεταγμένες, που περιγράφηκαν στην αντίστοιχη υποπαράγραφο, η αντιστοίχιση σημείων παίρνει τη μορφή

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & & 0 \\ & f & 0 \\ & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.60)$$

Θέτοντας

$$\hat{X} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.61)$$

Και συμβολίζοντας με \hat{x} το σημείο στο image plain, παίρνουμε την παρακάτω κομψότερη αναπαράσταση της αντιστοίχισης των σημείων

$$\hat{x} = P\hat{X} \quad (1.62)$$

όπου ο πίνακας P , μεγέθους 3×4 , ονομάζεται και camera projection matrix

$$P = \begin{bmatrix} f & & 0 \\ & f & 0 \\ & & 1 & 0 \end{bmatrix} \quad (1.63)$$

Σχετικά με το principal point

Στα προηγούμενα θεωρήθηκε ότι το σύστημα συντεταγμένων του image plane έχει την αρχή του στο principal point, κάτι που δεν ισχύει απαραίτητα. Έτσι, πρέπει να τροποποιηθεί το παραπάνω μοντέλο ώστε να ληφθεί υπόψη αυτή η “μετατόπιση” του συστήματος συντεταγμένων [91]:

Εάν

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad (1.64)$$

είναι οι συντεταγμένες του principal point, έχουμε

$$(1.59) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} f \frac{X}{Z} + p_x \\ f \frac{Y}{Z} + p_y \end{bmatrix} \quad (1.65)$$

$$(1.60) \Rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & p_x & 0 \\ & f & p_y \\ & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.66)$$

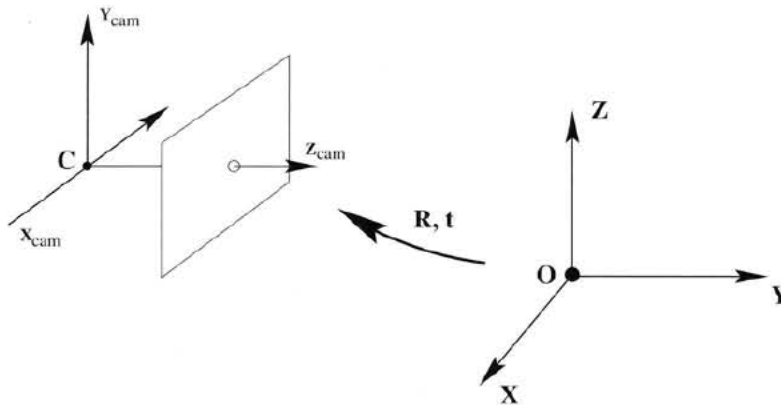
$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \quad (1.67)$$

όπου ο 3×3 πίνακας K ονομάζεται camera calibration matrix.

Σχετικά με τα συστήματα συντεταγμένων

Κατά την προβολή του 3d κόσμου σε μία 2d εικόνα διαχειριζόμαστε σημεία που ανήκουν σε δύο διαφορετικά συστήματα συντεταγμένων (βλ εικόνα 54):

- Το σύστημα συντεταγμένων της κάμερας
- Το σύστημα συντεταγμένων του κόσμου - WCS (World Coordinate)



Εικόνα 54 - Το σύστημα συντεταγμένων της κάμερας και το WCS (World Coordinate System)

Ένα σημείο μπορεί να εκφραστεί με την κατάλληλη μορφή σε οποιοδήποτε από αυτά τα δύο συστήματα συντεταγμένων.

Έτσι, εάν

- \tilde{X} το ομογενές διάνυσμα ενός σημείου στο WCS και
- \tilde{X}_{cam} το ομογενές διάνυσμα του ίδιου σημείου στο σύστημα συντεταγμένων της κάμερας
- \tilde{C} οι συντεταγμένες του camera centre στο WCS
- R ο 3x3 πίνακας rotation

Έχουμε [91]

$$\tilde{X}_{cam} = R(\tilde{X} - \tilde{C}) \quad (1.68)$$

Ενώ αν θέσουμε

$$t = -R\tilde{C} \quad (1.69)$$

Έχουμε

$$\tilde{X}_{cam} = R\tilde{X} + t \quad (1.70)$$

Και ο πίνακας της προβολής είναι

$$P = K[R | t] \quad (1.71)$$

Ο πίνακας rotation R περιγράφει τον προσανατολισμό του συστήματος συντεταγμένων της κάμερας. Η “συνολική” περιστροφή που περιγράφει αυτός ο 3×3 πίνακας εκτελείται μέσω των στοιχειωδών περιστροφών που ορίζουν οι τρεις στήλες του [92], με την παρακάτω σειρά:

- η πρώτη στήλη περιστρέφει το αρχικό σύστημα συντ/νων ως προς τον άξονα x
- η δεύτερη περιστρέφει το (ήδη περιστρεμμένο) σύστημα ως προς τον άξονα y
- η τρίτη περιστρέφει το (ήδη 2 φορές περιστρεμμένο) σύστημα ως προς τον άξονα z .

Σχετικά με τον αισθητήρα της κάμερας

Πριν φτάσει η κατασκευή των αισθητήρων στο σημερινό επίπεδο, ο αισθητήρας εισήγαγε κι αυτός ορισμένες παραμορφώσεις στο μοντέλο της κάμερας.

Έτσι, αν

- s (skew) είναι η γωνία των δύο πλευρών του κάθε pixel του αισθητήρα (που μπορεί με αυτό τον τρόπο να μην έχουν ορθογώνιο σχήμα αλλά απλά παραλληλόγραμμο), και
- το εστιακό μήκος (focal length) δίνεται, λόγω των άνισων πλευρών του κάθε pixel από τους τύπους

$$f_x = f \cdot m_x \quad (1.72)$$

$$f_y = f \cdot m_y \quad (1.73)$$

τότε ο πίνακας Calibration matrix δίνεται πλέον από τη σχέση [91]

$$(1.67) \Rightarrow K = \begin{bmatrix} f_x & s & x_0 \\ & f_y & y_0 \\ & & 1 \end{bmatrix} \quad (1.74)$$

Πρέπει να τονιστεί όμως ότι με το σημερινό επίπεδο κατασκευής αισθητήρων, μπορεί να θεωρηθεί με ασφάλεια ότι τα pixels του αισθητήρα είναι ορθογώνια, οπότε το s παίρνει μηδενική τιμή τις περισσότερες φορές.

Σχετικά με το φακό

Προσθέτοντας στο μοντέλο της κάμερας το φακό που έχουν όλες οι κάμερες, εισάγονται αυτόματα οι παραμορφώσεις των οπτικών στοιχείων. Ανάλογα με την ποιότητα της κάμερας διαφέρει, όπως είναι φυσικό, και το μέγεθος των παραμορφώσεων.

Δυστυχώς, όμως, τα κινητά τηλέφωνα με ενσωματωμένη κάμερα, που ενδιαφέρουν πολύ τους τομείς του pervasive computing και της augmented reality, ενώ φέρουν αισθητήρες αρκετά καλής ποιότητας και ολοένα περισσότερων megapixels (κυρίως για λόγους marketing), συνοδεύονται από οπτικά στοιχεία με έντονες παραμορφώσεις. Είναι σημαντικό λοιπόν το μοντέλο της κάμερας που επιλέγεται για μία εφαρμογή αυτού του είδους σε κινητά να μοντελοποιεί τις εν λόγω παραμορφώσεις.

Αν και υπάρχουν αρκετές μέθοδοι για αυτό το πρόβλημα, θα παρουσιάσουμε τη μέθοδο που ακολουθείται στο Camera calibration toolbox for Matlab [80].

Συμβολίζουμε με

$$f_{xy} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} \quad \text{το εστιακό μήκος}$$

$$p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \quad \text{το principal point}$$

$$s \quad \text{το skew}$$

$$k_c \quad \text{ένα } 5 \times 1 \text{ διάνυσμα που μοντελοποιεί τις παραμορφώσεις (radial και tangential)}$$

Εάν ένα σημείο του χώρου παριστάνεται, ως προς το σύστημα συντεταγμένων της κάμερας, ως

$$\hat{X}_{cam} = \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \quad (1.75)$$

η προβολή του στο image plane θα είναι το κανονικοποιημένο διάνυσμα

$$x_n = \begin{bmatrix} X_{cam} / Z_{cam} \\ Y_{cam} / Z_{cam} \end{bmatrix} \quad (1.76)$$

Θέτοντας

$$r^2 = x^2 + y^2 \quad (1.77)$$

το παραπάνω κανονικοποιημένο διάνυσμα παίρνει την παρακάτω μορφή [93], που συμπεριλαμβάνει την επίδραση των παραμορφώσεων

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = (1 + k_c(1) \cdot r^2 + k_c(2) \cdot r^4 + k_c(5) \cdot r^6) x_n + d_x \quad (1.78)$$

όπου

$$x_d = \begin{bmatrix} 2k_c(3) \cdot xy + k_c(4) \cdot (r^2 + 2x^2) \\ 2k_c(4) \cdot xy + k_c(3) \cdot (r^2 + 2y^2) \end{bmatrix} \quad (1.79)$$

Έχοντας μοντελοποιήσει λοιπόν και τις παραμορφώσεις, η αντιστοίχιση που περιγράφεται σε αυτή την υποπαράγραφο παίρνει πλέον τη μορφή

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} x_d(1) \\ y_d(2) \\ 1 \end{bmatrix} \quad (1.80)$$

με

$$(1.74) \Rightarrow K = \begin{bmatrix} f_x & s \cdot f_x & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1.81)$$

Θα πρέπει να τονιστεί όμως ότι συνήθως θεωρούμε

- $s = 0$, θεωρώντας με ασφάλεια ότι έχουμε ορθογώνια pixels αισθητήρα
- $k_c(5) = 0$, στη συνήθη περίπτωση που χρησιμοποιούμε φακούς περίπου κανονικής⁹⁰ εστιακής απόστασης (και όχι αρκετά ευρυγώνιους φακούς)

Σε περίπτωση που η κάμερα μοντελοποιείται χωρίς παραμορφώσεις από το φακό, η σχέση (1.80) γράφεται

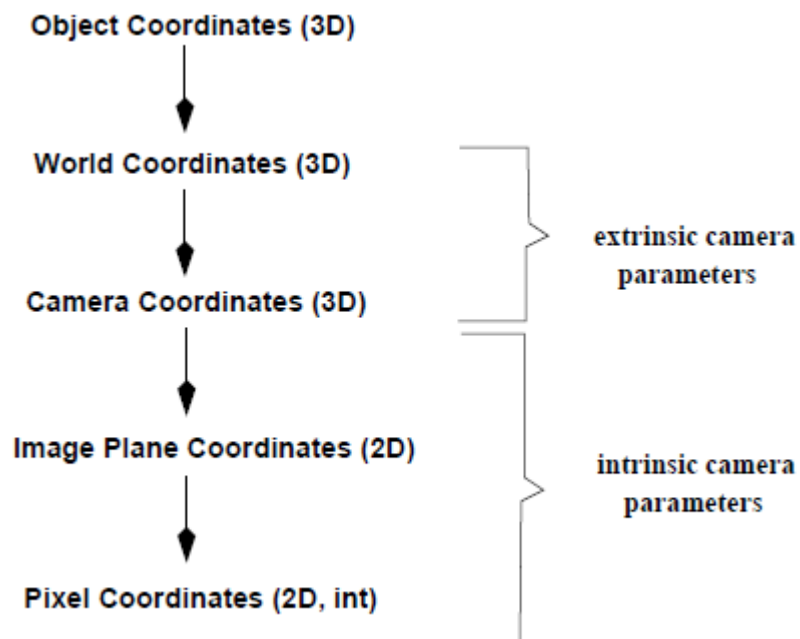
$$(1.80) \Rightarrow \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} x_n(1) \\ y_n(2) \\ 1 \end{bmatrix} \quad (1.82)$$

⁹⁰ Ως κανονική ορίζεται μία εστιακή απόσταση που είναι ισοδύναμη με τα 50mm στο κλασικό format του φιλμ 35mm, η οποία προσεγγίζει περισσότερο τον τρόπο με τον οποίο αντιλαμβάνεται ο άνθρωπος το περιβάλλον του

Παράμετροι της κάμερας και εύρεσή τους

Οι διάφορες παράμετροι της κάμερας, που είδαμε ως τώρα, μπορούν να χωριστούν σε δύο κατηγορίες [93]:

- Εσωτερικοί παράμετροι (intrinsic camera parameters), που ορίζουν την αντιστοίχιση των συντεταγμένων των σημείων στο σύστημα συντεταγμένων της εικόνας (2D) και στο σύστημα συντεταγμένων της κάμερας (3D)
 - Το εστιακό μήκος
 - Η θέση του principal point
 - Ο συντελεστής s σχετικά με το σχήμα των pixels του αισθητήρα
 - Οι παραμορφώσεις λόγω του φακού
- Εξωτερικοί παράμετροι (extrinsic camera parameters), που ορίζουν τη θέση και τον προσανατολισμό της κάμερας (ή ακριβέστερα του συστήματος συντεταγμένων της κάμερας σε σχέση με το WCS⁹¹ της σκηνής)
 - Η περιστροφή (rotation) και
 - Η translation της κάμερας



Εικόνα 55 - Η σχέση των διαφόρων συστημάτων συντεταγμένων με τις παραμέτρους της κάμερας

Ο τρόπος εύρεσης των παραμέτρων αυτών παρουσιάζεται παρακάτω.

⁹¹ World Coordinate System

Camera calibration

Η εύρεση των εσωτερικών παραμέτρων της κάμερας είναι γνωστή ως camera calibration ή camera resectioning. Παρόλο που το αντικείμενο παρουσιάζει ιδιαίτερο ερευνητικό ενδιαφέρον, δεν είναι στους στόχους της εργασίας μία αναλυτική παρουσίαση. Εν γένει, όμως, θα μπορούσε να αναφερθεί ότι οι διάφορες μέθοδοι χωρίζονται σε 2 κύριες κατηγορίες [94]:

- Photogrammetric calibration, στις οποίες έχουμε calibration μέσω της παρατήρησης ενός αντικειμένου, η τρισδιάστατη γεωμετρία του οποίου είναι εκ των προτέρων γνωστή με ακρίβεια. Τα αποτελέσματα είναι συνήθως πολύ ακριβή, αποκτώνται όμως με το κόστος του επιπλέον (και ακριβού σε ειδικές περιπτώσεις) εξοπλισμού.
- Self-calibration, στις οποίες έχουμε calibration χωρίς της χρήση κάποιου ειδικού αντικειμένου. Απλά με την παρατήρηση μιας οποιασδήποτε σκηνής, ακόμα και 3 εικόνες είναι αρκετές για τη διενέργεια των απαραίτητων υπολογισμών. Παρόλα αυτά οι μέθοδοι αυτές δεν είναι ακόμα αρκετά ώριμες, και εξαιτίας του πλήθους των μεταβλητών που πρέπει να βρεθούν πολλές φορές παίρνουμε αναξιόπιστα αποτελέσματα.

Παρατηρούνται, όμως, και μερικές μέθοδοι που δεν εμπίπτουν στις παραπάνω κατηγορίες, όπως αυτές με χρήση vanishing points ή απλής περιστροφής κτλ

Στα πλαίσια της εργασίας απαιτείται μία απλή μέθοδος, με χρήση συνηθισμένου εξοπλισμού και κάποιου φθηνού αντικειμένου με γνωστή γεωμετρία για την calibration.

Καταφεύγουμε λοιπόν, όπως προαναφέρθηκε, στη λύση του Camera calibration toolbox for Matlab [80], που προσφέρει μία έτοιμη και αρκετά αποτελεσματική λύση, με χρήση μιας απλής κάμερας και μιας απλής επίπεδης εικόνας εν είδει ασπρόμαυρης σκακιέρας, που εκτυπώνεται με ένα συνηθισμένο εκτυπωτή.

Pose estimation

Η εύρεση των εξωτερικών παραμέτρων (με δεδομένη την εκ των προτέρων γνώση των εσωτερικών παραμέτρων) που δίνουν τη θέση και τον προσανατολισμό της κάμερας σε σχέση με τον πραγματικό τρισδιάστατο κόσμο, είναι γνωστή σαν pose estimation.

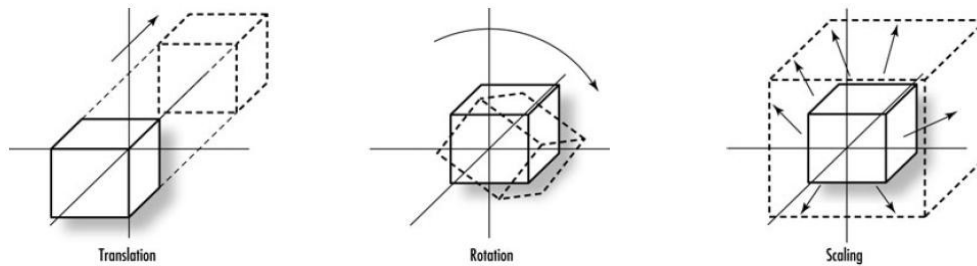
Η σημασία της στην augmented reality είναι πολύ μεγάλη, ιδιαίτερα στην περίπτωση που είναι επιθυμητή η επικάλυψη τρισδιάστατων γραφικών σε μία εικόνα-όψη του πραγματικού κόσμου. Βρίσκοντας τη θέση και τον προσανατολισμό της πραγματικής κάμερας, μπορούμε να “ευθυγραμμίσουμε” με αυτή την εικονική κάμερα του αντίστοιχου τρισδιάστατου μοντέλου. Σαν αποτέλεσμα έχουμε την εμφάνιση της σωστής όψης του τρισδιάστατου μοντέλου, στο κατάλληλο μέγεθος και στο κατάλληλο μέρος.

Οι εξωτερικές παράμετροι της κάμερας που αναζητούνται κατά την pose estimation είναι δύο πίνακες (βλ εικόνα 56 για τη σημασία τους):

- Ο πίνακας R (rotation), μεγέθους 3×3 , που αποτελείται από 3 διανύσματα στοιχειωδών περιστροφών $R = [R_1 \ R_2 \ R_3]$. Τα διανύσματα αυτά

αντιπροσωπεύουν, όπως προαναφέρθηκε, την περιστροφή της κάμερας γύρω από τους 3 άξονες ενός συστήματος συντεταγμένων (3 βαθμοί ελευθερίας για την rotation)

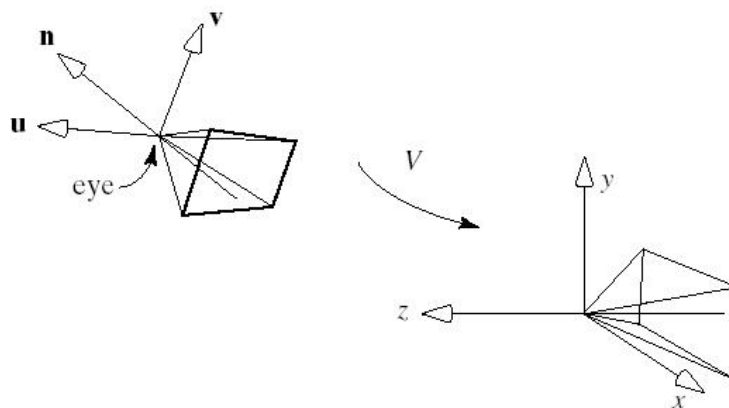
- Ο πίνακας t (translation), μεγέθους 3×1 , που αντιπροσωπεύει τη μετακίνηση της κάμερας. Κάθε στοιχείο του διανύσματος εκφράζει τη μετακίνηση στον κατάλληλο άξονα του συστήματος συντεταγμένων (3 βαθμοί ελευθερίας για το translation).



Εικόνα 56 - Rotation, Translation και Scaling ενός σώματος

Συνοψίζοντας, κατά την pose estimation έχουμε την προσπάθεια εύρεσης 6 παραμέτρων συνολικά, γι' αυτό και γίνεται λόγος για pose estimation 6 βαθμών ελευθερίας⁹² (6 DoF - Degrees of Freedom).

Όπως μπορεί να γίνει εύκολα αντιληπτό, ο συνδυασμός αυτός της rotation και της translation μπορεί να μοντελοποιήσει οποιαδήποτε κίνηση της κάμερας (βλ εικόνα 57).



Εικόνα 57 - Αναπαράσταση Rotation και Translation μιας κάμερας

Η ανάλυση των μεθόδων pose estimation ξεφεύγει από τους στόχους αυτής της εργασίας. Στην κατάλληλη ενότητα, όμως, θα παρουσιαστεί η μέθοδος που χρησιμοποιήθηκε.

⁹² Ο ελάχιστος αριθμός των ανεξάρτητων παραμέτρων που ορίζουν πλήρως ένα γεωμετρικό μέγεθος. Έτσι, για παράδειγμα ένα σημείο έχει 2 βαθμούς ελευθερίας, καθώς ορίζεται πλήρως από δύο συντεταγμένες. Όμοια, μία ευθεία έχει 2 βαθμούς ελευθερίας, καθώς μπορεί να οριστεί από την κλίση της και από την τομή της με κάποιον άξονα (πχ άξονα y) [91]

5. Προτεινόμενη Μεθοδολογία

Γενικά λόγια

Ένα marker system προορισμένο για AR εφαρμογές αποτελείται εν γένει από

- τη μέθοδο κωδικοποίησης κάποιας ψηφιακής πληροφορίας στα markers
- τη μέθοδο ανίχνευσης των markers
- τη μέθοδο ταυτοποίησής τους (αποκωδικοποίηση της φέρουσας πληροφορίας) και
- τη μέθοδο “επικάθισης” του ψηφιακού περιεχομένου

Όπως αναφέρθηκε και στο σχετικό κεφάλαιο, στο marker system που προτείνεται στην παρούσα εργασία θα πρέπει:

- τα markers που χρησιμοποιούνται να
 - έχουνε όσο το δυνατόν μικρότερη πυκνότητα “printed bits” (μικρότερο μέγεθος cell) με το ανάλογο trade off για το διαφορετικό πλήθος διαφορετικών markers
 - περιβάλλονται από τετράγωνο περίγραμμα, ώστε να είναι δυνατή η ακριβής θέση της κάμερας ως προς τα markers (6 DoF tracking)
 - είναι *bitonal* για υψηλότερη αντίθεση και αποτελεσματικότερη ανάγνωση
 - περιέχουν *error correction* για αύξηση της αξιοπιστίας, αλλά και της απόστασης ανάγνωσης
- να γίνεται χρήση *edge detection*, ώστε να έχουμε όσο το δυνατόν αποτελεσματικότερη ανάγνωση, τόσο υπό μη ελεγχόμενες συνθήκες φωτισμού, όσο και υπό μερική επικάλυψη
- να έχουμε αποτελεσματική ανίχνευση των markers από (σχετικά) *ευρυγώνιους φακούς* και από (σχετικά) *μακρινή απόσταση* (ανάλογα με την εφαρμογή)
- το μέγεθος του marker κατά την εκτύπωση να καθορίζεται από το μέγεθος της απόστασης κάμερας-marker, στην οποία απαιτείται αξιόπιστη ανάγνωση και αποκωδικοποίηση.

Στις παραγράφους που ακολουθούν παρουσιάζονται αναλυτικά οι μέθοδοι που επιλέχτηκαν για το προτεινόμενο marker based AR σύστημα της παρούσας εργασίας, οι οποίοι προκύπτουν άμεσα από τις εργασίες των Zisserman et al [50] και Hirzer [95], με μερικές τροποποιήσεις και προσθήκες.

Σε κάθε παράγραφο έχουμε παρουσίαση του χρησιμοποιούμενου αλγορίθμου, επεξηγήσεις, καθώς και screenshots από την εφαρμογή του σε κάποια εικόνα, για καλύτερη εποπτεία της λειτουργίας και του αποτελέσματός του.

Επειδή σε κάθε αλγόριθμο γίνεται έντονη χρήση από nested loops, διάφορα κατώφλια και συνθήκες, επιλέχθηκε η παρουσίασή τους να μη γίνει αποκλειστικά με τη μορφή κειμένου, αλλά στην ευκολότερα κατανοητή μορφή των κειμένων σε blocks διαγραμμμάτων ροής.

Εντοπισμός των markers

Το αρχικό βήμα προς την κατεύθυνση του εντοπισμού των markers είναι ο εντοπισμός των πλευρών τους.

Περί line detection

Στην εργασία τους οι Guru et al [96] αναφέρουν ότι οι διάφορες μέθοδοι line detection που έχουν προταθεί κατά καιρούς, μπορούν να ταξινομηθούν σε 4 κύριες κατηγορίες

- *Statistical based*, όπου γίνεται υπόθεση και επιβεβαίωση της ύπαρξης γραμμικών τμημάτων συγκεκριμένου μήκους με βάση την τοπική πληροφορία στην εικόνα
- *Gradient based*, όπου γίνεται χρήση του μέτρου και της κλίσης της gradient σε κάθε pixel της εικόνας
- *Pixel connectivity - edge linking based*, όπου
 - βρίσκονται πρώτα τοπικά edgels (σε μία περιοχή της εικόνας) και
 - ενώνονται σε μικρές edges βάσει της μεταξύ τους απόστασης και της κατεύθυνσής τους (στο “επίπεδο” της περιοχής της εικόνας), οι οποίες
 - ενώνονται, με τη σειρά τους, σε μεγαλύτερες ευθείες γραμμές (στο “επίπεδο” της συνολικής εικόνας)
- *Hough Transform based*, όπου γίνεται ανίχνευση των διάφορων γεωμετρικών χαρακτηριστικών μέσω παραμετρικών αναπαραστάσεών τους. Για την ανίχνευση γραμμών χρησιμοποιείται συνήθως η πολική μορφή⁹³ $r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$. Παρόλο που ο Hough Transform είναι ανθεκτικός στο θόρυβο και στις ασυνέχειες στην εικόνα, αδυνατεί να βρει τα άκρα των γραμμών, δυσκολεύεται στον εντοπισμό μικρών γραμμικών τμημάτων, ενώ επίσης απαιτεί αρκετή μνήμη και αρκετούς υπολογισμούς⁹⁴.

Από αυτές, η τρίτη κατηγορία (Pixel connectivity - edge linking based) έχει σαφώς καλύτερη απόδοση σε σχέση με τις άλλες, καθώς οι μέθοδοί της βασίζονται στην συνδεσιμότητα των pixels που ανιχνεύονται ως edgels, κάτι που είναι σε μεγάλο βαθμό εξασφαλισμένο.

Μία γρήγορη και αποτελεσματική μέθοδος, που εμπίπτει σε αυτή την κατηγορία, είναι η μέθοδος των Zisserman et al [50], που υιοθετείται για την παρούσα εργασία με ορισμένες τροποποιήσεις. Κι αυτό γιατί, ενώ η μέθοδος αυτή αναφέρεται σε ανίχνευση οποιονδήποτε γραμμικών τμημάτων σε μία εικόνα, στην παρούσα εργασία ενδιαφέρει η ανίχνευση γραμμικών τμημάτων σε περιοχές της εικόνας με υψηλή αντίθεση (στις πλευρές ενός marker όπου έχουμε υψηλή αντίθεση ανάμεσα στο περίγραμμα του και στο φόντο του).

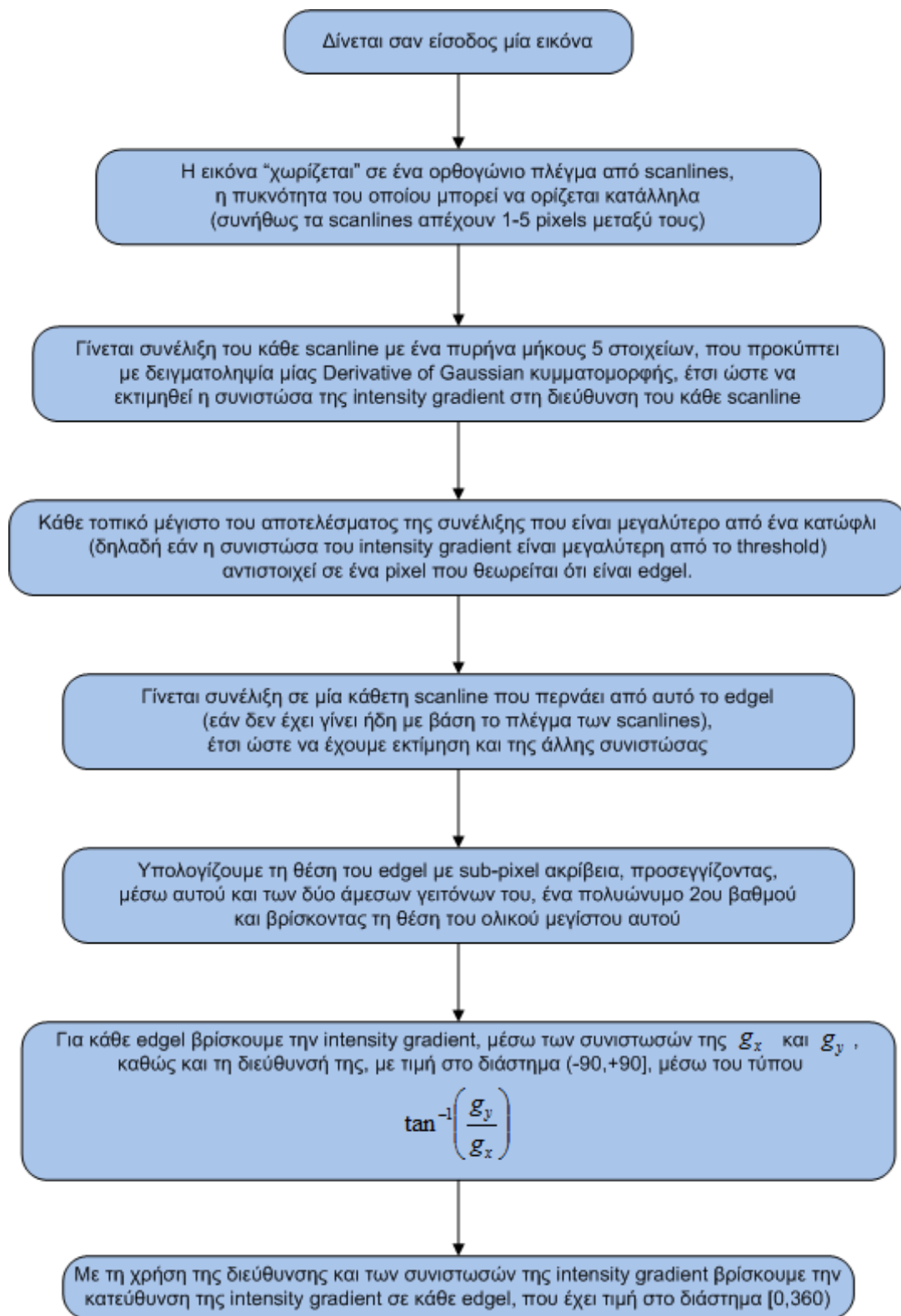
Λεπτομέρειες για το μαθηματικό κομμάτι του χρησιμοποιούμενου αλγορίθμου, μπορούν να βρεθούν στο σχετικό κεφάλαιο περί μαθηματικού υποβάθρου..

⁹³ Όπου

r η απόσταση της γραμμής από την αρχή του συστήματος συντεταγμένων
 θ Η γωνία που σχηματίζει η γραμμή σε σχέση με τον άξονα x

⁹⁴ Κατά καιρούς έχουν προταθεί διάφορες παραλλαγές, που επιλύουν κάποια από τα μειονεκτήματα

Εύρεση των Edgels



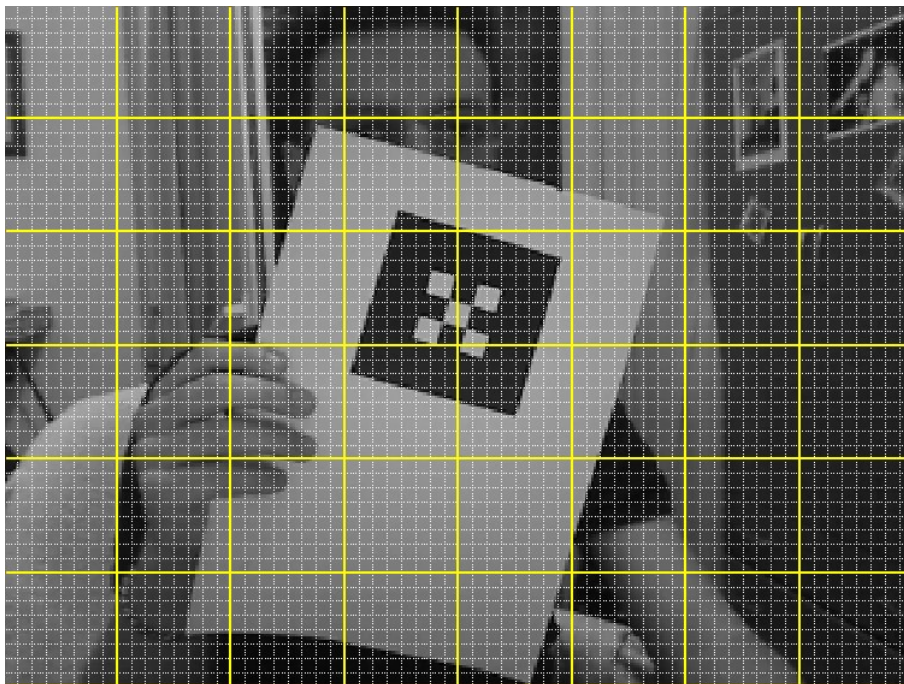
Αλγόριθμος 1 - Προτεινόμενος αλγόριθμος για εντοπισμό edgels

Όλα τα screenshots της παρούσας παραγράφου προκύπτουν από τη χρήση κάποιας τοπικής περιοχής της παρακάτω εικόνας.



Εικόνα 58 - Η εικόνα που θα υποστεί επεξεργασία από τους προτεινόμενους αλγόριθμους για την παρουσίαση της λειτουργίας τους

Στην εικόνα 59 φαίνεται ο τεμαχισμός της προηγούμενης εικόνας σε ισομεγέθεις περιοχές (έντονη κίτρινη γραμμή), όπως επίσης και τα scanlines (άσπρες διακεκομμένες γραμμές) κατά μήκος των οποίων γίνεται συνέλιξη κατά το edge detection. Τα scanlines αποτελούν πάντα ένα ορθογώνιο πλέγμα, που μπορεί να έχει οποιαδήποτε κατεύθυνση⁹⁵.



Εικόνα 59 - Διαχωρισμός της εικόνας σε ισομερείς περιοχές για τοπική επεξεργασία, και απεικόνιση του

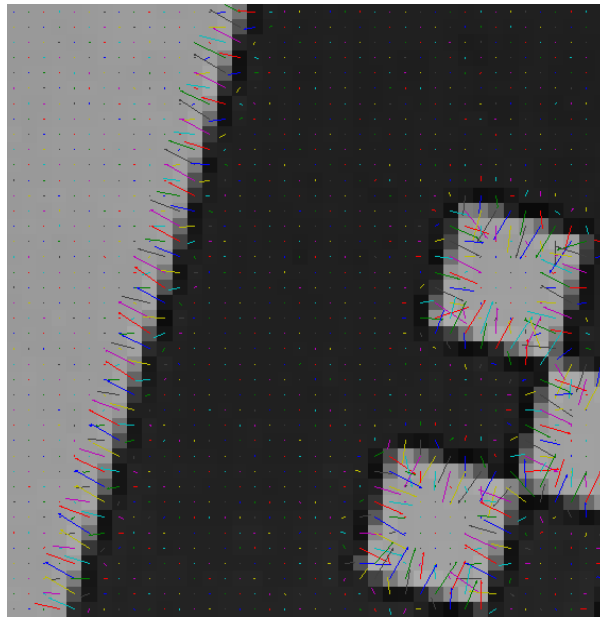
Όπως θα εξηγηθεί αργότερα, η εικόνα που δίνεται σαν είσοδος στον αλγόριθμο αυτής της υποπαραγράφου δεν είναι ολόκληρο το εκάστοτε frame της κάμερας, αλλά κάποια περιοχή αυτού, για τη διεξαγωγή τοπικής επεξεργασίας.

⁹⁵ Εδώ παρουσιάζονται στην περίπτωση που οι διευθύνσεις τους συμπίπτουν με αυτές τους συστήματος συντεταγμένων της εικόνας

Συνέλιξη κατά μήκος των Scanlines

Με τη συνέλιξη κατά μήκος της κάθε scanline, έχουμε μία ένδειξη για το μέγεθος της συνιστώσας της intensity gradient σε αυτή τη διεύθυνση. Έπειτα γίνεται ανίχνευση των τοπικών μεγίστων, ενώ ακολουθεί και η συνέλιξη κατά μήκος μίας ορθογώνιας scanline (εάν δεν έχει γίνει ήδη με βάση το υπάρχον πλέγμα) που περνάει από το κάθε τοπικό μέγιστο, ώστε να βρεθεί και η άλλη συνιστώσα.

Για να δούμε το αποτέλεσμα της συνέλιξης κατά μήκος των scanlines, εφαρμόζουμε το παραπάνω αλγόριθμο στο “κουτάκι” της εικόνας που βρίσκεται στην 3^η γραμμή και στην 4^η στήλη της εικόνας 59. Θεωρούμε μοναδιαία απόσταση μεταξύ των scanlines, ώστε η επεξεργασία να γίνει σε ολόκληρο το κουτάκι και να έχουμε πιο ολοκληρωμένη εικόνα του αποτελέσματος.



Εικόνα 60 - Γραφική απεικόνιση της intensity gradient σε κάθε σημείο μιας εικόνας

Τα βελάκια της εικόνας δείχνουν, έπειτα από υπολογισμό της οριζόντιας και της κάθετης συνιστώσας της intensity gradient για κάθε pixel, μία εκτίμηση

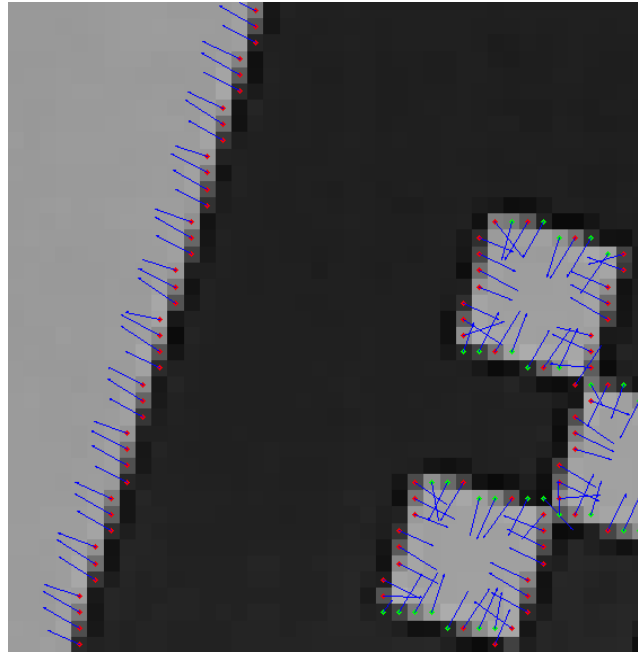
- του μεγέθους της αλλαγής της φωτεινότητας (το μήκος των βελών) και
- της κατεύθυνσης της μέγιστη αλλαγής (η κατεύθυνση των βελών).

Όπως φαίνεται, οι εκτίμηση και των δύο μεγεθών είναι αρκετά καλή, παρόλο που η μέθοδος που χρησιμοποιούμε (για λόγους ταχύτερης εκτέλεσης) δεν είναι η ακριβέστερη δυνατή. Ακόμα και υπό την παρουσία σφάλματος, η εκτίμηση αυτή μπορεί να δώσει πολύ καλά αποτελέσματα με τη δημιουργία του κατάλληλου αλγορίθμου.

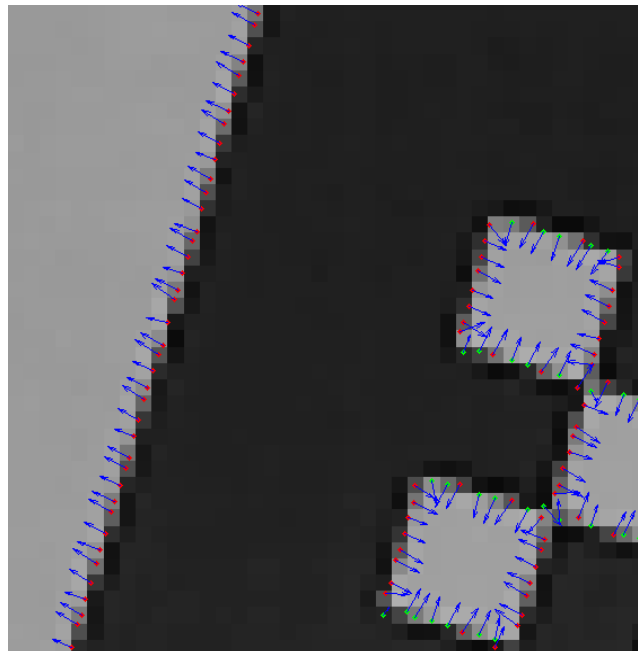
Σε πολλά από τα σημεία που δεν υπάρχει κάποια edge, όπως γύρω από το marker ή στο εσωτερικό του marker, τα βελάκια, αν και πολύ μικρά, φαίνεται να μην έχουν μηδενικό μήκος, όπως θα ήταν και το ιδανικό. Αυτό μπορεί να οφείλεται σε διάφορους λόγους, όπως στον αισθητήρα της κάμερας, στην κβαντοποίηση που γίνεται για την αναπαράσταση της εικόνας με πίνακα ακέραιων αριθμών, σε ηλεκτρονικό θόρυβο, στο φωτισμό τους περιβάλλοντος, στον τρόπο εκτύπωσης του marker, κτλ. Σε όποιο pixel, όμως, υπολογίζεται μία ανίσχυρη συνιστώσα (με βάση κάποιο σχετικό κατώφλι που ορίζουμε), θεωρούμε ότι δεν υπάρχει edge.

Edgel detection και Sub-pixel ακρίβεια

Παρακάτω φαίνεται η εφαρμογή του αλγορίθμου στο ίδιο “κουτάκι”. Αυτή τη φορά φαίνονται μόνο οι θέσεις των edgels και η intensity gradient σε αυτά. Με κόκκινο χρώμα παριστάνονται τα edgels που ανιχνεύθηκαν με συνέλιξη κατά μήκος των οριζόντιων scanlines, ενώ με πράσινο τα edgels που ανιχνεύτηκαν με συνέλιξη κατά μήκος των κάθετων scanlines.



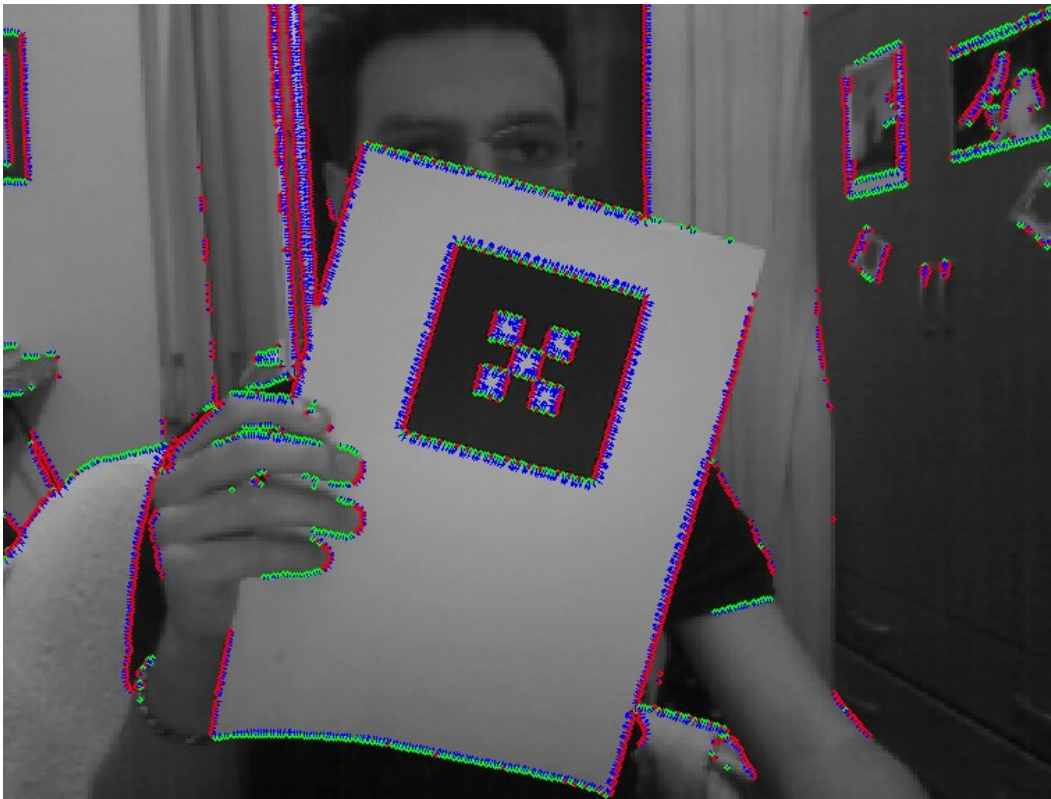
Εικόνα 61 - Ανίχνευση edgels χωρίς sub-pixel ακρίβεια



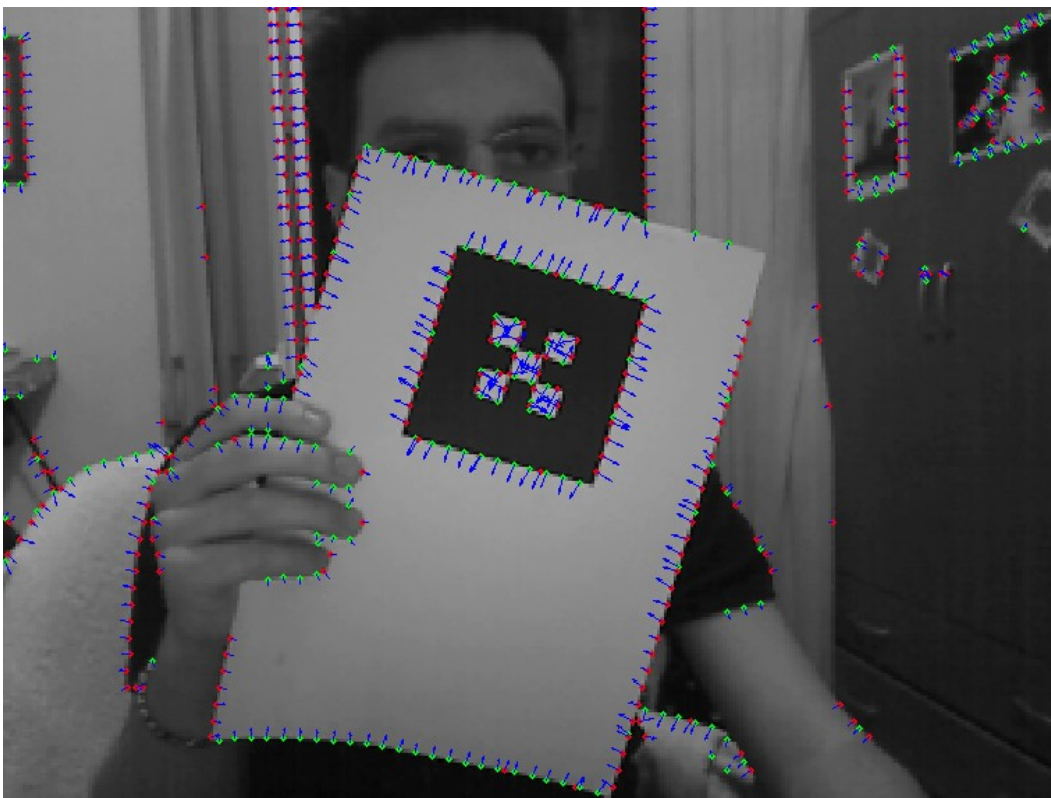
Εικόνα 62 - Ανίχνευση edgels με sub-pixel ακρίβεια

Είναι φανερό ότι τα edgels στις δύο εικόνες είναι ίδια, με τη διαφορά ότι στην πρώτη όλα βρίσκονται στο κέντρο του κάθε pixel, ενώ στη δεύτερη έχουμε sub-pixel ακρίβεια, με αποτέλεσμα τα edgels να βρίσκονται πιο κοντά πάνω σε μία νοητή γραμμή-edge.

Για να έχουμε μία πιο ολοκληρωμένη εικόνα του αποτελέσματος της χρήσης των αραιών scanlines, παρουσιάζονται τα edgels που ανιχνεύτηκαν σε όλη την εικόνα, για διαφορετική απόσταση μεταξύ των scanlines.



Εικόνα 63 - Ανίχνευση edgels σε ολόκληρη την εικόνα (πυκνό πλέγμα scanlines)



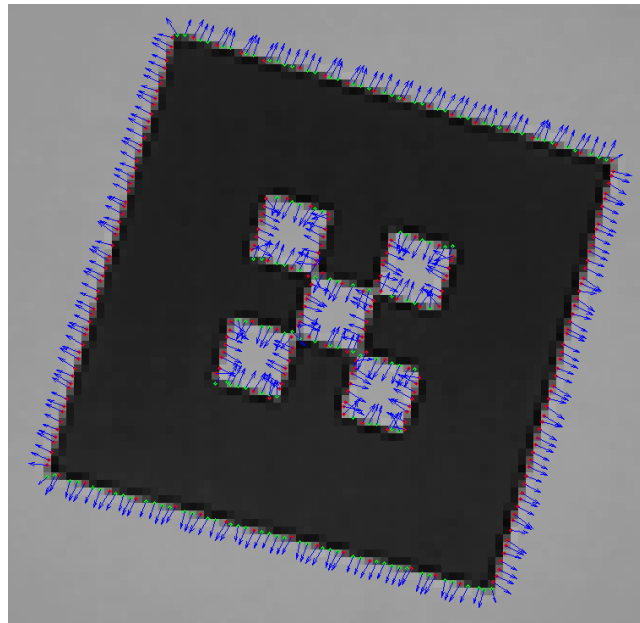
Εικόνα 64 - Ανίχνευση edgels σε ολόκληρη την εικόνα (αραιό πλέγμα scanlines)

Στην πρώτη εικόνα έχουμε μοναδιαία απόσταση μεταξύ των scanlines, ενώ στη δεύτερη έχουμε απόσταση 5 pixels. Επίσης γίνεται χρήση πολύ χαμηλού κατωφλίου φωτεινότητας κατά την ανίχνευση των edgels, για καλύτερη επόπτευση του αποτελέσματος των scanlines.

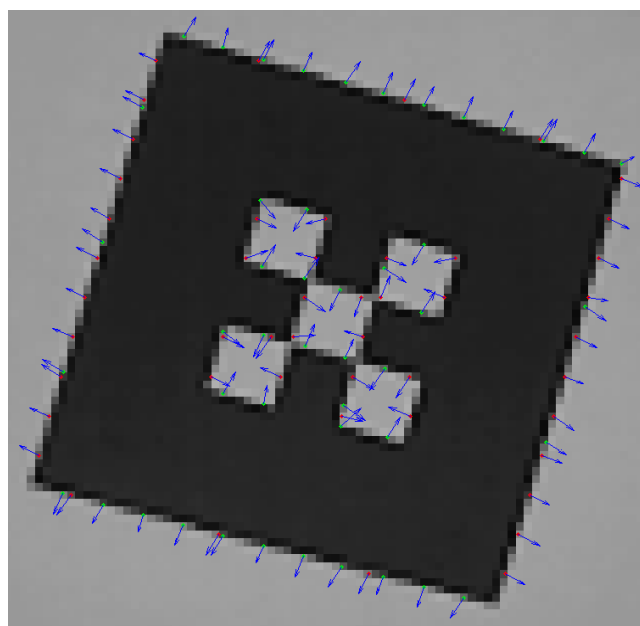
Όπως φαίνεται με τη χρήση αραιών scanlines, μπορεί να γλιτώνουμε το φόρτο επεξεργασίας σε ολόκληρη την επιφάνεια της εικόνας, υπάρχει κάποια επίπτωση όμως στον αριθμό των edgels που “υποστηρίζουν” κάθε edge.

Αυτό το trade off θα μας απασχολήσει, όμως, σε μεταγενέστερο σημείο.

Παρακάτω ακολουθεί μία συνολική παρατήρηση ολόκληρου του marker (με χειροκίνητη συγκόλληση των εικόνων που επιστρέφει ο αλγόριθμος για διάφορες τοπικές περιοχές) και των edgels που βρέθηκαν σε αυτό, για διαφορετική απόσταση scanlines.



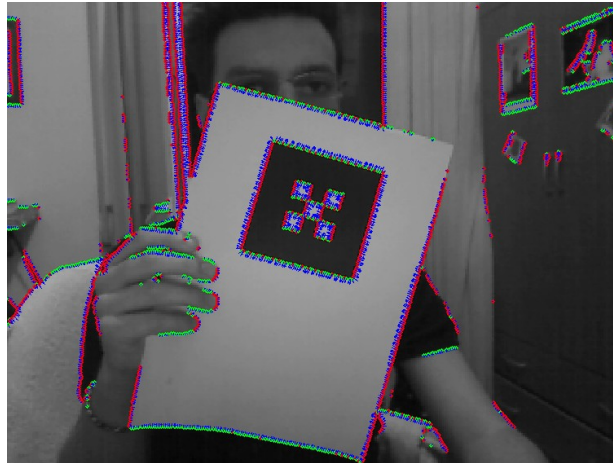
Εικόνα 65 - Εντοπισμός των edgels ενός marker με πυκνό πλέγμα scanlines



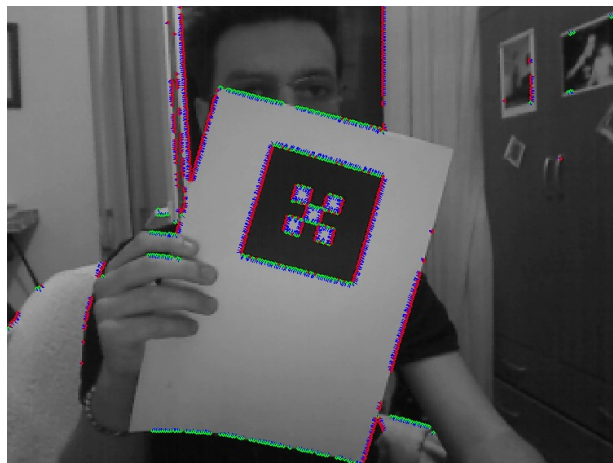
Εικόνα 66 - Εντοπισμός των edgels ενός marker με αραιό πλέγμα scanlines

Παρακάτω φαίνεται η επίδραση του κατωφλίου φωτεινότητας κατά την ανίχνευση των edgels (χαμηλό-20, μεσαίο-40 και υψηλό-60 αντίστοιχα).

Με χαμηλό κατώφλι εντοπίζονται πολλά “αδύναμα” edgels (και έχουμε πολλή περιττή πληροφορία), ενώ με υψηλό εντοπίζονται μόνο τα πολύ “ισχυρά”, με κίνδυνο όμως να αποτύχει η ανίχνευση κάποιου marker λόγω φωτιστικών συνθηκών. Και αυτό το κατώφλι θα εξεταστεί σε μεταγενέστερο σημείο και όχι σε αυτή την ενότητα.



Εικόνα 67 - Επίδραση χαμηλής τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels

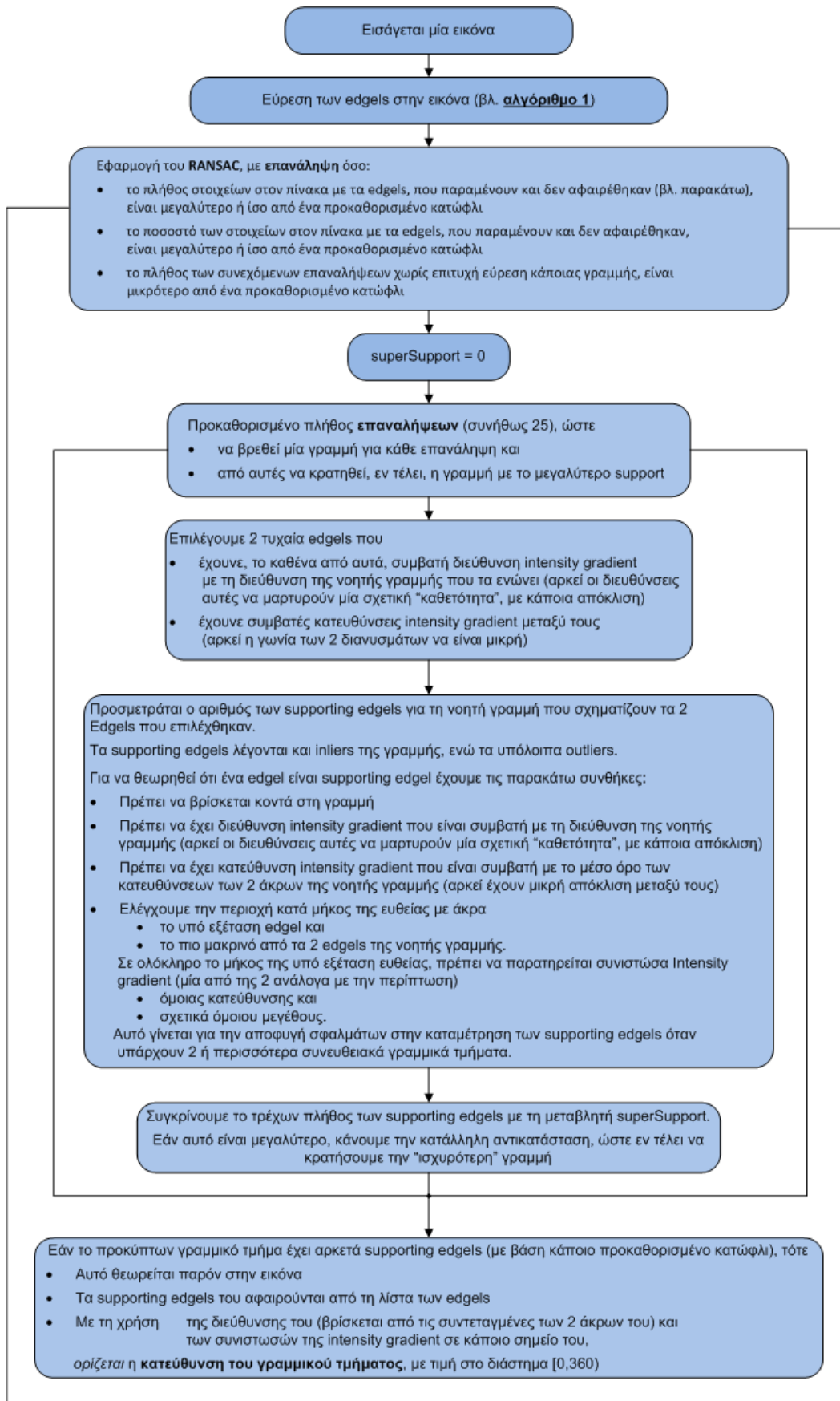


Εικόνα 68 - Επίδραση μέσης τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels



Εικόνα 69 - Επίδραση υψηλής τιμής κατωφλίου φωτεινότητας στην ανίχνευση των edgels

Εύρεση των Γραμμικών Τμημάτων



Αλγόριθμος 2 - Προτεινόμενος αλγόριθμος για εντοπισμό μικρών γραμμικών τμημάτων

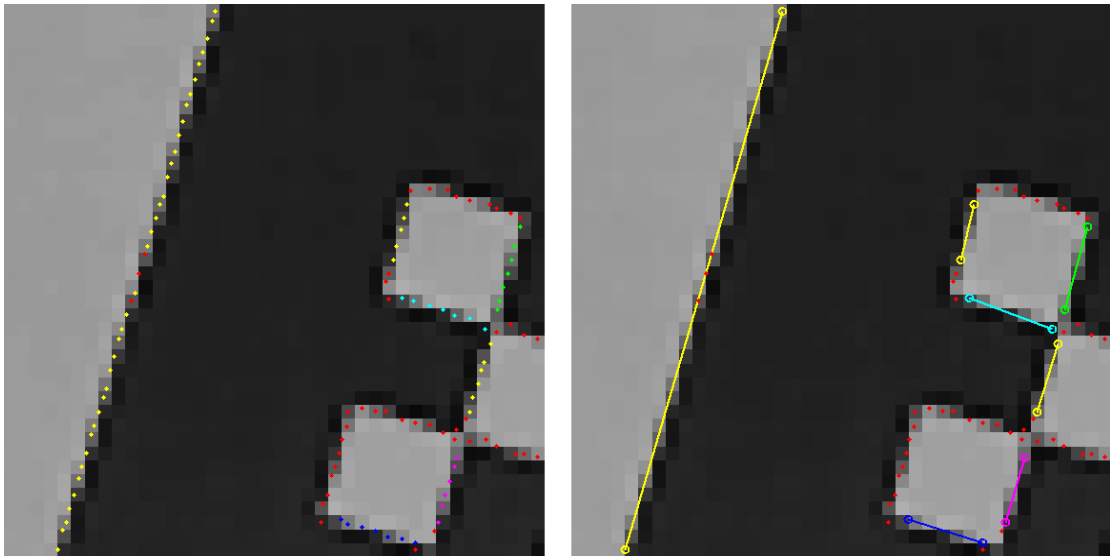
Όπως θα φανεί αργότερα, και σε αυτό τον αλγόριθμο έχουμε επεξεργασία σε τοπική περιοχή του εκάστοτε frame.

Παρακάτω φαίνεται το αποτέλεσμα της ομαδοποίησης των edgels σε μικρά γραμμικά τμήματα για διάφορες πυκνότητες του πλέγματος των scanlines.

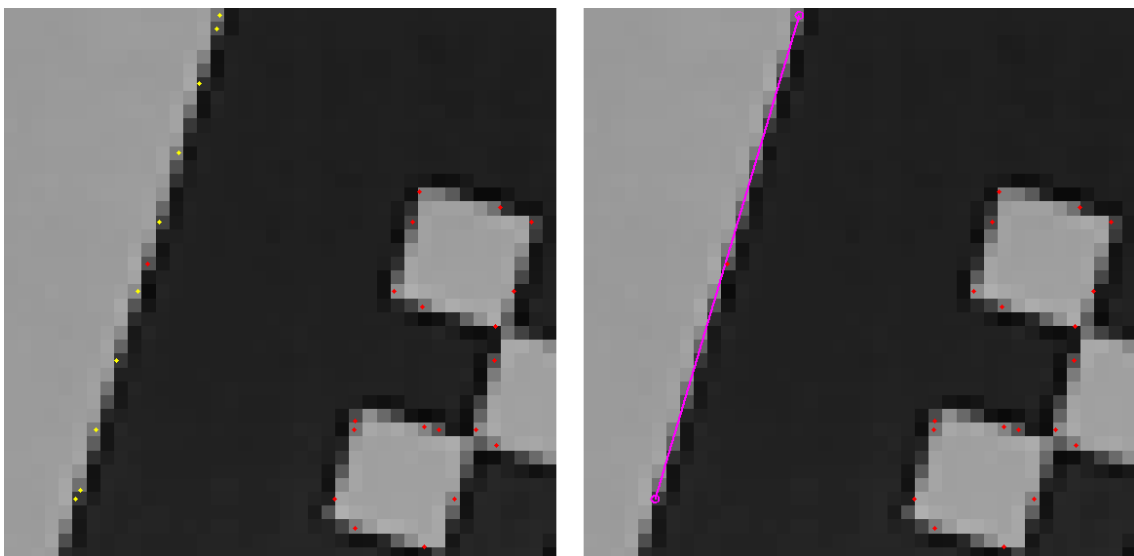
Το πρώτο ζεύγος αντιστοιχεί σε μοναδιαία απόσταση μεταξύ των scanlines, ενώ το δεύτερο σε απόσταση 5 pixels.

Επίσης, σε κάθε ζεύγος, στην αριστερή εικόνα παρουσιάζεται με χρωματισμό η ομαδοποίηση των edgels σε γραμμικά τμήματα, ενώ στη δεξιά απεικονίζονται σαφέστερα οι διάφορες γραμμές που θεωρούνται παρούσες στην εικόνα.

Και στις δύο εικόνες, όμως, όσα edgels παρουσιάζονται με κόκκινο χρώμα είναι outliers, δεν “υποστηρίζουν” δηλαδή κάποια γραμμή.



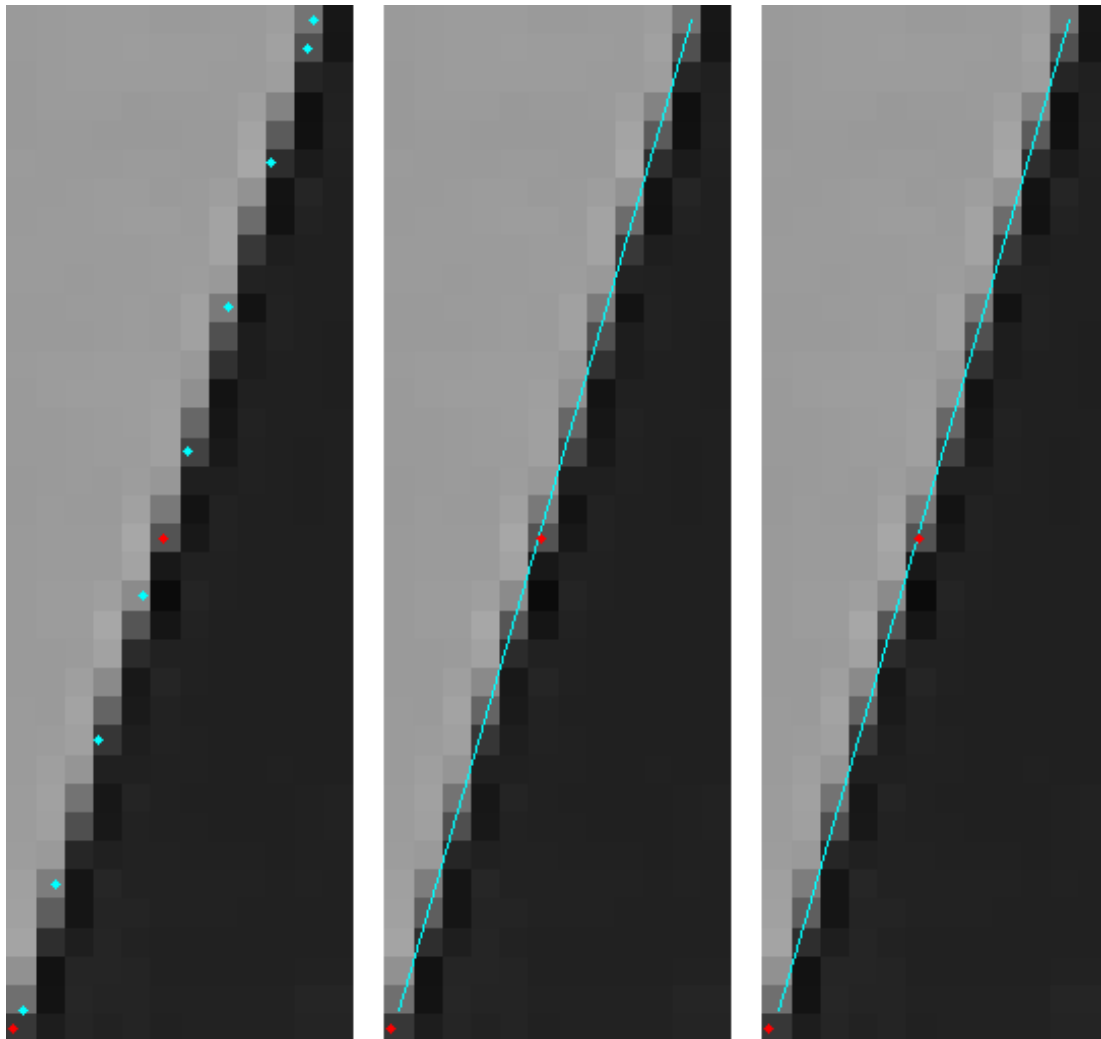
Εικόνα 70 - Εντοπισμός γραμμικών τμημάτων (πυκνό πλέγμα scanlines)



Εικόνα 71 - Εντοπισμός γραμμικών τμημάτων (πυκνό πλέγμα scanlines)

Στον αλγόριθμο των Zisserman et al [50] στο τέλος ακολουθεί και επαναυπολογισμός της κάθε γραμμής, με *orthogonal regression*, για μεγαλύτερη ακρίβεια. Εκεί, όμως, οι συγγραφείς, στόχευαν απλά στο *tracking* μικρών γραμμικών τμημάτων, χωρίς να επιχειρούν ένωσή τους.

Ενδεικτικά, παρακάτω φαίνονται οι γραμμές που δίνονται από τα *supporting edgels* της αριστερής εικόνας. Στη μεσαία έχουμε απλό υπολογισμό της γραμμής, ενώ στη δεξιά έχουμε επαναυπολογισμό της γραμμής με *orthogonal regression*, για μεγαλύτερη ακρίβεια.



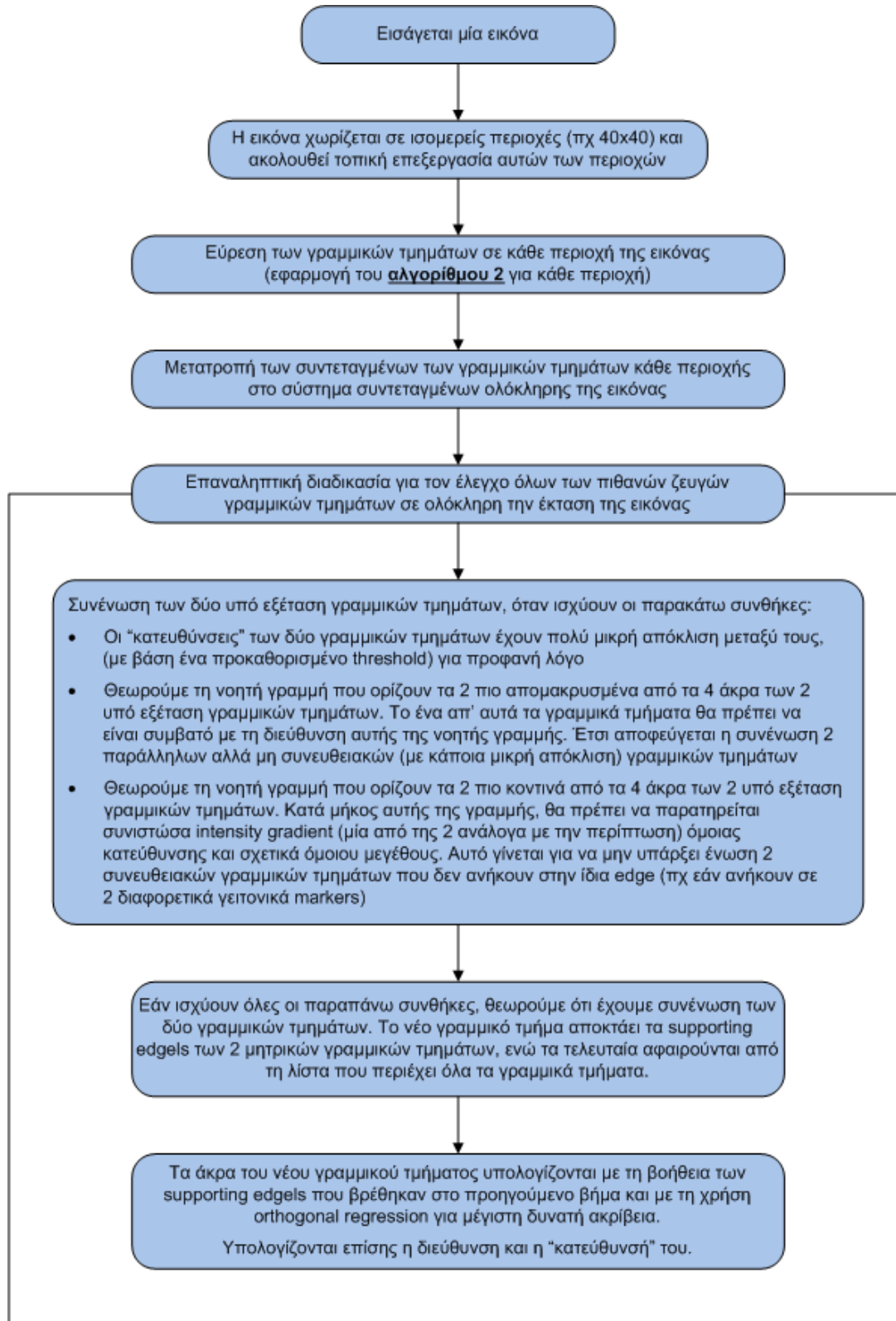
Εικόνα 72 - Επίδραση της *orthogonal regression* στην κλίση μιας γραμμής

Στον αλγόριθμο της παρούσας εργασίας, όπως θα δούμε, η χρήση *orthogonal regression* γίνεται σε μεταγενέστερο στάδιο, μετά την ένωση των μικρών γραμμικών τμημάτων σε μεγαλύτερα (με χρήση όλων των *supporting edgels* αυτών) και πριν την επιμήκυνση των νέων, μεγαλύτερων γραμμών.

Η διαδικασία της μετέπειτα ένωσης δεν επηρεάζεται από (σχετικά μεγάλα) σφάλματα στις κλίσεις των γραμμικών τμημάτων. Κατά την επιμήκυνση όμως, παρόλο που η διαφορά στις κλίσεις φαίνεται αμελητέα (βλ παραπάνω εικόνα), μπορεί να έχουμε σημαντικές επιπτώσεις. Στην περίπτωση του απλού υπολογισμού η γραμμή μπορεί να “αναχωρεί” από την *edge*, με αποτέλεσμα να αποτυγχάνει η επιμήκυνση. Ο επαναυπολογισμός με *orthogonal regression* βοηθά στην καταπολέμηση αυτής ακριβώς της “αναχώρησης”.

Συνένωση γραμμικών τμημάτων σε μεγαλύτερα

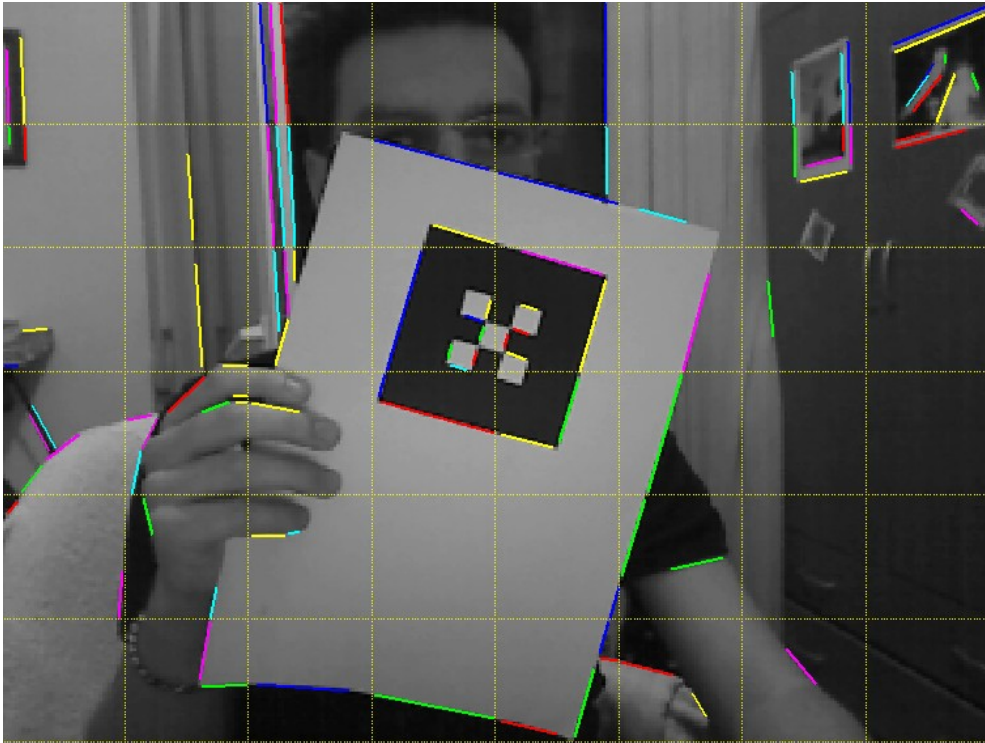
Μετά την εύρεση μικρών γραμμικών τμημάτων σε κάθε περιοχή της εικόνας έπεται η κατάλληλη συνένωσή τους, ώστε να σχηματιστούν γραμμές μεγαλύτερου μήκους που πλησιάζουν περισσότερο στις ευθείες γραμμικές edges που θα διέκρινε ένας άνθρωπος στην εικόνα.



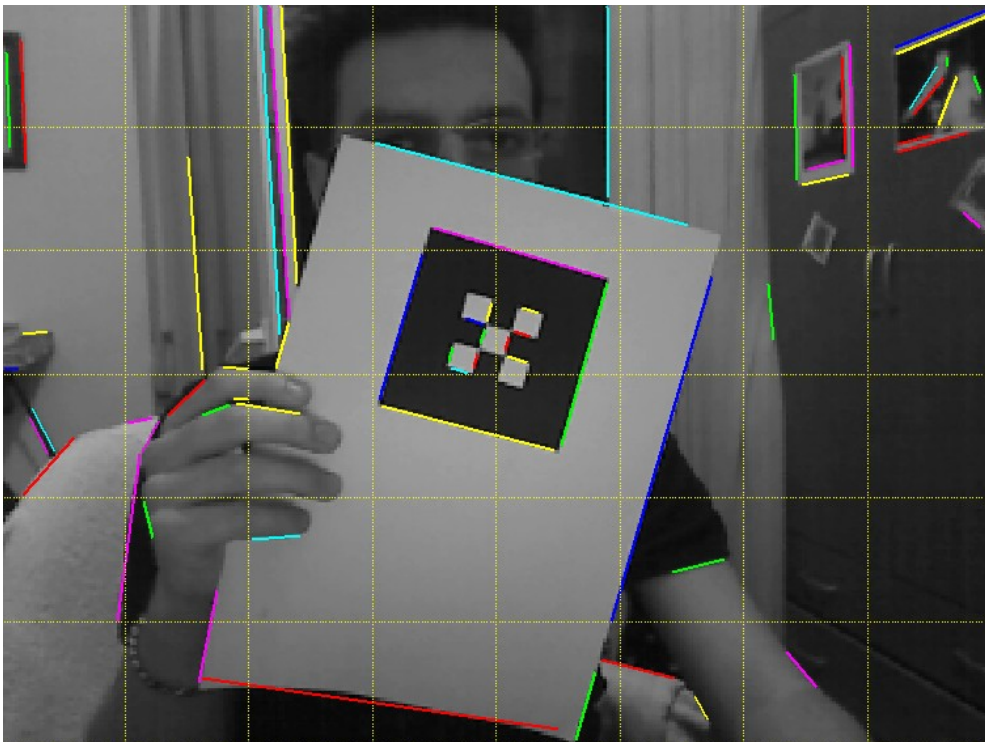
Αλγόριθμος 3 - Προτεινόμενος αλγόριθμος συνένωσης μικρών γραμμικών τμημάτων σε μεγαλύτερα

Ακολουθούν μερικά screenshots εφαρμογής του παραπάνω αλγορίθμου, στα οποία γίνεται χρήση χαμηλού κατωφλίου φωτεινότητας, ώστε να προκύψουν αρκετά γραμμικά τμήματα για καλύτερη εποπτεία του αλγορίθμου.

Αρχικά παρουσιάζονται τα αποτελέσματα για πυκνό πλέγμα των scanlines (μοναδιαία απόσταση μεταξύ αυτών)

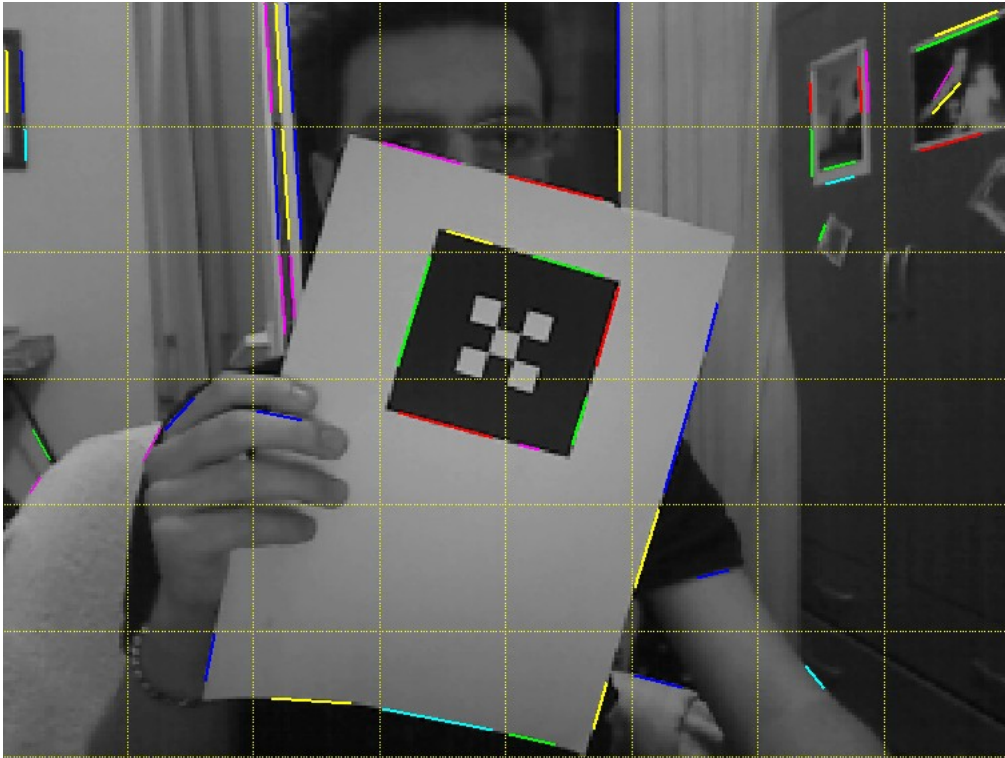


Εικόνα 73 - Γραμμικά τμήματα πριν τη συνένωση (πυκνό πλέγμα scanlines)

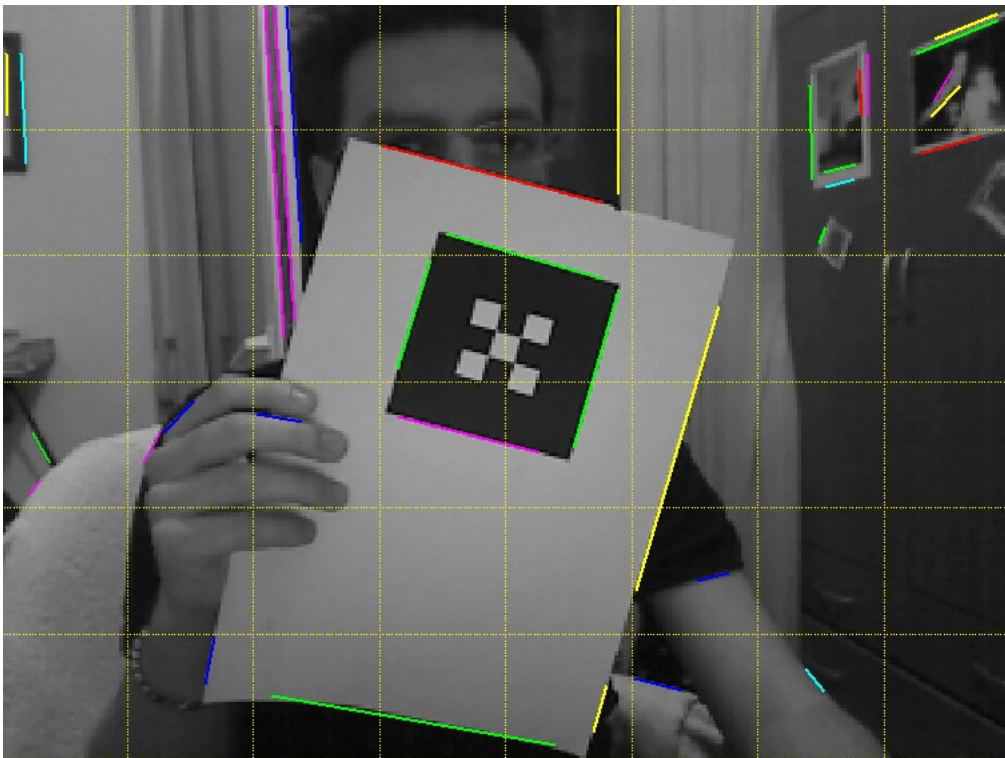


Εικόνα 74 - Γραμμικά τμήματα μετά τη συνένωση (πυκνό πλέγμα scanlines)

Παρακάτω έχουμε εφαρμογή του αλγορίθμου για αραιό πλέγμα των scanlines (5 pixels μεταξύ αυτών)



Εικόνα 75 - Γραμμικά τμήματα πριν τη συνένωση (αραιό πλέγμα scanlines)



Εικόνα 76 - Γραμμικά τμήματα μετά τη συνένωση (αραιό πλέγμα scanlines)

Επιμήκυνση των γραμμών

Όπως βλέπουμε στα αποτελέσματα της προηγούμενης παραγράφου, μετά τη συνένωση των μικρών γραμμικών τμημάτων καταλήγουμε σε μεγαλύτερα γραμμικά τμήματα, που προσεγγίζουν καλύτερα τις πραγματικές edges.

Ακόμα κι αυτές όμως δεν είναι πλήρους μεγέθους. Αντιθέτως, λείπει ένα (συνήθως) μικρό κομμάτι και στις 2 πλευρές, όσο πυκνό κι αν είναι το πλέγμα των scanlines.

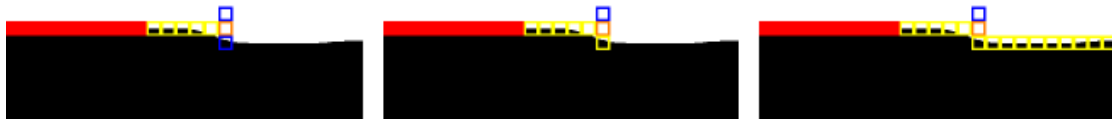
Καταφεύγουμε, έτσι, στη λύση της επιμήκυνσης του κάθε γραμμικού τμήματος και προς τις δύο κατευθύνσεις, προκειμένου να καταλήξουμε σε edges πλήρους μεγέθους.

Παρόλο που αρχικά μπορεί να φαίνεται σχετικά εύκολη η διαδικασία της επιμήκυνσης (με τη χρήση και πάλι της intensity gradient), προκύπτουν προβλήματα, μιας και οι πλευρές του marker στην εικόνα δεν είναι απολύτως ευθύγραμμες

- είτε λόγω της χαμηλής ανάλυσης (ιδίως σε μεγάλες κλίσεις του marker σε σχέση με την κάμερα)
- είτε λόγω παραμόρφωσης της χάρτινης επιφάνειας του marker (η επιφάνειά του δεν είναι απολύτως επίπεδη)
- είτε λόγω κακής εκτύπωσης

Προς επίλυση αυτού του προβλήματος, κατά την επέκταση ενός γραμμικού τμήματος προς μία κατεύθυνση δεν εξετάζουμε μόνο το επόμενο συνευθειακό σημείο.

Απεναντίας, σε περίπτωση αποτυχίας εξετάζουμε επιπλέον τα 2 γειτονικά pixels του εκάστοτε εξεταζόμενου pixel, που βρίσκονται σε διεύθυνση κάθετη με αυτή του ευθύγραμμου τμήματος, ενώ η διαδικασία εξέτασης των pixels συνεχίζεται σε μία μετατοπισμένη νοητή ευθεία με την ίδια κλίση, όπως φαίνεται στην εικόνα 77.



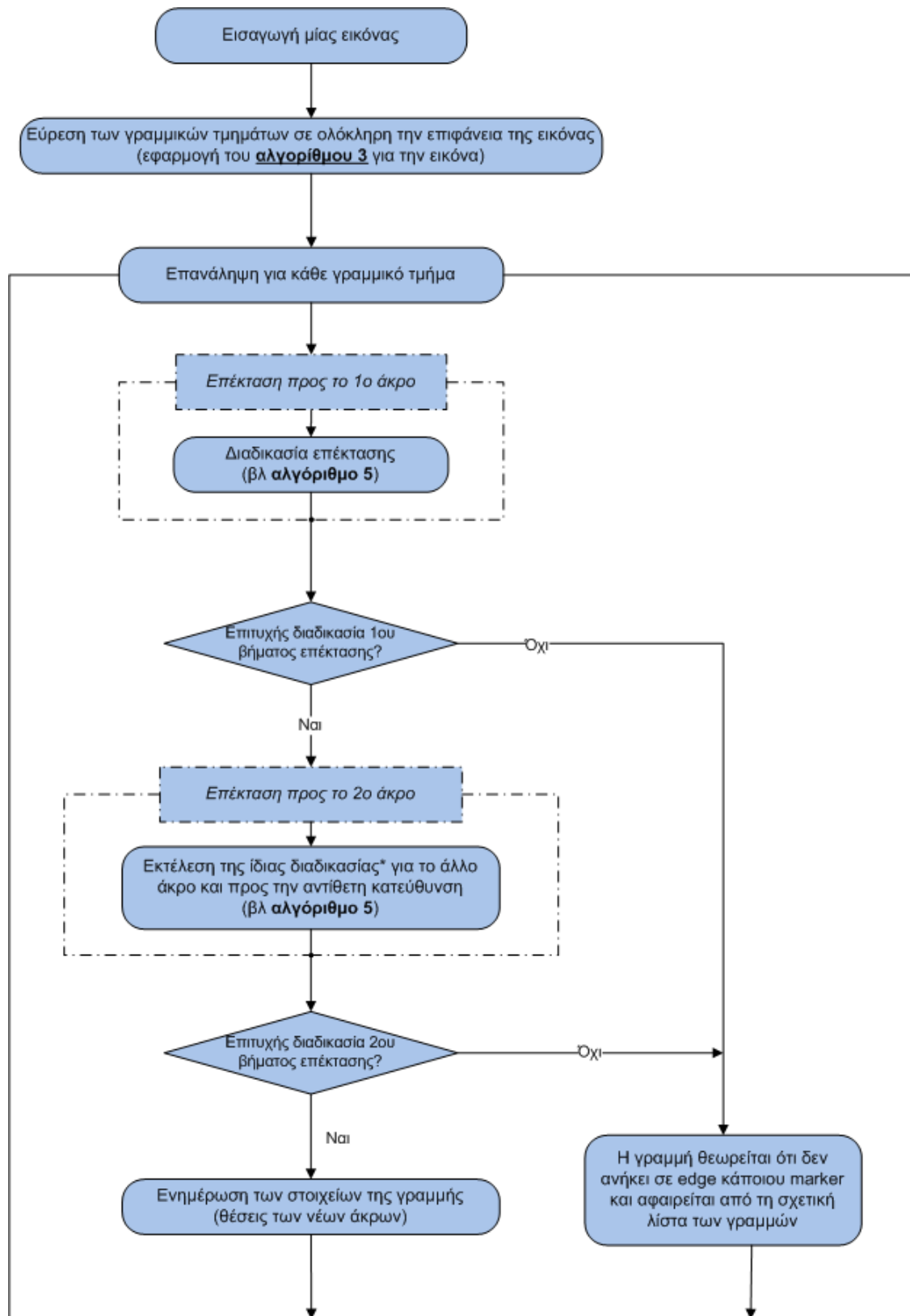
Εικόνα 77 - Επέκταση παραμορφωμένου γραμμικού τμήματος

Η όλη διαδικασία τερματίζεται, όπως φαίνεται παρακάτω, όταν το εξεταζόμενο pixel έχει αρκετά μεγάλη τιμή φωτεινότητας.

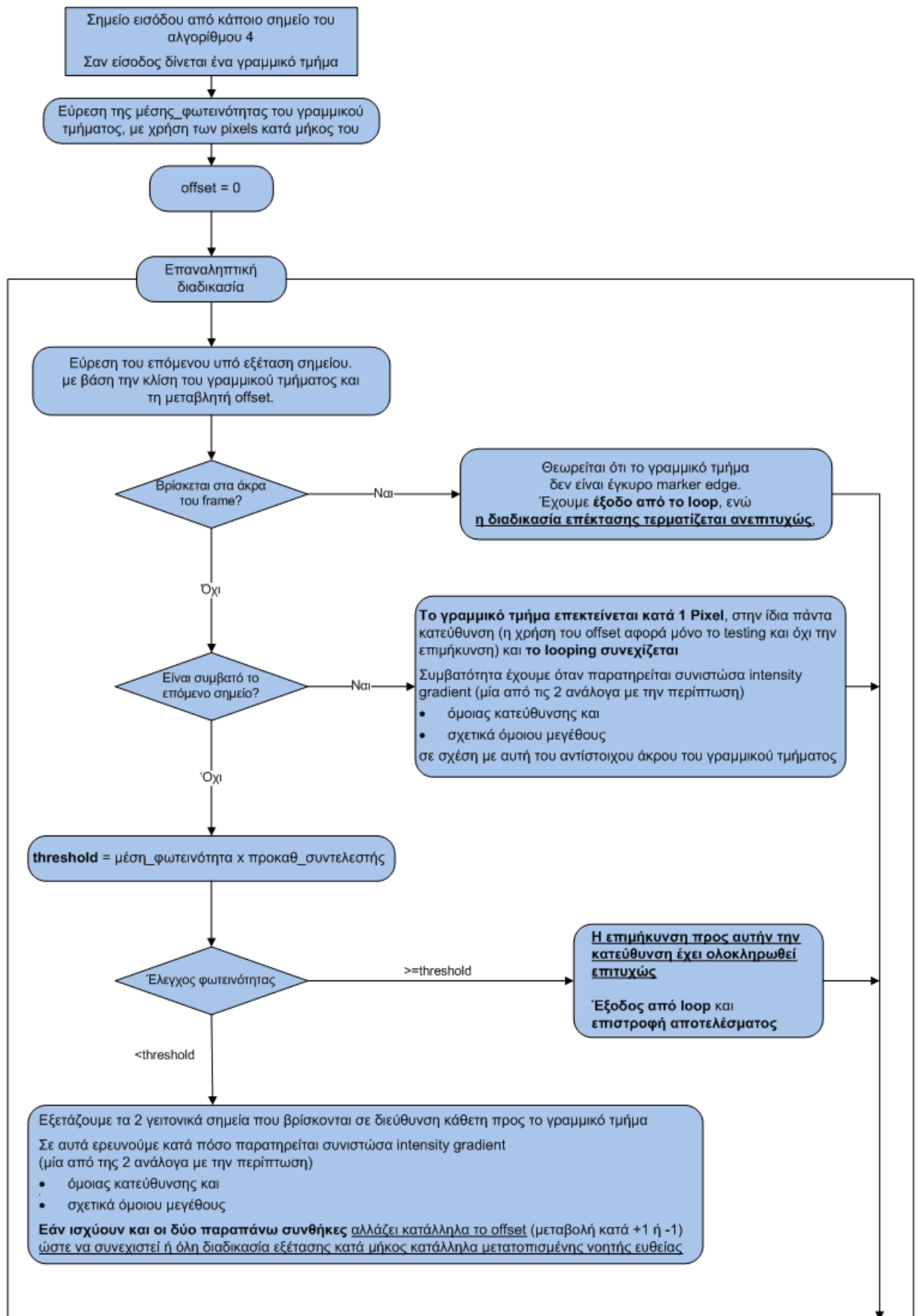


Εικόνα 78 - Επέκταση γραμμικού τμήματος και συνθήκη τερματισμού

Οι εικόνες αυτής της σελίδας προέρχονται από την εργασία [95] και δείχνουν παραστατικά την επιλογή των εξεταζόμενων pixels κατά την επιμήκυνση.

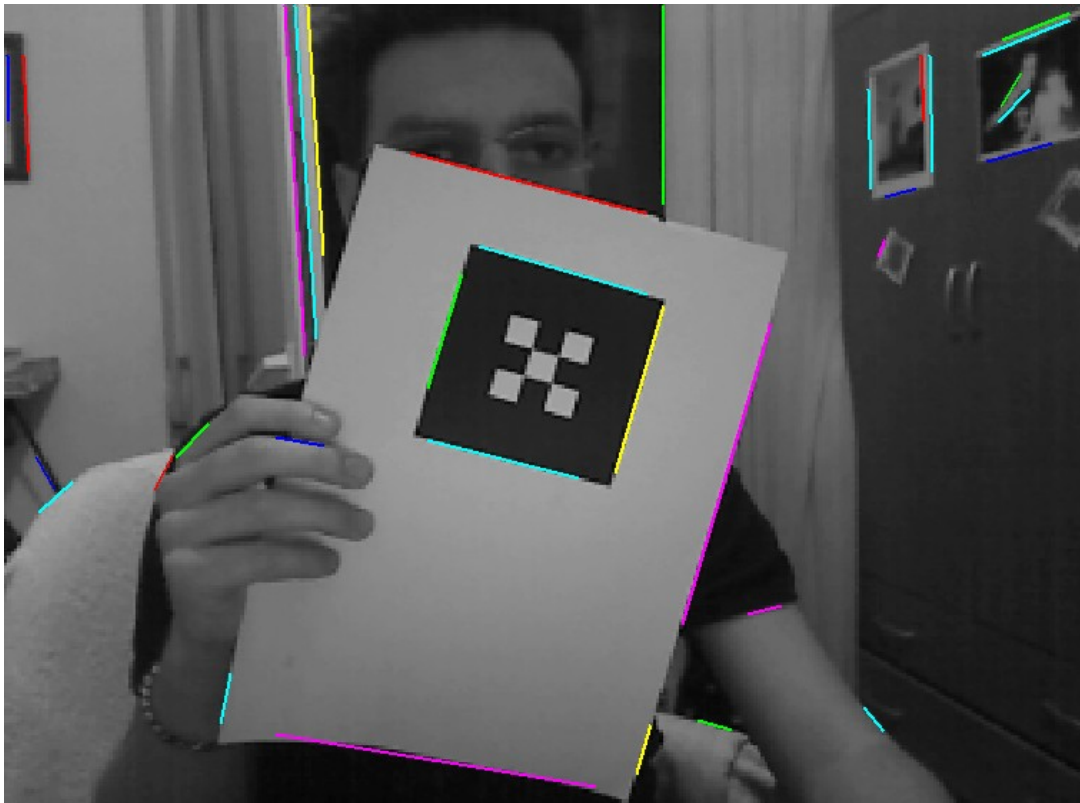


Αλγόριθμος 4 - Προτεινόμενος αλγόριθμος επιμήκυνσης γραμμών και προς τα 2 άκρα

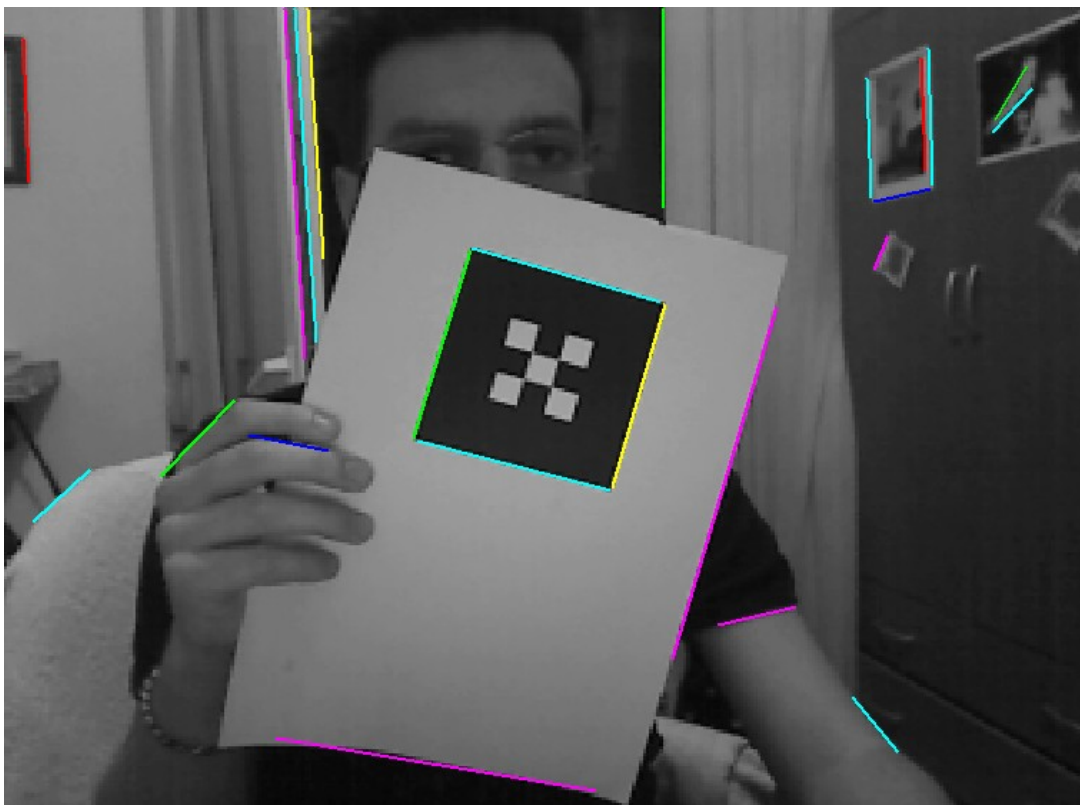


Αλγόριθμος 5 - Προτεινόμενος αλγόριθμος επιμήκυνσης γραμμής προς ένα άκρο

Τα παρακάτω screenshots, πριν και μετά, αντίστοιχα, από την επιμήκυνση, δείχνουν τα αποτελέσματα των παρουσιαζόμενων αλγορίθμων.



Εικόνα 79 - Γραμμικά τμήματα πριν την επιμήκυνση



Εικόνα 80 - Γραμμικά τμήματα μετά την επιμήκυνση

Εντοπισμός των περιγράμματα των markers

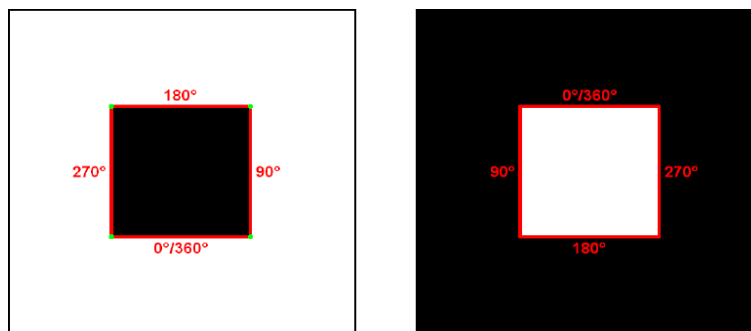
Το τελικό βήμα για τον εντοπισμό ενός marker είναι ο εντοπισμός του τετράπλευρου περιγράμματός του. Όπως είναι αναμενόμενο, αυτό γίνεται με κατάλληλο συνδυασμό των edges που έχουν ήδη βρεθεί, συνδυάζοντας τες ανά δύο ώστε να προκύψουν με την τομή τους οι 4 γωνίες του marker.

Σημαντικό ρόλο σε αυτό το βήμα παίζουν οι “κατεύθυνσεις” των γραμμικών τμημάτων της εικόνας, όπως αυτές ορίστηκαν και υπολογίστηκαν σε προηγούμενο αλγοριθμικό βήμα.

Η “κατεύθυνση” ορίζεται με τη χρήση

- της διεύθυνσης του γραμμικού τμήματος, όπως αυτή υπολογίζεται από τις συντεταγμένες των άκρων του, και
- με την κατεύθυνση της intensity gradient σε κάποιο (οποιοδήποτε) σημείο του, δεδομένου ότι όλα τα σημεία έχουν πανομοιότυπη intensity gradient με μικρές μόνο αποκλίσεις

Έτσι, δύο παράλληλα γραμμικά τμήματα μπορεί να έχουν ίδια διεύθυνση, αλλά η “κατεύθυνσή” τους να διαφέρει κατά 180 μοίρες, όπως φαίνεται και στο σχήμα που ακολουθεί.



Εικόνα 81 - Παράδειγμα “κατεύθυνσεων” των πλευρών ενός τετραπλεύρου

Αυτή η κατεύθυνση αποτελεί, όπως αναμένεται, πολύτιμο εργαλείο κατά την ανίχνευση των markers στην εικόνα, καθώς οι κατεύθυνσεις των γειτονικών πλευρών συνδέονται με κάποια λογική σχέση.

Η λογική αυτή σχέση εξάγεται από το γεγονός ότι τα markers στην εικόνα αποτελούνται από ένα τετράπλευρο με μαύρο περίγραμμα σε άσπρο background. Έτσι, μελετώντας την εκάστοτε edge του marker, αναμένουμε συγκεκριμένες κλίσεις για τις γειτονικές edges στην πλευρά του κάθε άκρου της. Όσες edges δεν συμφωνούν με αυτές τις τιμές, απορρίπτονται αμέσως κατά τον έλεγχο της γειννίαςσης (πχ edge με κατεύθυνση 0° περίπου δεν μπορεί να έχει γειτονική edge με κατεύθυνση 180°).

Επίσης, ανάλογα με την κατεύθυνση των εκάστοτε ζευγών γραμμικών τμημάτων που εξετάζονται για συμβατότητα μεταξύ τους, δεν ελέγχουμε και τις τέσσερις πιθανές ενώσεις τους, αλλά μόνο μία, βρίσκοντας με κάποιο μοντέλο “πρόβλεψης” σε ποιο άκρο κάθε γραμμής θα πρέπει να εστιάσουμε το ενδιαφέρον.

Σαν παράδειγμα παρατίθεται το παρακάτω frame, όπου φαίνεται για κάθε edge που ανιχνεύτηκε η “κατεύθυνσή” και η διεύθυνσή του, στο format

κατεύθυνση⁹⁶ / διεύθυνση⁹⁷



Εικόνα 82 - Οι διευθύνσεις και οι “κατευθύνσεις” όλων των edges σε ένα frame

Με βάση τους παρακάτω κανόνες

Δύο γραμμικά τμήματα είναι **συμβατά** όταν ισχύουν όλες οι παρακάτω συνθήκες

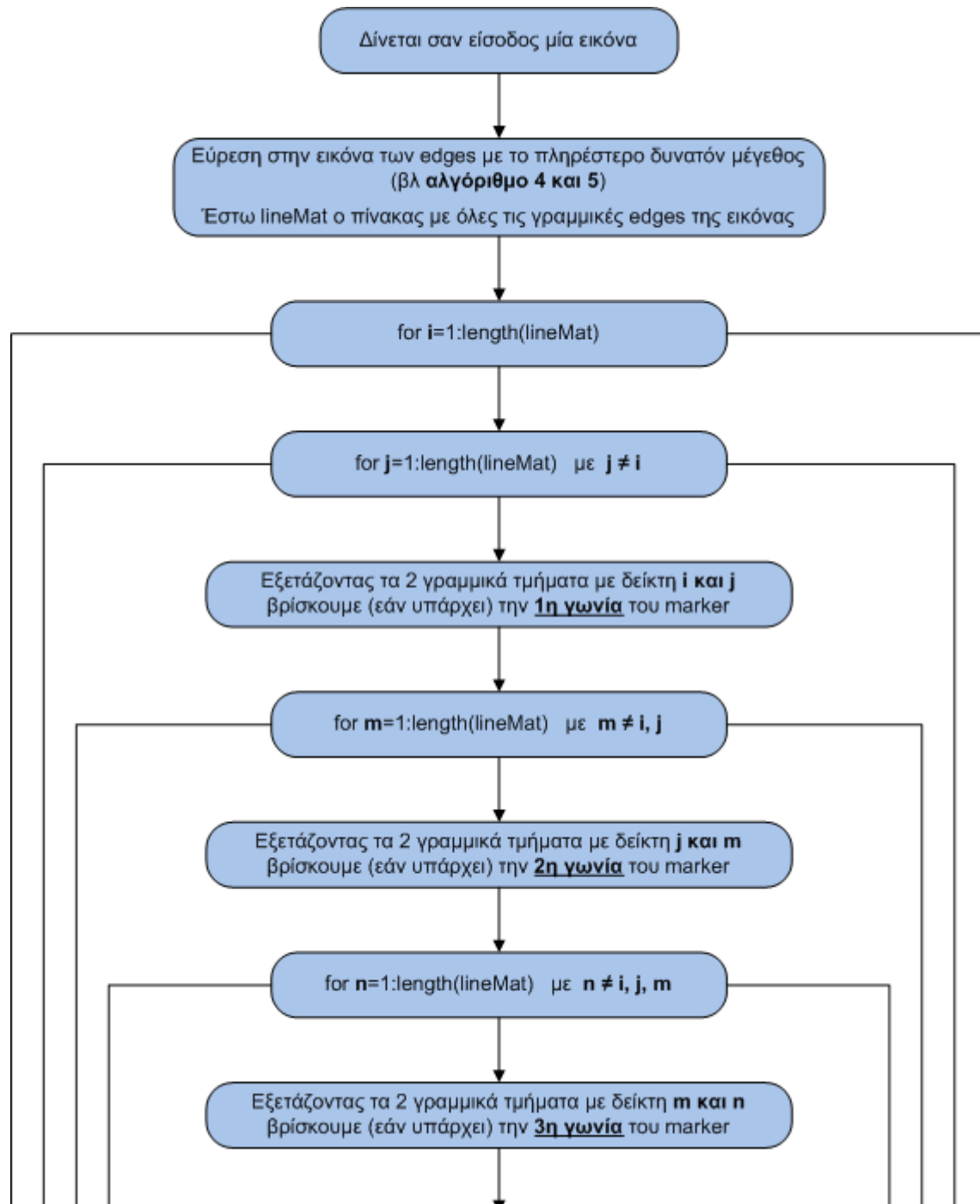
- Και τα δύο έχουν μήκος μεγαλύτερο από ένα προκαθορισμένο κατώφλι
- Η απόσταση των “άκρων ένωσης” των δύο γραμμικών τμημάτων είναι μικρότερη από ένα προκαθορισμένο κατώφλι.
(βλ παρακάτω σχετικά με την πρόβλεψη για τα άκρο κάθε γραμμή που συμμετέχει στην ένωση)
- Οι κατευθύνσεις τους μαρτυρούν ότι δεν είναι σχεδόν παράλληλα
- Οι κατευθύνσεις τους μαρτυρούν ότι είναι γειτονικές edges ενός marker

Εάν 2 γραμμικά τμήματα (edges του marker) είναι συμβατά, τότε βρίσκουμε την **γωνία** του marker που ορίζουν, με βάση την τομή των ευθειών στις οποίες ανήκουν τα γραμμικά τμήματα

⁹⁶ Με πεδίο τιμών $(-90, +90]$

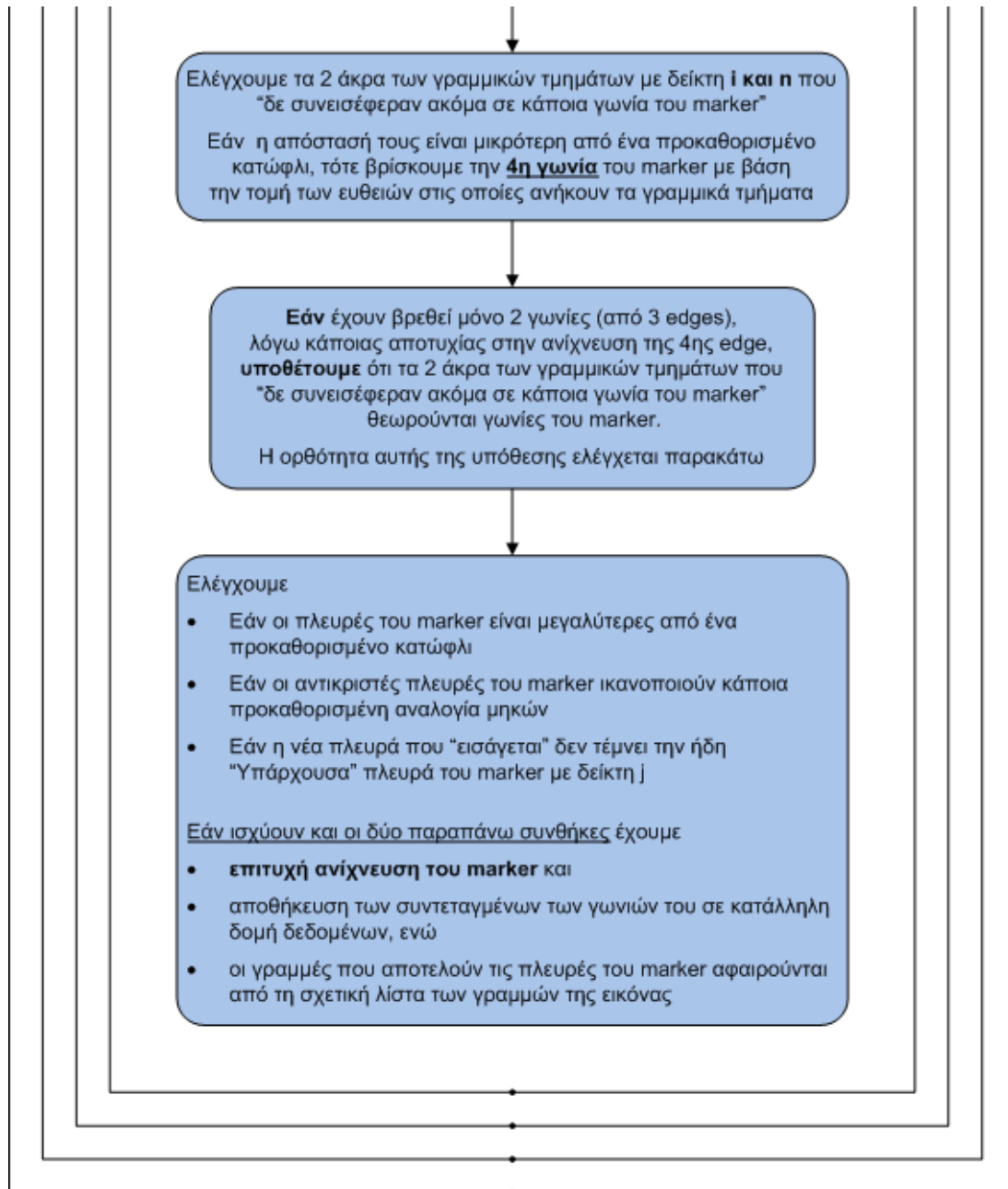
⁹⁷ Με πεδίο τιμών $[0, 360)$

ακολουθεί η παράθεση του προτεινόμενου αλγορίθμου για τον εντοπισμό των περιγραμμάτων και μερικών επεξηγηματικών screenshots.



Αλγόριθμος 6 - Προτεινόμενος αλγόριθμος εντοπισμού του περιγράμματος ενός marker

Συνεχίζεται στην επόμενη σελίδα

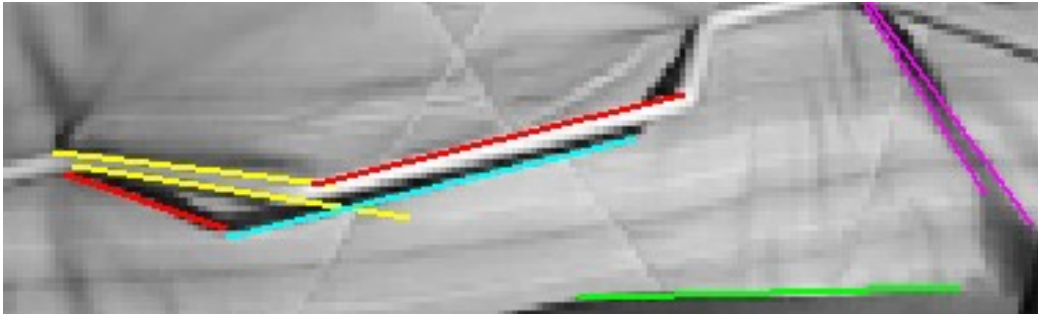


Αλγόριθμος 6 - Προτεινόμενος αλγόριθμος εντοπισμού του περιγράμματος ενός marker

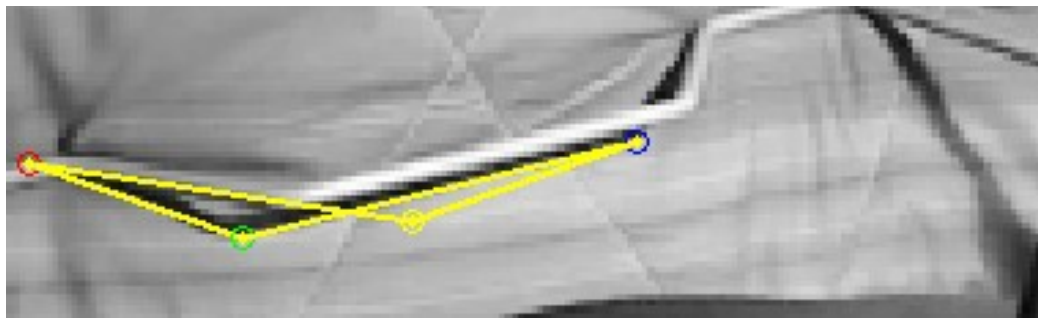
Σε αυτό το σημείο θα πρέπει να τονιστεί κάτι που είναι λίγο δύσκολο να αναφερθεί στο διάγραμμα του αλγορίθμου.

Όταν εξετάζεται εάν δύο γραμμικά τμήματα γειτνιάζουν και είναι συμβατά μεταξύ τους, ώστε να δώσουν μία γωνία του marker, δεν εξετάζονται και τα τέσσερα άκρα των δύο γραμμών (δηλαδή και οι τέσσερις πιθανές ενώσεις τους), αλλά ανάλογα με την κλίση και την "κατεύθυνσή" της κάθε γραμμής εστιάζουμε το ενδιαφέρον μας σε ένα και μοναδικό άκρο της (και άρα ελέγχουμε μόνο έναν τρόπο ένωσης).

Επειδή η χρησιμότητα της παραπάνω τακτικής δεν είναι πολύ εύκολο να περιγραφεί φραστικά, ακολουθεί η αναπαράσταση μιας σπάνιας μεν, πιθανής δε, κατάστασης false positive detection σε περιβάλλον με πολλές ακμές και χρήση (πολύ) χαμηλού κατωφλίου φωτεινότητας⁹⁸ κατά την ανίχνευση των edges (έχουμε εύρεση τριών πλευρών και “εισαγωγής” της τέταρτης, εξού και το κίτρινο περίγραμμα).



Εικόνα 83 - Περίπτωση positive false detection (γραμμές)



Εικόνα 84 - Περίπτωση positive false detection (τετράπλευρο)

Με τη μέθοδο της εστίασης του ενδιαφέροντος μόνο στο (ένα) κατάλληλο άκρο κάθε γραμμής η περίπτωση positive false detection του παραπάνω είδους αποφεύγεται επιτυχώς.

Στην ίδια κατεύθυνση λειτουργεί και το τμήμα ελέγχου⁹⁹ για την τομή των ήδη υπάρχουσών πλευρών του τετραπλεύρου (ενεργοποιήθηκε κι αυτό στην επεξεργασία του παραπάνω frame), κατά την “εισαγωγή” από τον αλγόριθμο μίας επιπλέον edge.

⁹⁸ Κάτι που γενικά δεν ενδείκνυται, ο αλγόριθμος όμως είναι καλό να εμπεριέχει τις κατάλληλες δικλείδες ασφαλείας. Σε περίπλοκο περιβάλλον μπορεί να προκύψει το ίδιο πρόβλημα ακόμα και για υψηλό κατώφλι

⁹⁹ παρουσιάζεται στον παραπάνω αλγόριθμο με το σημάδακι ‘✖’ (στην προηγούμενη σελίδα

Τα παρακάτω screenshots δείχνουν το αποτέλεσμα της εφαρμογής του αλγορίθμου



Εικόνα 85 - Οι edges του marker πριν την ανίχνευση του περιγράμματός του

Σαν αποτέλεσμα έχουμε την εύρεση των 4 γωνιών του marker, που με διαδοχική ένωσή τους δίνουν και το τετράπλευρο περιγράμματά του.



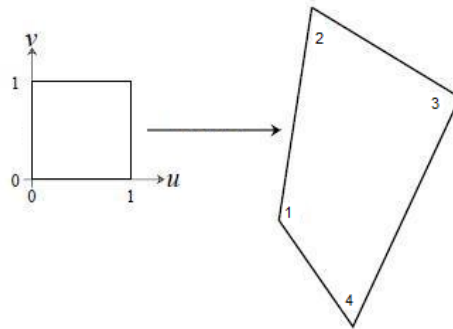
Εικόνα 86 - Ανίχνευση του περιγράμματος του marker

Ταυτοποίηση του ID των markers

Για την αποκωδικοποίηση του εσωτερικού του marker, απαιτείται η σωστή ανάγνωσή του με τη χρήση scanlines ανάγνωσης, παρόλη της προοπτική “παραμόρφωση” που μπορεί να έχει υποστεί.

Δειγματοληψία του marker

Προς αυτή την κατεύθυνση χρησιμοποιείται projective mapping από μοναδιαίο τετράγωνο προς γενικό τετράπλευρο

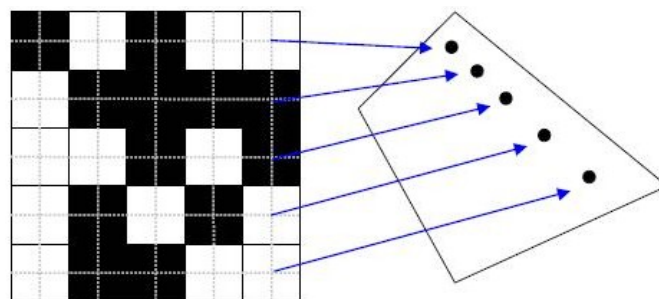


Εικόνα 87 - Projective mapping στον προτεινόμενο αλγόριθμο

Η μέθοδος αυτή παρουσιάζεται αναλυτικά στο σχετικό κεφάλαιο περί μαθηματικού υποβάθρου (βλ εικόνα 49, περίπτωση 1)

Ο λόγος που το projective mapping γίνεται από μοναδιαίο τετράγωνο προς γενικό τετράπλευρο, είναι γιατί στο μοναδιαίο τετράγωνο-marker είναι πολύ εύκολο να θεωρήσουμε το πλέγμα των scanlines ανάγνωσης και να βρούμε τις συντεταγμένες των τομών των scanlines, ανάλογα φυσικά με τη δομή του marker (πλήθος cells σε κάθε διεύθυνση).

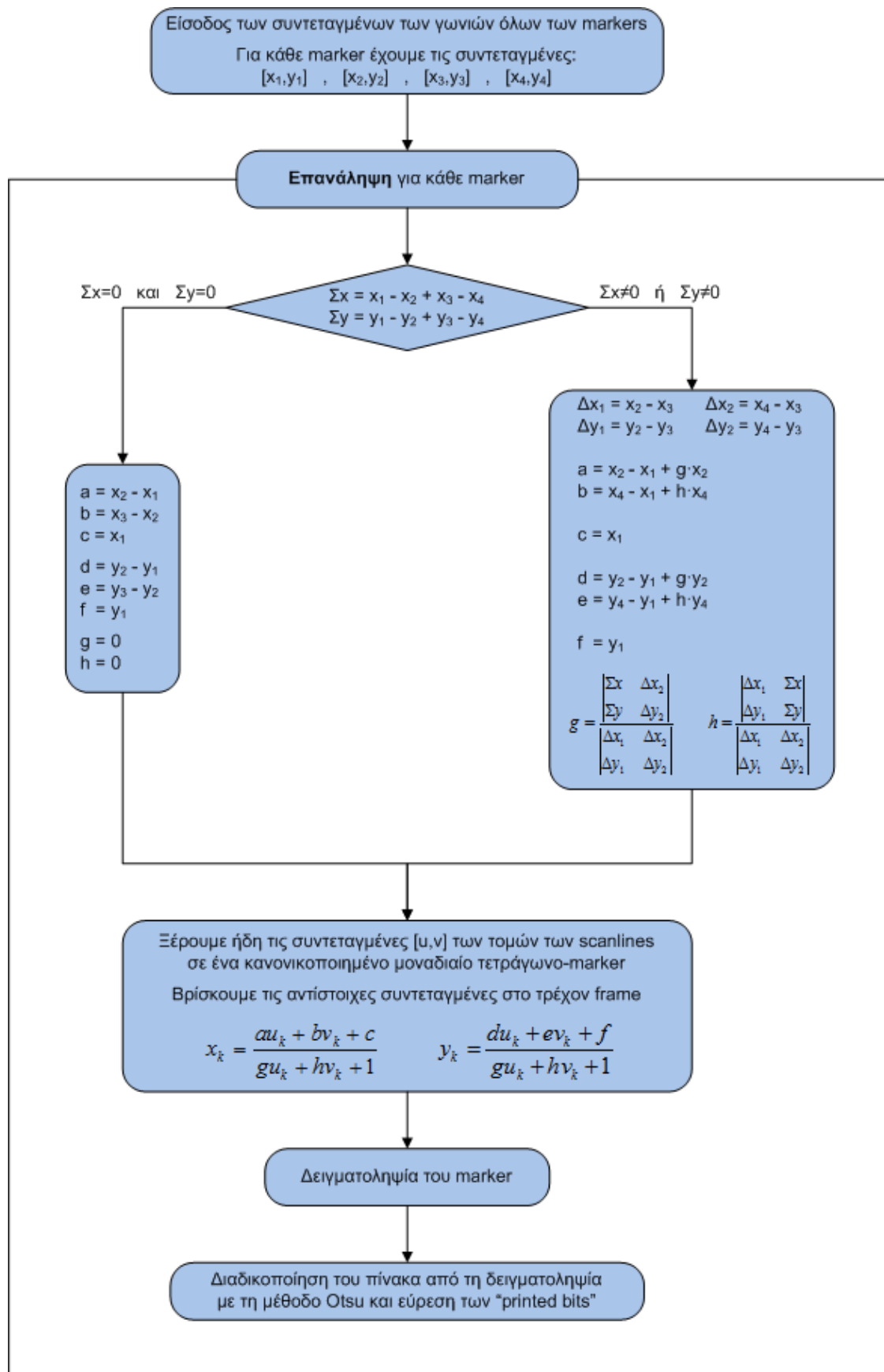
Με το projective mapping βρίσκουμε εν συνεχεία τις συντεταγμένες αυτές στο προοπτικά “παραμορφωμένο” marker στο επεξεργαζόμενο frame, δειγματοληπτούμε τις τιμές φωτεινότητας που ενδιαφέρουν (με απλή ανάγνωση της φωτεινότητας των pixels) και με δυαδικοποίηση με τη μέθοδο του Otsu¹⁰⁰ “αποκαθιστούμε” το δυαδικό περιεχόμενο του marker, βρίσκοντας τα “printed bits”.



Εικόνα 88 - Αντιστοίχιση σημείων από μοναδιαίο τετράγωνο προς τετράπλευρο στην εικόνα

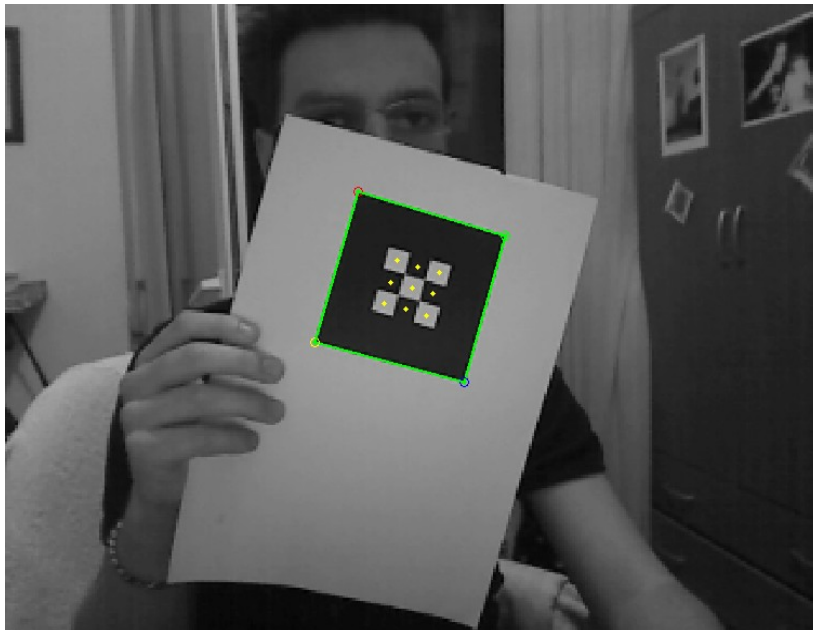
¹⁰⁰ προσφέρεται σαν built-in συνάρτηση στο Matlab - συνάρτηση “graythresh”

Ακολουθεί ο προτεινόμενος αλγόριθμος και μερικά επεξηγηματικά screenshots.



Αλγόριθμος 7 - Προτεινόμενος αλγόριθμος ανάγνωσης-δειγματοληψίας των "printed bits" ενός marker

Οι 4 γωνίες που βρέθηκαν από τον προηγούμενο αλγόριθμο χρησιμοποιούνται για τη σωστή δειγματοληψία του εσωτερικού του marker, για την ανάγνωση του pattern του και την αποκωδικοποίησή του

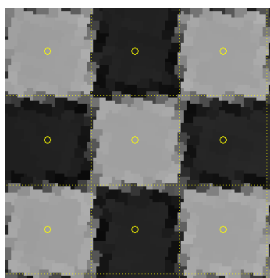


Εικόνα 89 - Το περίγραμμα του marker και η δειγματοληψία του εσωτερικού του

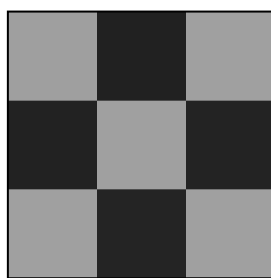
Στην εικόνα 90 φαίνονται τα σημεία όπου γίνεται η δειγματοληψία, με την τομή των scanline ανάγνωσης. Η εικόνα αυτή προέρχεται από την εικόνα 89 μετά από projective mapping, ώστε να “ανακατασκευάσουμε” (απλά για τις ανάγκες της παρουσίασης) το τετράγωνο περιεχόμενο του marker.

Αποκωδικοποίηση του marker

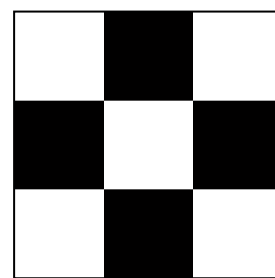
Αμέσως μετά την δειγματοληψία (εικόνας 89, 90) παίρνουμε έναν 2D πίνακα ακεραίων αριθμών που αντιπροσωπεύουν τη φωτεινότητα του κάθε cell του marker (εικόνα 91).



Εικόνα 90 - Δειγματοληψία στο εσωτερικό του marker (canonical view)



Εικόνα 91 - Το αποτέλεσμα της δειγματοληψίας του εσωτερικού του marker



Εικόνα 92 - Εύρεση των “printed bits” του marker με δυαδικοποίηση (Otsu)

Με τη χρήση της μεθόδου του Otsu μετατρέπουμε αυτό τον πίνακα σε δυαδικό (εικόνα 92), βρίσκοντας με αυτό τον τρόπο τα “printed bits” που “εκτυπώθηκαν” στο marker.

Τα “printed bits” είναι αυτά που θα αποκωδικοποιηθούν εύκολα από το κατάλληλο αλγοριθμικό module, ώστε να βρούμε το αντίστοιχο ID.

Ενώνοντας τα κομμάτια: Ο συνολικός αλγόριθμος

Όπως θα έγινε κατανοητό μέχρι αυτό το σημείο, ο αλγόριθμος που παρουσιάζεται στην εργασία δεν αναπτύχθηκε με “μονολιθική” λογική.

Απεναντίας, αποτελείται από επιμέρους ανεξάρτητες αλγοριθμικές ενότητες¹⁰¹ με σαφώς διακριτό ρόλο. Όλες όμως οι ενότητες φτιάχνονται ακολουθώντας κάποιο αυστηρό “πρωτόκολλο επικοινωνίας” μεταξύ τους, αναφορικά με τα ορίσματα εισόδου και εισόδου της καθεμιάς, ώστε να εξασφαλίζεται η αποτελεσματική λειτουργία του συνολικού αλγορίθμου.

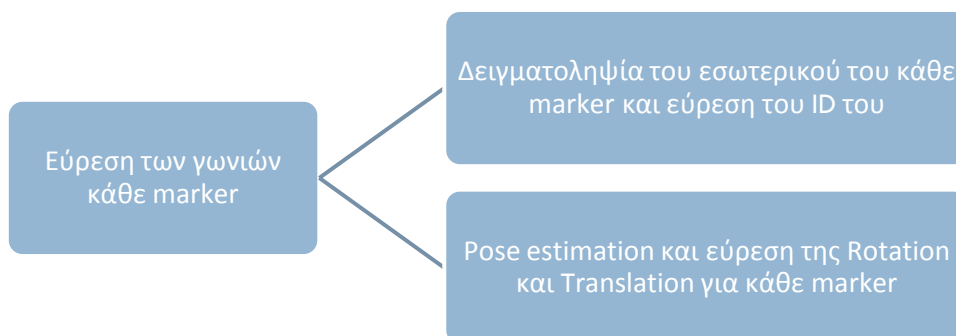
Με αυτόν τον τρόπο μπορούν να γίνουν πολύ ευκολότερα αλλαγές-βελτιώσεις σε κάθε ενότητα ή ακόμα και να αντικατασταθεί πλήρως κάποια από αυτές, χωρίς να επηρεάζεται ο συνολικός αλγόριθμος¹⁰².

Έτσι, μπορούμε να κάνουμε λόγο για μία αλγοριθμική ιεραρχία.

- Αρχικά κάθε ενότητα μεταφέρει τον έλεγχο του προγράμματος στην ενότητα του αμέσως χαμηλότερου επιπέδου, ζητώντας της κάποια δεδομένα που της είναι απαραίτητα για τη διεκπεραίωση της αποστολής της.
- Όταν τελειώσει η επεξεργασία στο κατώτατο επίπεδο, επιστρέφονται τα κατάλληλα δεδομένα στο αμέσως ανώτερο, οπότε αυτό μπορεί να φέρει πλέον εις πέρας το ρόλο του. Έπειτα, επιστρέφει κι αυτό με τη σειρά του τα δικά του αποτελέσματα “προς τα πάνω” και η όλη διαδικασία συνεχίζεται ως το ανώτατο επίπεδο.

Στην εικόνα 94 φαίνεται παραστατικότερα αυτή η αλγοριθμική ακολουθία, καθώς και ο ρόλος της κάθε αλγοριθμικής ενότητας. Στην κορυφή της πυραμίδας βρίσκεται το ανώτατο επίπεδο, ενώ στη βάση της βρίσκεται το κατώτατο.

Στο ανώτατο επίπεδο, αφού βρεθούν τα τετράπλευρα περιγράμματα των markers, παρουσιάζεται μία νέα “υπολογιστική αλυσίδα”, όπως φαίνεται στην εικόνα 93

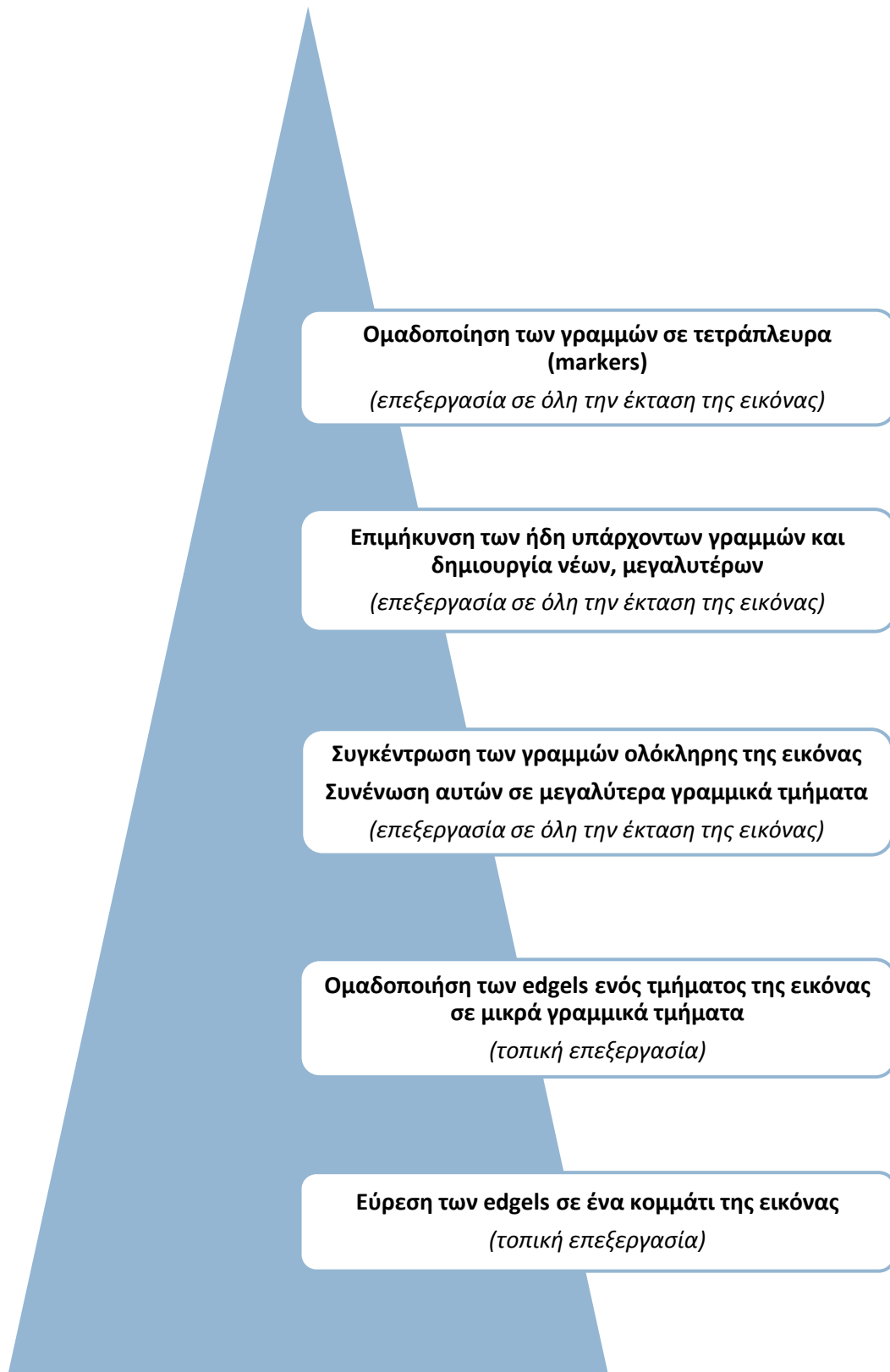


Εικόνα 93 - Το ανώτατο επίπεδο της αλγοριθμικής ακολουθίας

¹⁰¹ Οι αλγοριθμικές ενότητες που παρουσιάστηκαν νωρίτερα στο παρών κεφάλαιο

¹⁰² Εφόσον βέβαια τηρείται το “πρωτόκολλο επικοινωνίας” που αναφέρθηκε νωρίτερα

Όπως φαίνεται στην ακόλουθη εικόνα, στα δύο χαμηλότερα επίπεδα δε γίνεται επεξεργασία σε όλη την έκταση της εικόνας. Απεναντίας, η εικόνα χωρίζεται σε ένα πλήθος μη επικαλυπτόμενων τμημάτων και η επεξεργασία γίνεται σε αυτά.



Εικόνα 94 - Αλγοριθμική ιεραρχία

6. Πειραματική εφαρμογή

Για τη χρησιμοποίηση των αλγορίθμων που παρουσιάζονται στην εργασία, προτείνεται μία πειραματική εφαρμογή με χρήση διαφόρων ειδών markers (δεδομένα ή απλό ID).

Η εφαρμογή αναπτύχθηκε με χρήση του Matlab και μίας webcam (Logitech QuickCam Pro for Notebooks¹⁰³).

Εξεταζόμενο σενάριο εφαρμογής

Πιο συγκεκριμένα, εξετάζεται το σενάριο μίας προσωπικής κάρτας ασθενών σε κάποιο νοσοκομειακό κέντρο. Στη μία πλευρά της κάρτας υπάρχει ένα marker που περιέχει κωδικοποιημένες, σαν αλφαριθμητικά δεδομένα, μερικές προσωπικές πληροφορίες του ασθενούς, ενώ στην άλλη υπάρχει ένα marker που περιέχει κωδικοποιημένο ένα αριθμητικό ID και χρησιμεύει για την παρουσίαση ιατρικών δεδομένων, όπως τρισδιάστατων MRI scans, με τη χρήση μεθόδων επαυξημένης πραγματικότητας.

Ακολουθεί περιγραφή για κάθε είδος marker ξεχωριστά.

Marker δεδομένων

Αν και η εργασία εστιάζει στα markers που κωδικοποιούν ένα απλό αριθμητικό ID (index-based 2d barcodes), αναπτύχθηκε σαν παράδειγμα χρήσης και ένας κωδικοποιητής/αποκωδικοποιητής για markers αλφαριθμητικών δεδομένων (database 2d barcodes).

Πιο συγκεκριμένα, ενδιαφερόμαστε στην κωδικοποίηση ορισμένων απλών δεδομένων ενός ασθενούς:

- Όνομα
- Επίθετο
- Ομάδα αίματος

Σχετικά με το μέγεθος του marker, επιλέχθηκε μέγεθος πλευράς 14 printed bits, ώστε να υπάρχει δυνατότητα κωδικοποίησης αρκετών bits.

Έτσι, με marker πλευράς 14 bits έχουμε δυνατότητα κωδικοποίησης 196 bits, που χρησιμοποιούνται ως εξής:

- Τα 4 γωνιακά bits χρησιμοποιούνται σαν bits προσανατολισμού (orientation bits), έτσι ώστε να μπορούμε να αποκωδικοποιήσουμε σωστά το marker παρά την πιθανή περιστροφή αυτού ως προς την κάμερα.
- Για την ομάδα αίματος έχουμε 8 δυνατές τιμές (A+/A-/B+/B-/AB+/AB-/O+/O-), οπότε χρησιμοποιούνται 3 bits για την επιλογή της κατάλληλης ομάδας¹⁰⁴

¹⁰³ http://www.logitech.com/index.cfm/notebook_products/webcams/devices/3055&cl=au,en

- Τα υπόλοιπα bits χρησιμοποιούνται για την κωδικοποίηση αλφαριθμητικών χαρακτήρων, με τη χρήση ASCII κώδικα (7 bits για κάθε χαρακτήρα -βλ εικόνα 95 για παράδειγμα). Έτσι, μπορούν να κωδικοποιηθούν 27 διαφορετικοί χαρακτήρες¹⁰⁵ για το ονοματεπώνυμο

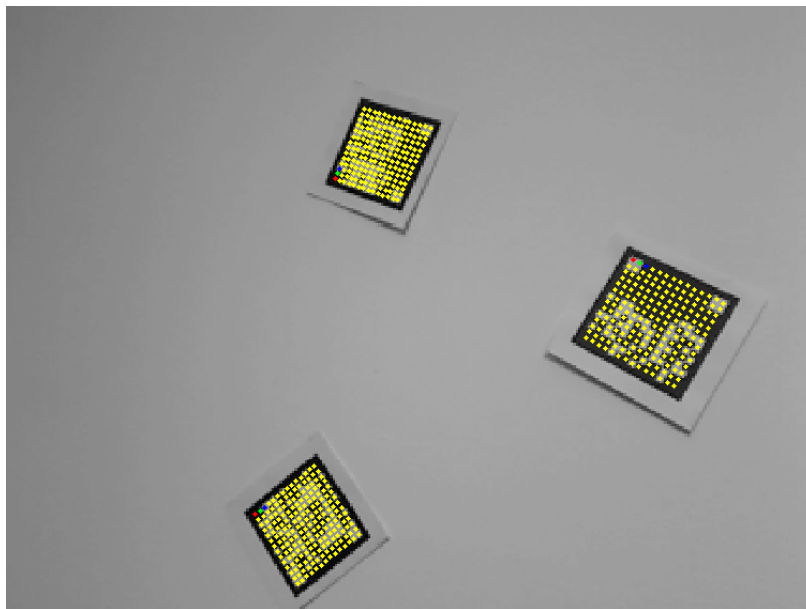
Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
100 0001	101	65	41	A	110 0001	141	97	61	a
100 0010	102	66	42	B	110 0010	142	98	62	b
100 0011	103	67	43	C	110 0011	143	99	63	c

Εικόνα 95 - Παράδειγμα κώδικα ASCII

Αν και δεν ενδείκνυται, δε χρησιμοποιήθηκε κάποια μέθοδος error correction για τα markers δεδομένων, καθώς η ανάπτυξή τους έγινε απλά σαν παρουσίαση της δυνατότητας αυτού του είδους κωδικοποίησης. Όλα τα συστήματα marker αυτού του είδους που έχουν επικρατήσει, όμως, χρησιμοποιούν κάποιο είδος error correction (συνήθως reed-solomon).

Παρόλα αυτά, παρατηρήθηκε αποτελεσματική ανάγνωση και αποκωδικοποίηση ακόμα και υπό μη ιδανικές φωτιστικές συνθήκες, ίσως λόγω του autogain και της καλής ποιότητας του φακού της κάμερας που χρησιμοποιήθηκε. Η απόσταση της ανάγνωσης των δεδομένων ήταν ικανοποιητική για το σενάριο εφαρμογής που εξετάζεται, ακόμα και με χαμηλή ανάλυση (320x240) και αραιό πλέγμα scanlines (απόσταση 5 pixels).

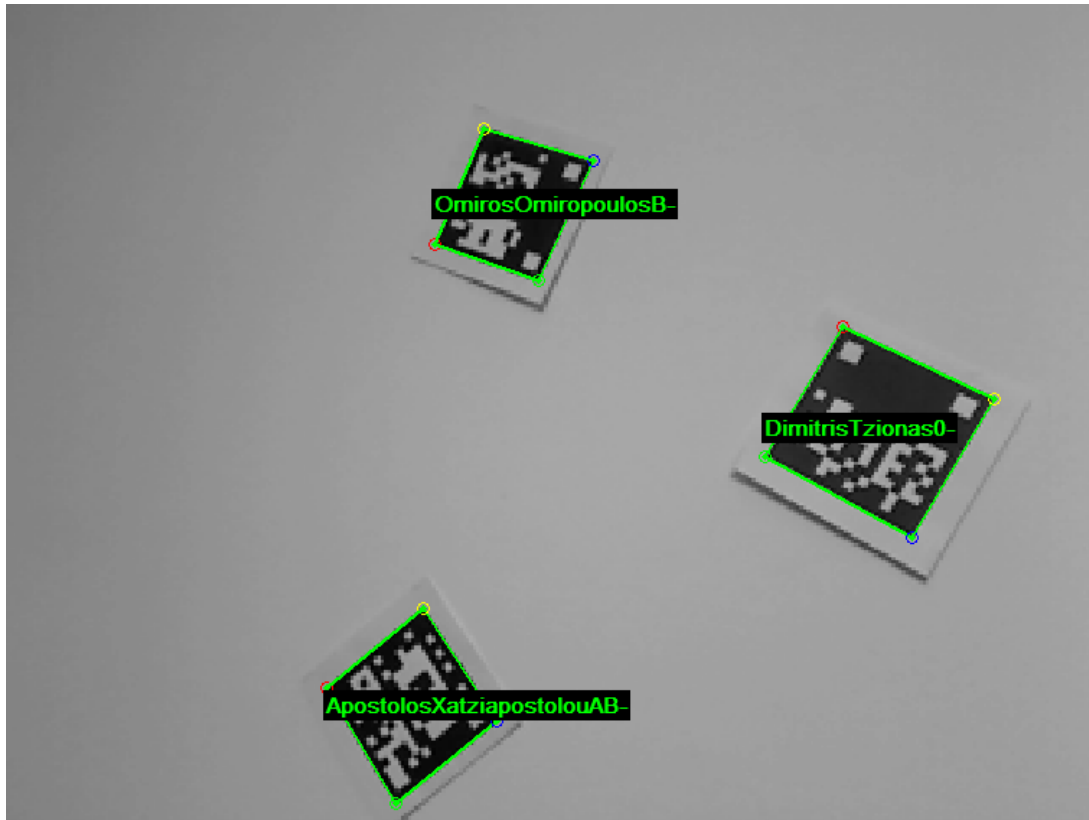
Παρακάτω παρουσιάζεται η αποτελεσματική ανίχνευση, δειγματοληψία και αποκωδικοποίηση μερικών markers δεδομένων στο ίδιο frame



Εικόνα 96 - Markers δεδομένων - Δειγματοληψία του εσωτερικού τους

¹⁰⁴ Τα 3 bits δίνουν $2^3=8$ διαφορετικούς συνδιασμούς

¹⁰⁵ Έχουμε δυνατότητα κωδικοποίησης $\frac{14^2-4-3}{7} = 27$ χαρακτήρων



Εικόνα 97 - Markers δεδομένων - Αποκωδικοποίηση και παρουσίαση των δεδομένων

Τα αποτελέσματα της αποκωδικοποίησης παρουσιάζονται στη μορφή

ΌνομαΕπίθετοΟμαδαίματος

Η σημαντικότερη διαφορά σε σχέση με κάποιον απλό κωδικοποιητή ενός συνηθισμένου τύπου database markers, όπως για παράδειγμα ενός QR-Code reader¹⁰⁶, είναι ότι

- με τη μέθοδο που αναπτύχθηκε έχουμε επιτυχή αναγνώριση πολλαπλών markers σε ένα frame, χωρίς την απαίτηση να έχουμε ένα και μοναδικό marker που να καλύπτει σχεδόν εξολοκλήρου το frame.
- δίνεται η δυνατότητα ανάγνωσης ακόμα και υπό όχι και τόσο ιδανικές φωτιστικές συνθήκες, καθώς και όταν το marker βρίσκεται υπό γωνία σε σχέση με το marker, εισάγοντας κάποια σημαντική προοπτική παραμόρφωση (οι απλοί readers απαιτούν το marker να βρίσκεται σε επίπεδο σχεδόν παράλληλο με το επίπεδο της κάμερας).

Με την έλλειψη όμως του error correction, όταν αυξάνεται αρκετά η απόσταση ανάγνωσης, αρχίζει σταδιακά η εμφάνιση σφαλμάτων σε ορισμένους χαρακτήρες

¹⁰⁶ Χρησιμοποιήθηκε ο reader της kaywa (<http://reader.kaywa.com/>) σε ένα κινητό τηλέφωνο Sony-Ericsson w610i για την διαπίστωση των δυνατοτήτων που παρέχονται από έναν συνηθισμένο reader

Markers απλού ID

Το σύστημα markers στο οποίο εστιάζει κυρίως η εργασία, είναι τα markers ενός απλού ID (index-based 2d barcodes). Σε κάθε marker κωδικοποιείται ένα απλό αριθμητικό ID, το οποίο κατά την αποκωδικοποίηση χρησιμοποιείται για την άντληση πληροφοριών κάθε είδους από κάποια βάση δεδομένων¹⁰⁷.

Για τη δημιουργία-κωδικοποίηση των markers αναπτύχθηκαν ορισμένες εφαρμογές, για διάφορα μεγέθη και για διαφορετικές κωδικοποιήσεις.

Για οικονομία χώρου παρακάτω παρουσιάζεται μόνο η εφαρμογή δημιουργίας markers μεγέθους 6x6 “printed bits” και με χρήση του κώδικα εντοπισμού και διόρθωσης σφαλμάτων BCH.

Όπως φαίνεται και στην εικόνα, έχουμε

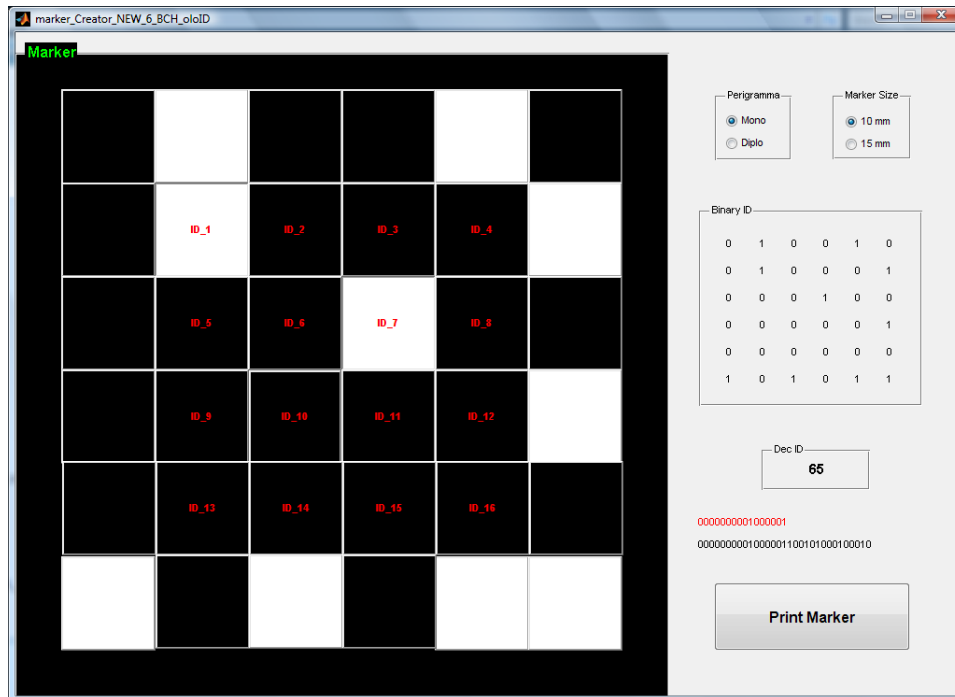
- 4 γωνιακά *bits* προσανατολισμού (πράσινο χρώμα των λέξεων περιγραφής),
- 16 *bits* πληροφορίας (κόκκινο χρώμα) που δίνουν 65536 διαφορετικά IDs
- 15 *parity bits* για τον εντοπισμό και τη διόρθωση έως και 3 σφαλμάτων στα bits πληροφορίας



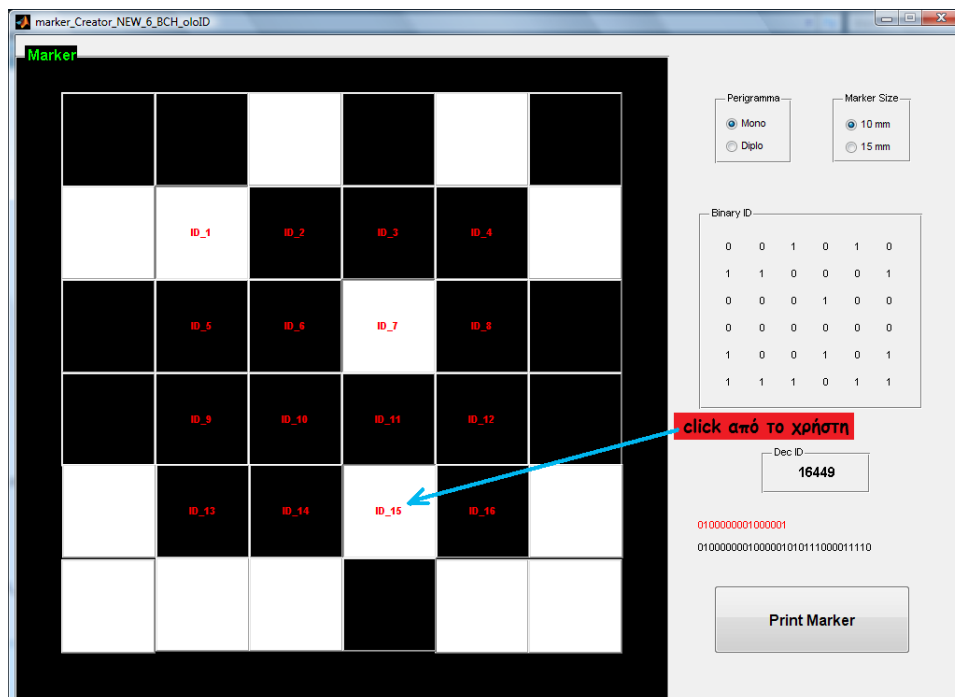
Εικόνα 98 - Εφαρμογή για την ανάπτυξη markers μεγέθους 6x6 με BCH(31,16)

¹⁰⁷ Το ID χρησιμεύει δηλαδή σαν κλειδί για τα ερωτήματα προς τη βάση δεδομένων

Ο χρήστης κάνοντας click πάνω στα bits πληροφορίας δημιουργεί το marker της αρεσκείας του, ενώ τα bits για τη διόρθωση των λαθών αλλάζουν αυτόματα, όπως φαίνεται στο παρακάτω ζεύγος εικόνων.



Εικόνα 99 - Δημιουργία ενός 6x6 BCH marker (1^η εικόνα)



Εικόνα 100 - Δημιουργία ενός 6x6 BCH marker (2^η εικόνα - προσθήκη ενός επιπλέον bit πληροφορίας)

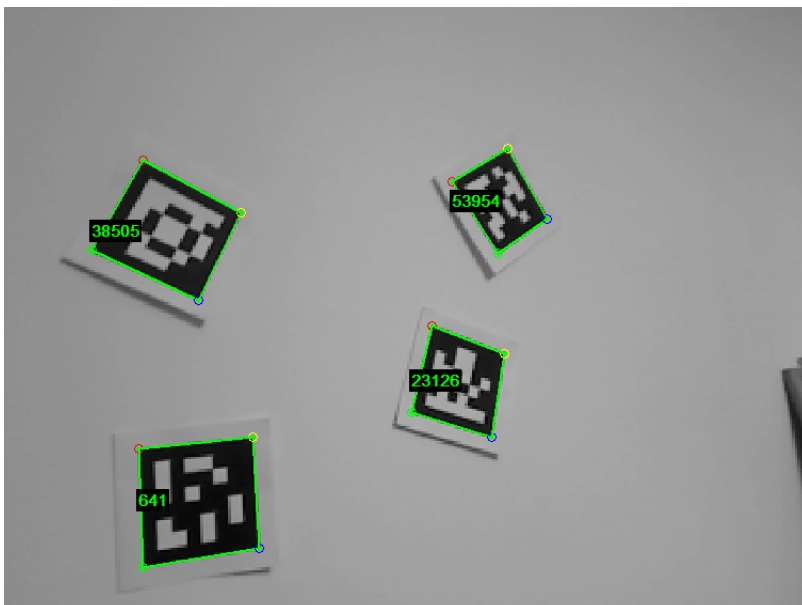
Αυτή η εφαρμογή είναι βολική στα πλαίσια της εργασίας για τη δημιουργία αρκετά διαφορετικών markers κατά βούληση. Σε κάποια πρακτική εφαρμογή θα ταίριαζε ασφαλώς

μία αυτοματοποιημένη δημιουργία markers, με βάση έναν αύξοντα κωδικό που θα αποτελούσε και τον προς κωδικοποίηση κωδικό-ταυτότητα του ασθενούς.

Για τη διόρθωση των σφαλμάτων επιλέχθηκε ο κώδικας BCH(31,16) έτσι ώστε να έχουμε υπεραρκετό πλήθος διαφορετικών IDs για την εφαρμογή, ενώ η δυνατότητα διόρθωσης έως και τριών σφαλμάτων είναι μάλλον επαρκής, δεδομένου ότι στο εξεταζόμενο σενάριο αναμένονται σχετικά ελεγχόμενες συνθήκες φωτισμού και σχετικά μικρή απόσταση ανάγνωσης.



Εικόνα 101 - Δειγματοληψία μερικών 6x6 BCH υπό καλές φωτιστικές συνθήκες



Εικόνα 102 - Ανίχνευση και αποκωδικοποίηση μερικών 6x6 BCH υπό καλές φωτιστικές συνθήκες



Εικόνα 103 - Δειγματοληψία μερικών 6x6 BCH υπό κακές φωτιστικές συνθήκες



Εικόνα 104 - Ανίχνευση και αποκωδικοποίηση μερικών 6x6 BCH υπό κακές φωτιστικές συνθήκες

Παραπάνω παρατίθενται δύο ζεύγη εικόνων για τον εντοπισμό, τη δειγματοληψία και την αποκωδικοποίηση μερικών 6x6 BCH markers στο ίδιο frame και υπό διαφορετικές φωτιστικές συνθήκες.

Στην περίπτωση των κακών φωτιστικών συνθηκών η ανάγνωση είναι επιτυχής, παρά τη χρησιμοποίηση χαμηλής ανάλυσης κάμερας (320x240) και αραιού πλέγματος scanlines (απόσταση 5 pixels μεταξύ τους)

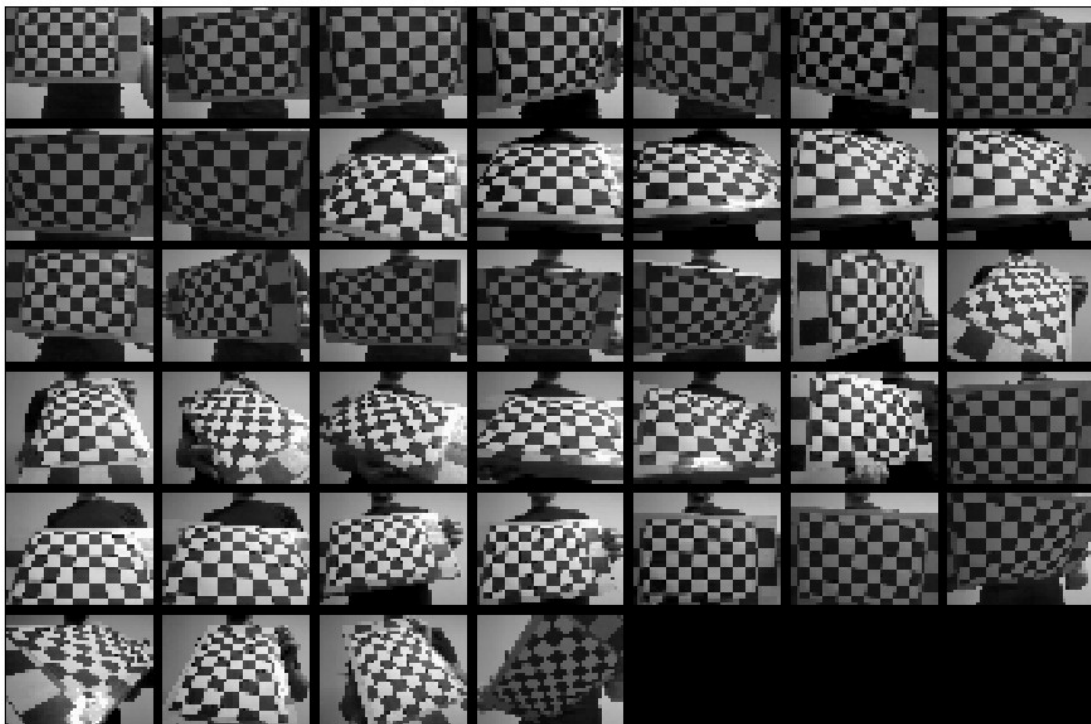
Camera Calibration

Για την εύρεση των εσωτερικών παραμέτρων της κάμερας επιλέχθηκε η απλή, έτοιμη και αποτελεσματική λύση του Camera Calibration Toolbox for Matlab [80].

Με τη χρήση του κάνουμε offline camera calibration κατά το στάδιο της ανάπτυξης της εφαρμογής. Πιο συγκεκριμένα κάνουμε ξεχωριστό calibration για κάθε διαφορετικό μοντέλο κάμερας που μπορεί να χρησιμοποιηθεί στην αναπτυσσόμενη εφαρμογή, καθώς και για κάθε διαφορετική ανάλυση αυτών των καμερών¹⁰⁸.

Το αποτέλεσμα είναι ένα ξεχωριστό αρχείο για κάθε κάμερα και κάθε ανάλυση, που περιέχει τις αντίστοιχες εσωτερικές παραμέτρους. Ανάλογα με τον εξοπλισμό και τις επιλογές του χρήστη φορτώνεται κατά την έναρξη της εφαρμογής το ανάλογο αρχείο. Έτσι οι εσωτερικές παράμετροι είναι γνωστές, ώστε να είναι δυνατή σε μεταγενέστερο στάδιο η εύρεση των εξωτερικών παραμέτρων.

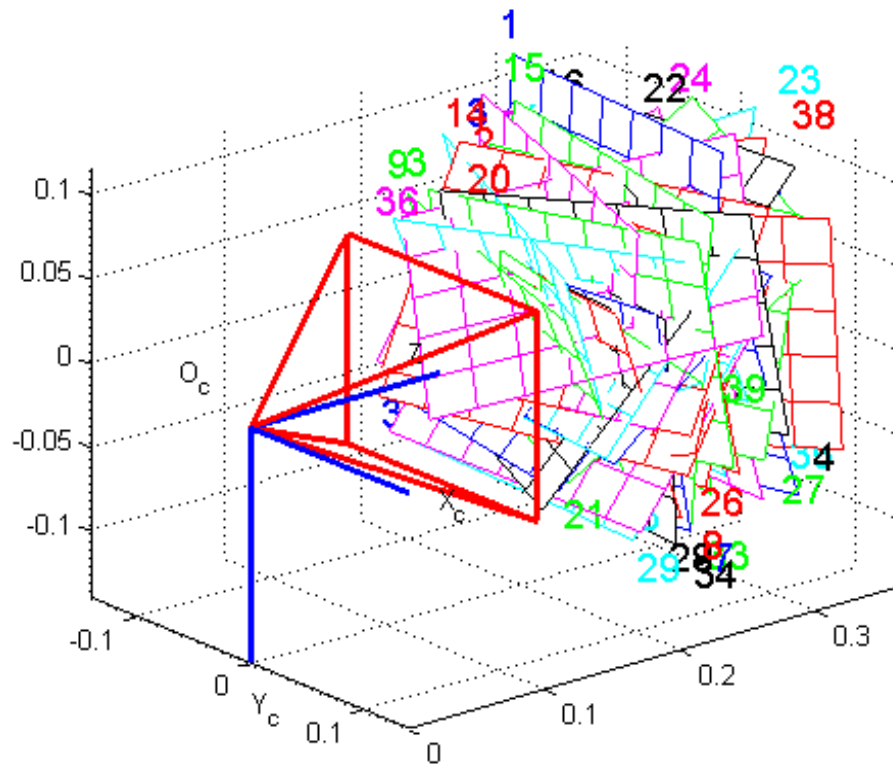
Για τη διεξαγωγή του camera calibration απαιτείται η χρήση μίας απλής, επίπεδης εικόνας εν είδει ασπρόμαυρης σκακιέρας, που μπορεί να εκτυπωθεί με έναν κοινό εκτυπωτή. Ο χρήστης βγάζει αρκετές φωτογραφίες αυτής της εικόνας, σε διάφορες αποστάσεις και κλίσεις, και στη συνέχεια απλά υποδεικνύει στο software του calibration τα άκρα της σκακιέρας για κάθε εικόνα. Οι όλοι υπολογισμοί γίνονται από εκεί και πέρα αυτόματα (γι' αυτό και η όλη διαδικασία χαρακτηρίζεται ημιαυτόματη).



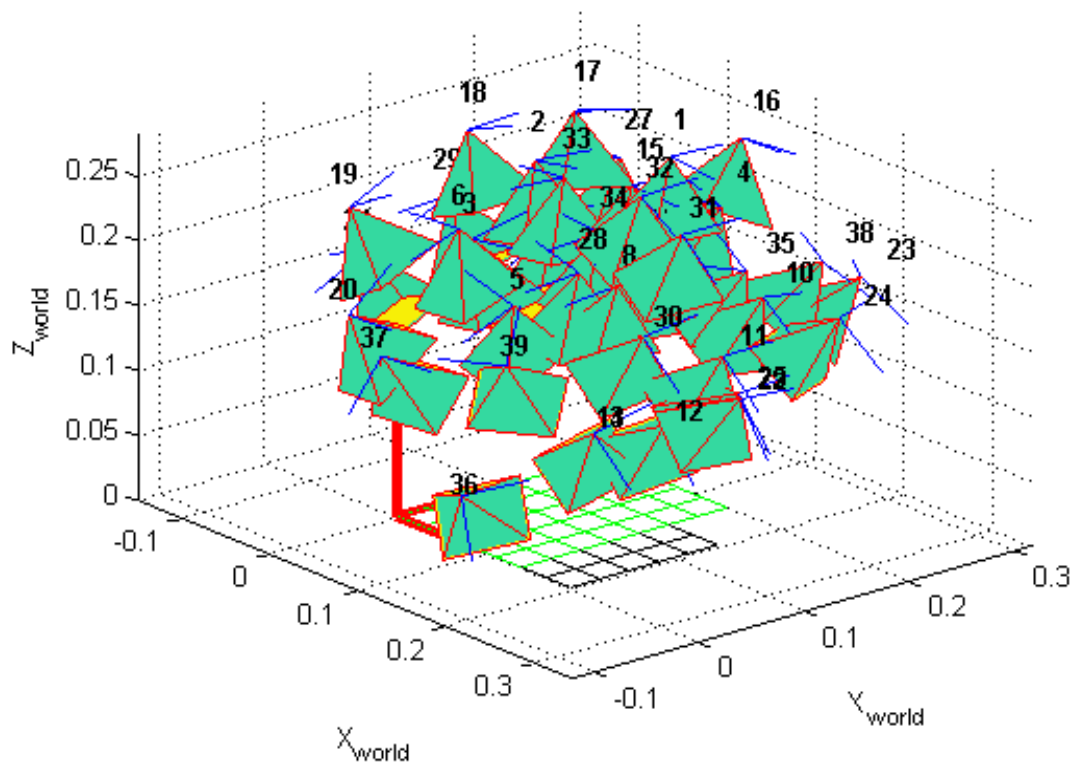
Εικόνα 105 - Οι διάφορες όψεις της "σκακιέρας" για το calibration σε ανάλυση 320x240

¹⁰⁸ Στη συγκεκριμένη εφαρμογή έγινε calibration της κάμερας Logitech QuickCam Pro for Notebooks, για ανάλυση 320x240 και 80x600

Τα παρακάτω screenshots δείχνουν τις εξωτερικές παραμέτρους για κάθε φωτογραφία από το calibration set που παρουσιάστηκε παραπάνω. Έτσι βλέπουμε παραστατικά τη δυνατότητα εύρεσης της ακριβούς θέσης της κάμερας ως προς το φωτογραφιζόμενο αντικείμενο (εικόνα 107) και το αντίστροφο (εικόνα 106).



Εικόνα 106 - Οι εξωτερικές παράμετροι του calibration set ως προς την κάμερα



Εικόνα 107 - Οι εξωτερικές παράμετροι του calibration set ως προς το WCS

Το τελικό αποτέλεσμα της εύρεσης των εσωτερικών παραμέτρων και για τις δύο αναλύσεις παρουσιάζεται αμέσως παρακάτω, όπου φαίνεται ότι η εκτίμηση των παραμέτρων γίνεται με πολύ καλή ακρίβεια.

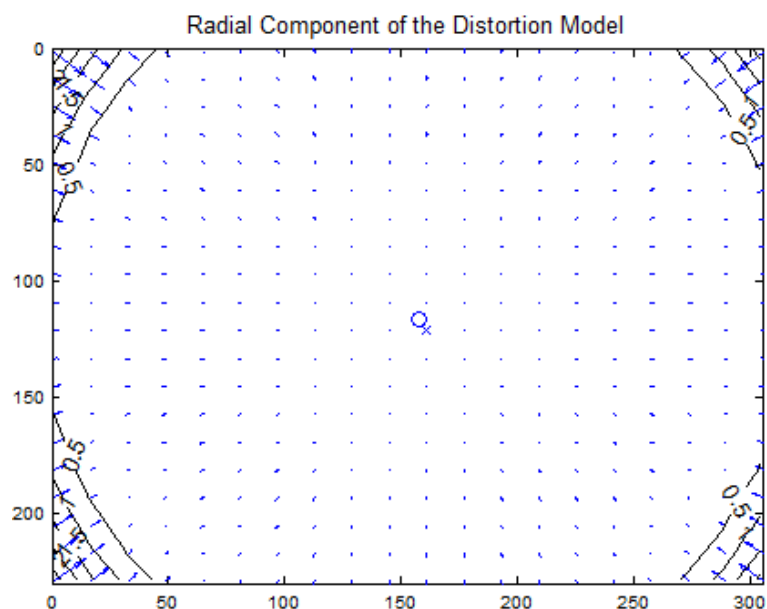
Pixel error	= [0.2471, 0.2849]	
Focal Length	= (269.034, 267.584)	+/- [0.7464, 0.8079]
Principal Point	= (156.323, 115.093)	+/- [1.043, 0.8158]
Skew	= 0	+/- 0
Radial coefficients	= (0.03432, -0.1199, 0)	+/- [0.009287, 0.0238, 0]
Tangential coefficients	= (0.001253, 0.0003624)	+/- [0.001009, 0.001307]

Εικόνα 108 - Εσωτερικές παράμετροι μετά το calibration για ανάλυση 320x240

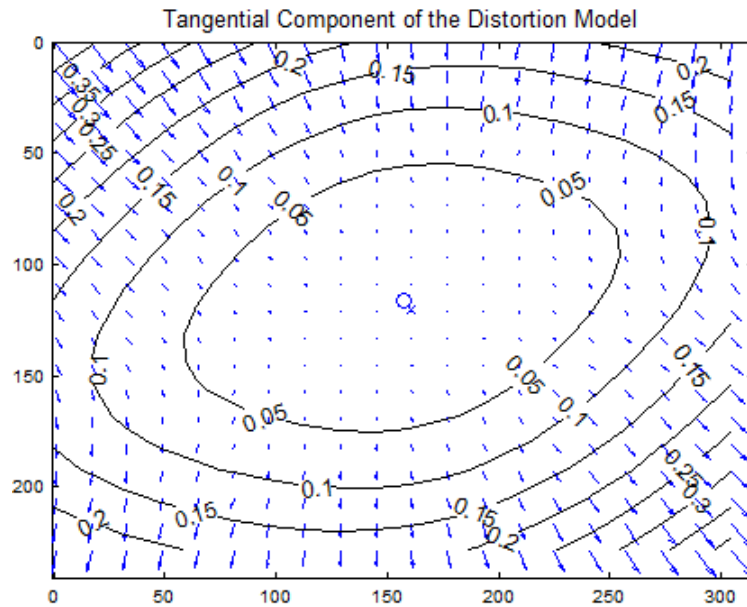
Pixel error	= [0.4719, 0.6218]	
Focal Length	= (665.432, 669.208)	+/- [1.529, 1.857]
Principal Point	= (390.219, 283.372)	+/- [2.146, 1.764]
Skew	= 0	+/- 0
Radial coefficients	= (0.03188, -0.1095, 0)	+/- [0.007969, 0.02213, 0]
Tangential coefficients	= (-0.001233, -0.001543)	+/- [0.0009285, 0.001128]

Εικόνα 109 - Εσωτερικές παράμετροι μετά το calibration για ανάλυση 800x600

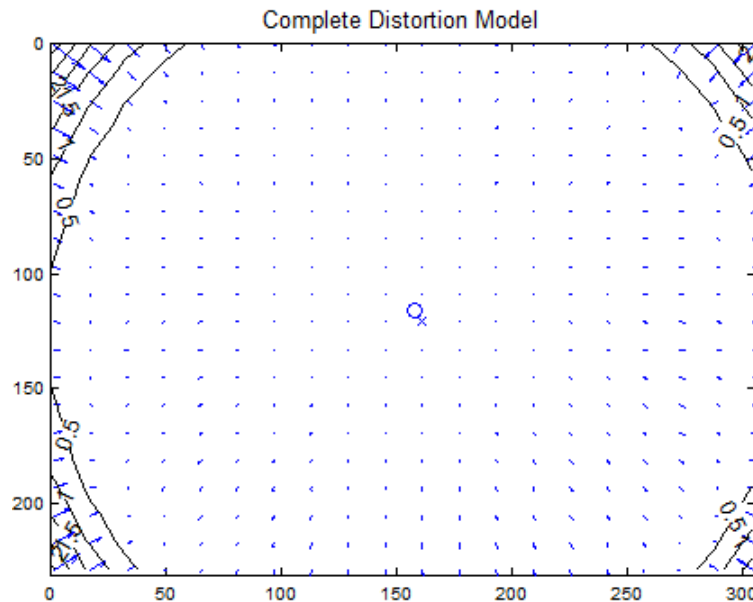
Ακολουθούν τα διαγράμματα για τα δύο είδη παραμόρφωσης που προκαλεί ο φακός (ακτινική και tangential), καθώς και της συνολικής παραμόρφωσης που προκύπτει από αυτά.



Εικόνα 110 - Η ακτινική παραμόρφωση του φακού



Εικόνα 111 - Η tangential παραμόρφωση του φακού



Εικόνα 112 - Η συνολική παραμόρφωση του φακού

Η αφαίρεση της επίδρασης του θορύβου με αναλυτική μέθοδο είναι πολύ κοστοβόρα, όπως φαίνεται από τις εξισώσεις (1.75)-(1.82) στο κεφάλαιο περί μαθηματικού υποβάθρου.

Έτσι, σε real time εφαρμογές, ειδικά σε κινητά τηλέφωνα, η διόρθωση των παραμορφώσεων του φακού¹⁰⁹ γίνεται με χρήση lookup tables [79] (με βάση τις εξισώσεις που αναφέρθηκαν).

Επειδή, όπως φαίνεται στην εικόνα 112, η παραμόρφωση που προκαλεί ο φακός μπορεί να θεωρηθεί αμελητέα, αποφασίστηκε η μη μοντελοποίηση των παραμορφώσεων του φακού στο χρησιμοποιούμενο μοντέλο κάμερας.

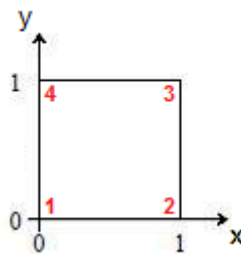
¹⁰⁹ Η διόρθωση δε γίνεται για ολόκληρη την εικόνα, αλλά **μόνο** για τις συντεταγμένες των γωνιών των markers που ανιχνεύθηκαν [27] (εκτός κι αν η διόρθωση γίνεται μέσω του hardware)

Pose Estimation

Χρησιμοποιήθηκε η μέθοδος Robust Pose Estimation from a Planar Target¹¹⁰ [97], με τον πρότυπο κώδικα Matlab που παρατίθεται στη σελίδα ενός εκ των δημιουργών [98], μιας και είναι μία από τις καλύτερες μεθόδους για tracking ενός marker, με πιο αξιόπιστα αποτελέσματα¹¹¹ σε σχέση με τις υπόλοιπους (πχ POSIT)

Στην παραπάνω υλοποίηση δίνουμε σαν όρισμα εισόδου τις συντεταγμένες των 4 άκρων κάθε marker και χρησιμοποιούμε σαν μοντέλο ένα μοναδιαίο τετράγωνο. Επειδή τα σημεία βρίσκονται στο ίδιο επίπεδο, έχουμε $z=0$ για όλα τα σημεία του μοντέλου.

	Σημείο 1	Σημείο 2	Σημείο 3	Σημείο 4
x	0	1	1	0
y	0	0	1	1
z	0	0	0	0



Εικόνα 113 - Το μοντέλο που χρησιμοποιείται για την pose estimation με τον RPP

Η παραπάνω σειρά έχει μεγάλη σημασία, ώστε το 3D σύστημα συντεταγμένων που θα προκύψει για το marker να έχει την αρχή του πάντα σε συγκεκριμένο orientation bit.

Σαν δείγμα χρησιμοποιούμε τα markers της εικόνας 114, τρέχοντας τον αλγόριθμο για κάθε marker χωριστά. Σαν είσοδο δίνουμε τις συντεταγμένες των άκρων κάθε marker, αφού προηγουμένως αφαιρέσουμε την επίδραση του φακού σε αυτές¹¹² (μέσω των εσωτερικών παραμέτρων)

$$(1.82) \Rightarrow \begin{bmatrix} x_{\text{είσοδος}}(1) \\ y_{\text{είσοδος}}(2) \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} x_{\text{μέτρησης}} \\ y_{\text{μέτρησης}} \\ 1 \end{bmatrix} \quad (1.83)$$

¹¹⁰ Αναφέρεται και ως "RPP"

¹¹¹ Οι Schweighofer και Pinz απέδειξαν ότι κατά την pose estimation παρατηρούνται (έως και) 2 τοπικά ελάχιστα της χρησιμοποιούμενης συνάρτησης σφάλματος (error function).

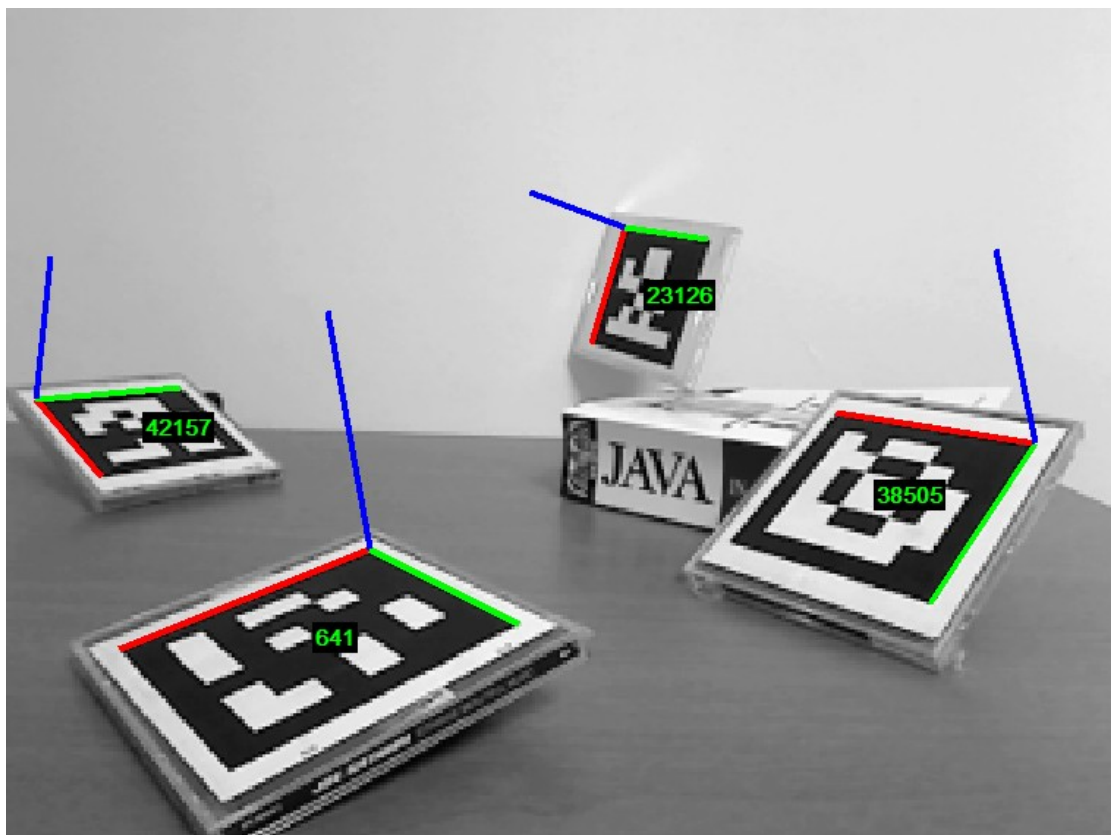
Το γεγονός ότι αυτό παραβλέπεται από τους υπόλοιπους (εκτός από τους Oberkampff et al στον αλγόριθμο POSIT [112]) οφείλεται για την παρατηρούμενη διακύμανση (pose jumps) των αποτελεσμάτων των υπόλοιπων αλγορίθμων, που γίνεται αντιληπτό στο χρήστη ως jittering.

Ο RPP λαμβάνει υπόψη και τα 2 τοπικά ελάχιστα, παρουσιάζοντας έτσι μεγαλύτερη ανοσία στο jittering, ενώ αποδίδει ελαφρώς καλύτερα από τον POSIT.

¹¹² Εάν στο μοντέλο της κάμερας λαμβάνεται υπόψη η radial και tangential distortion, πρέπει να αφαιρεθούν και αυτές οι παραμορφώσεις



Εικόνα 114 - Τα σημεία εισόδου στον αλγόριθμο RPP για την pose estimation



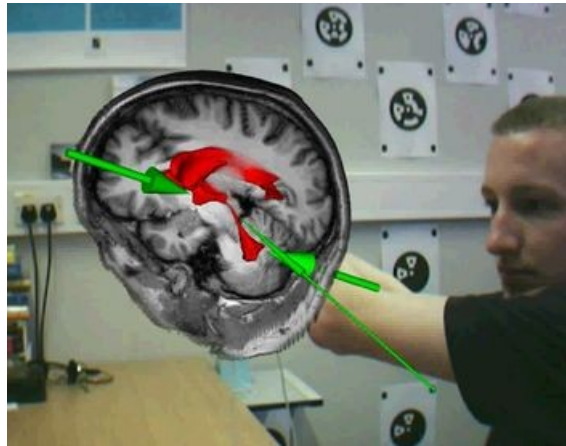
Εικόνα 115 - Το αποτέλεσμα μετά την pose estimation

Ο αλγόριθμος επιστρέφει τους κατάλληλους πίνακες rotation (R) και translation (t), τους οποίους μπορούμε να χρησιμοποιήσουμε για επικάλυψη οπουδήποτε 3D μοντέλου.

Ενδεικτικά, στην εικόνα 115 φαίνεται η επικάλυψη των αξόνων ενός 3D συστήματος συντεταγμένων, ενώ η αρχή του συστήματος βρίσκεται πάντα στο ίδιο σημείο σε σχέση με τα orientation bits¹¹³.

Στο εξεταζόμενο σενάριο, εάν υπήρχε ένα 3D μοντέλο κάποιου ανθρώπινου όργανου, θα μπορούσαμε να είχαμε την παρουσίαση των αποτελεσμάτων των εξετάσεων ενός ασθενούς πάνω σε αυτό, με τη μορφή γραφικών annotations. Έτσι ο γιατρός θα μπορούσε να έχει ένα εργαλείο επαυξημένης πραγματικότητας για καλύτερη επόπτευση των αποτελεσμάτων των εξετάσεων.

Κάτι τέτοιο θα έμοιαζε αρκετά με την εικόνα που ακολουθεί



Εικόνα 116 - Παράδειγμα επόπτευσης ιατρικών εξετάσεων με AR μέθοδο

¹¹³ Τα γωνιακά bits κάθε marker είναι τα orientation bits. Όπως φαίνεται έχουμε 2 μαύρα και 2 άσπρα bits, τα οποία βοηθούν τόσο στη σωστή δειγματοληψία του κάθε marker, όσο και στη σωστή επικάλυψη γραφικών.

7. Συμπεράσματα

Γενικά

Στην παρούσα εργασία αναπτύχθηκε ένας ευέλικτος αλγόριθμος ανίχνευσης markers σε εικόνα¹¹⁴ για χρήση σε εφαρμογές επαυξημένης πραγματικότητας.

Ο αλγόριθμος αυτός διαβάζει ένα frame από την κάμερα και εάν εντοπίσει σε αυτό κάποια markers, επιστρέφει σαν έξοδο τις συντεταγμένες των άκρων τους. Με βάση αυτά τα άκρα μπορούμε να διαβάσουμε με ασφάλεια το εσωτερικό του κάθε marker, βρίσκοντας έτσι τη μοναδική¹¹⁵ ταυτότητά του (ID). Αυτή μπορεί να χρησιμοποιηθεί για την πρόσβαση σε μία βάση δεδομένων με ψηφιακό περιεχόμενο κάθε είδους, καθιστώντας έτσι δυνατή τη δημιουργία μιας εφαρμογής στα πρότυπα του internet of things και της επαυξημένης πραγματικότητας.

Εάν επιπλέον οι συντεταγμένες των άκρων εισαχθούν σε κάποιον αλγόριθμο pose estimation, μπορούν να βρεθούν οι πίνακες rotation (R) και translation (t) που μαρτυρούν τη θέση της κάμερας (γωνία και απόσταση) ως προς το κάθε marker. Δίνεται έτσι η δυνατότητα επικάλυψης 3D μοντέλων σε αυτά, δημιουργώντας μία ρεαλιστική οπτική μίξη του εικονικού με τον πραγματικό κόσμο.

Τεχνικά χαρακτηριστικά

Ο αλγόριθμος χαρακτηρίζεται ως ευέλικτος, γιατί αποτελείται από αυτόνομες και άμεσα συνεργαζόμενες αλγοριθμικές ενότητες, με σαφώς διακριτό ρόλο. Έτσι, θα μπορούσαν να γίνουν αλλαγές σε κάποια ενότητα ή ακόμα και να γίνει πλήρης αντικατάστασή της, χωρίς να επηρεάζεται ο συνολικός αλγόριθμος, απ' τη στιγμή που τηρείται το "πρωτόκολλο" επικοινωνίας μεταξύ των ενότητων¹¹⁶.

Σε αντίθεση με τους περισσότερους αλγορίθμους, ο αλγόριθμος που παρουσιάζεται δε βασίζεται σε blob detection, αλλά σε edge detection. Δίνεται έτσι η δυνατότητα ανίχνευσης των markers ακόμα και υπό μη ιδανικές φωτιστικές συνθήκες ή υπό μερική επικάλυψη αυτών. Δυστυχώς όμως, αυτά τα θεμιτά πλεονεκτήματα έρχονται με το κόστος του επιπλέον υπολογιστικού φορτίου.

Επίσης ο αλγόριθμος κάνει χρήση διάφορων κατωφλίων σε κάθε ενότητα, που θα μπορούσαν να ρυθμιστούν κατάλληλα ανάλογα με την εφαρμογή και τις επιθυμίες του χρήστη. Έτσι ο χρήστης μπορεί προαιρετικά να κάνει ρύθμιση του αλγορίθμου για τις ειδικές ανάγκες κάποιας εφαρμογής, καθώς και να βρει την κατάλληλη τομή ανάμεσα στην ταχύτητα και στην "ευαισθησία" του αλγορίθμου (βλ παρακάτω).

Τέλος, ο αλγόριθμος μπορεί να εφαρμοστεί με οποιοδήποτε εξοπλισμό και για οποιαδήποτε ανάλυση κάμερας, με μικρές ίσως αλλαγές ανάλογα με την περίπτωση.

¹¹⁴ Σε μία στατική εικόνα και όχι σε video

¹¹⁵ Σε σχέση με τα υπόλοιπα markers

¹¹⁶ Με βάση τα δεδομένα εισόδου και εξόδου κάθε ενότητας

Τεχνικές παρατηρήσεις

Ο αλγόριθμος δεν εγγυάται κάποια ελάχιστη ταχύτητα εκτέλεσης, αλλά απεναντίας αυτή εξαρτάται από τα διάφορους παράγοντες:

- *Ρύθμιση κατωφλίων*: ανάλογα με τη ρύθμιση των διάφορων κατωφλίων (και ειδικότερα του κατωφλίου έντασης φωτεινότητας κατά την edge detection) παρατηρείται ένα trade off ανάμεσα στην ταχύτητα και στην “αποτελεσματικότητα ανίχνευσης” του αλγορίθμου.
 - Χαμηλή τιμή κατωφλίων επιφέρει πολλή περιττή πληροφορία¹¹⁷, πράγμα που σημαίνει πτώση της ταχύτητας εκτέλεσης λόγω του πρόσθετου φόρτου εργασίας. Παράλληλα έχουμε όμως αύξηση της αποτελεσματικότητας ανίχνευσης, καθώς ανιχνεύονται ακόμα και οι edges των markers που παρουσιάζονται “αδύναμες” λόγω ανακλάσεων του φωτός ή λόγω blurring.
 - Υψηλή τιμή κατωφλίων, επιφέρει τα αντίθετα αποτελέσματα: κατακράτηση μόνο της σημαντικότερης πληροφορίας¹¹⁸ και γρήγορη εκτέλεση. Πέφτει όμως η αποτελεσματικότητα ανίχνευσης των markers, καθώς κάποια από αυτά μπορεί να έχουν κάποια αδύναμη edge (για τους παραπάνω λόγους).
- *Σκηνή φωτογράφισης*: μία απλή σκηνή¹¹⁹ οδηγεί σε γρήγορη εκτέλεση, μιας και ανιχνεύονται σχεδόν αποκλειστικά μόνο οι ωφέλιμες πληροφορίες. Μία πολύπλοκη σκηνή¹²⁰ όμως “δίνει” μεγάλο όγκο δεδομένων προς επεξεργασία, ρίχνοντας την ταχύτητα του αλγορίθμου. Μάλιστα εάν υπάρχει κάποιο bug στον αλγόριθμο είναι πιο πιθανό να εμφανιστεί σε μία πολύπλοκη σκηνή, για προφανείς λόγους.
- *Η ανάλυση της κάμερας*: η ανάλυση παίζει και αυτή σημαντικό ρόλο, για τον προφανή λόγο της έκτασης της εικόνας που δίνεται προς επεξεργασία. Η μεγάλη ανάλυση ρίχνει κατακόρυφα την ταχύτητα εκτέλεσης, ενώ μπορεί να ρίξει και την αποτελεσματικότητα ανίχνευσης, όπως θα δούμε παρακάτω. Βέβαια η αρνητική επίδραση της ανάλυσης μπορεί να μετριαστεί κατάλληλα¹²¹, κάνοντας επεξεργασία μόνο σε επιλεγμένες περιοχές της. Το κόστος της μνήμης που καταναλώνεται όμως παραμένει.
- *Πλήθος markers*: Τέλος, έχει επίδραση και ο αριθμός των markers βρίσκεται σε κάθε frame. Όσο λιγότερα markers υπάρχουν σε μία σκηνή τόσο γρηγορότερη είναι η συνολική εκτέλεση και το αντίστροφο, για προφανείς λόγους.

¹¹⁷ Δηλαδή πολλές edges, οι περισσότερες εκ των οποίων δεν έχουν καμία σχέση με τα markers

¹¹⁸ “Ισχυρές” edges

¹¹⁹ Πχ μερικά markers κολλημένα σε έναν τοίχο. Οι γραμμές που ανιχνεύονται είναι λίγες και ανήκουν είτε στα markers είτε στη γεωμετρία του χώρου

¹²⁰ Πχ ένα marker τοποθετημένο σε μία βιβλιοθήκη γεμάτη βιβλία. Ο αλγόριθμος θα ανιχνεύσει πολλά γραμμικά τμήματα λόγω των βιβλίων και του επίπλου (ειδικά εάν χρησιμοποιείται και χαμηλό κατώφλι φωτεινότητας), κάτι που θα οδηγήσει σε μεγάλο υπολογιστικό φορτίο και αργή εκτέλεση.

¹²¹ με τη ρύθμιση της απόστασης των scanlines και το βαθμό του τεμαχισμού της εικόνας σε μη επικαλυπτόμενα κουτάκια για τοπική επεξεργασία

Πιθανά προβλήματα

Όπως συμβαίνει με κάθε αλγόριθμο, είναι δεδομένη η ύπαρξη ορισμένων περιπτώσεων στις οποίες ο προτεινόμενος αλγόριθμος είτε αποτυγχάνει είτε υπολειτουργεί. Παρακάτω ακολουθεί μία περιγραφή αυτών, καθώς και των πιθανών μέτρων που μπορούν να ληφθούν για να αποφευχθούν αυτές οι περιπτώσεις.

Αρχικά, είναι φυσικό να αποτυγχάνει η ανίχνευση ενός marker όταν ανακλάται σε αυτό πολύ έντονο τεχνητό φως¹²². Εξαιτίας της έντονης ανάκλασης η φωτεινότητα των pixels ενός marker μπορεί να αλλάζει σε τέτοιο βαθμό, που ακόμα και η μέθοδος edge detection να αποτυγχάνει στην εύρεση των πλευρών του marker. Προκειμένου να αποφευχθεί αυτό το γεγονός, μπορεί να ληφθεί πρόνοια κατά τη δημιουργία των marker, χρησιμοποιώντας για παράδειγμα κάποιο ματ υλικό και όχι γυαλιστερό χαρτί (πχ φωτογραφικό χαρτί), ενώ θα πρέπει να αποφεύγεται η χρήση των markers πολύ κοντά σε δυνατές πηγές τεχνητού φωτισμού (πχ πολύ κοντά σε μία δυνατή λάμπα φθορισμού).

Ο εντοπισμός αναμένεται να αποτυγχάνει και για τα markers των οποίων η χάρτινη επιφάνεια έχει παραμορφωθεί (όχι απαραίτητα πολύ), για παράδειγμα λόγω υγρασίας ή λόγω τσαλακώματος. Ο αλγόριθμος προϋποθέτει την ύπαρξη επίπεδων markers, ενώ είναι ανεκτές μόνο μικρές παραμορφώσεις της επιφάνειάς τους. Έτσι, θα πρέπει να χρησιμοποιείται σχετικά χοντρό χαρτί κατά την εκτύπωση και εάν είναι δυνατόν να προσκολλούνται τα markers πάνω σε κάποια φέρον αντικείμενο με σκληρή, επίπεδη επιφάνεια (πχ πάνω σε κάποια θήκη CD).

Επίσης, είναι δύσκολη η ανίχνευση των markers που βρίσκονται υπό πολύ μεγάλη κλίση¹²³ σε σχέση με την κάμερα, κάτι που είναι φυσικό, μιας και ούτε το ανθρώπινο μάτι μπορεί να “διαβάσει” το marker σε αυτή την περίπτωση. Μάλιστα, ακόμα κι αν το περίγραμμα του marker ανιχνευθεί κανονικά, η μεγάλη “προοπτική παραμόρφωση” οδηγεί σε αναξιόπιστη ανάγνωση/δειγματοληψία του εσωτερικού του marker και εσφαλμένη αποκωδικοποίηση του ID του. Τα ίδια φαινόμενα παρουσιάζονται και στην ανίχνευση των markers που βρίσκονται σε μεγάλη απόσταση από την κάμερα. Ένα απλό κατώφλι κατώτατου μήκους πλευράς μπορεί να είναι αρκετό για να αποφευχθούν αυτές τις δύο περιπτώσεις.

Τέλος, πρόβλημα μπορεί να ανακύψει και λόγω της ανάλυσης της κάμερας. Σε αντίθεση με ό,τι θα περίμενε κανείς, η αυξανόμενη ανάλυση από ένα σημείο και πέρα δημιουργεί προβλήματα, κυρίως κατά την επιμήκυνση των γραμμών¹²⁴. Χρησιμοποιώντας πολύ υψηλή ανάλυση είναι πιο πιθανό να σταματήσει η επιμήκυνση μιας γραμμής πριν τον επιθυμητό “στόχο”. Έτσι, πρέπει να χρησιμοποιείται μία σχετικά χαμηλή ανάλυση που να καλύπτει τις ανάγκες της εκάστοτε εφαρμογής, χωρίς να καταφεύγουμε στις υψηλότερες δυνατές αναλύσεις.

¹²² Όταν το ανακλώμενο φως δεν είναι υπερβολικά δυνατό η ανίχνευση αναμένεται να γίνεται κανονικά

¹²³ Εάν η κλίση δεν είναι υπερβολικά μεγάλη η ανίχνευση αναμένεται να γίνεται κανονικά

¹²⁴ με υψηλή ανάλυση είναι “αναγνώσιμες” πιο εύκολα οι όποιες παραμορφώσεις της (θεωρητικά επίπεδης) επιφάνειας των markers

8. Μελλοντικές Επεκτάσεις

Η ανάπτυξη του αλγορίθμου που παρουσιάζεται στην παρούσα εργασία έγινε με τη χρήση του MATLAB, ενός περιβάλλοντος ιδανικού για την ανάπτυξη και τη δοκιμή νέων αλγορίθμων. Οι όποιες ευκολίες όμως προσφέρει, έρχονται με το trade off της αργής εκτέλεσης.

Έτσι, η εκτέλεση του αλγορίθμου γίνεται στην παρούσα φάση για ένα μόνο frame από την κάμερα, ενώ η όποια επαναλαμβανόμενη εκτέλεση μέσα σε κάποιο βρόχο απέχει πολύ¹²⁵ από το να χαρακτηριστεί real time. Προκειμένου να είναι δυνατή η εκτέλεση σε πραγματικό χρόνο είναι επιβεβλημένη η “μεταφορά” του αλγορίθμου σε C/C++¹²⁶, και η χρήση OpenGL/directX για επικάλυψη γραφικών μέσω επιτάχυνσης από το κατάλληλο hardware.

Πέρα όμως από την επιτάχυνση των 3D γραφικών, μπορεί ίσως να επιταχυνθεί και η συνολική διαδικασία, με καλύτερη εκμετάλλευση του hardware (πέρα από τη CPU), όπως κυρίως της GPU (πχ openCL [99], CUDA [100]).

Επίσης, ειδικά στην περίπτωση υλοποίησης του αλγορίθμου για χρήση σε κινητά τηλέφωνα¹²⁷ επιβάλλεται η χρήση lookup tables όπου είναι δυνατόν για επιτάχυνση, καθώς και η χρήση fixed point υπολογισμών, η οικονομικότερη διαχείριση της μνήμης και η κατάλληλη ρύθμιση των διαφόρων παραμέτρων του αλγορίθμου (ανάλυση κάμερας, ρύθμιση κατωφλίων ώστε να αποφεύγεται η πολύ έντονη επεξεργασία κτλ).

Οι όποιες βελτιώσεις όμως, δεν αφορούν μόνο τις τεχνικές υλοποίησης του αλγορίθμου, αλλά και τη λογική του. Η επεξεργασία στην παρούσα φάση γίνεται σε κάθε frame ξεχωριστά, αγνοώντας τη σχέση που μπορεί να έχουν τα διαδοχικά frames της κάμερας. Έτσι όταν γίνεται ανίχνευση των markers σε κάποιο frame, είναι δυνατή η “εξυπνότερη” επεξεργασία στα επόμενα frames για κάποιο (περιορισμένο) διάστημα (πχ για τα επόμενα 10 frames). Θα μπορούσε δηλαδή να γίνεται επεξεργασία μόνο στην περιοχή που έχει ήδη ανιχνευθεί κάποιο marker και στην άμεση γειτονιά της, αναμένοντας μία σχετική μετατόπιση [95]. Η κατεύθυνση της όποιας μετατόπισης μπορεί να “προβλεφθεί” σε κάποιο βαθμό με χρήση κάποιας από τις υπάρχουσες τακτικές. Είναι λοιπόν δυνατό ο αλγόριθμος να βελτιστοποιηθεί με τεχνικές ήδη γνωστές και διαδεδομένες, όπως το Group of Pictures ή τα Motion Vectors [101] [102] που χρησιμοποιούνται στα διάφορα πρότυπα video.

Τέλος, θα μπορούσε το marker tracking να συνδυαστεί με κάποια μορφή natural feature tracking έτσι ώστε το tracking να μπορεί να συνεχιστεί με αρκετή αξιοπιστία για κάποιο διάστημα, ακόμα και όταν το marker βγει από το οπτικό πεδίο της κάμερας [83] (πχ συνδυασμός marker tracking και pixel flow ή FAST corner detection)

¹²⁵ Αν και η εκτέλεση ίσως μπορεί να επιταχυνθεί σημαντικά με μεταφορά του κώδικα MATLAB από την CPU στην GPU με κάποιο εργαλείο όπως: www.accelereyes.com ή www.gp-you.org

¹²⁶ Με παράλληλη βελτιστοποίηση του κώδικα, μιας και ο κώδικας του MATLAB που αναπτύχθηκε περιέχει πολλά κομμάτια που στοχεύουν απλά στην αποσφαλμάτωση και στην εκτύπωση-παρουσίαση αποτελεσμάτων σε κάθε στάδιο.

¹²⁷ Επειδή ο τομέας αυτός παρουσιάζει έντονο ενδιαφέρον, ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει για περισσότερες πληροφορίες στις εργασίες [41] [85] [27] [83]

Βιβλιογραφία

- [1] H. and Ullmer, B. Ishii, "Tangible Bits: Towards Seamless Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1997, pp. 234-241.
- [2] M. Weiser, "The Computer for the 21st Century," *Scientific American*, pp. 94-104, September 1991.
- [3] Christian Sandor Bruce H. Thomas, "What Wearable Augmented Reality Can Do for You," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 8-11, April-June 2009.
- [4] Ronald T. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, pp. 355-385, August 1997.
- [5] J. Pilet, P. Fua, and V. Lepetit C. Scherrer, "The Haunted Book," in *Proceedings of International Symposium on Mixed and Augmented Reality*, 2008.
- [6] J.D. Macintyre, B. Bolter, "Is It Live or Is It AR?," *Spectrum, IEEE*, vol. 44, no. 8, pp. 30-35, August 2007.
- [7] Joseph Rozier, "Hear&There: An Augmented Reality System of Linked Audio," MIT, 1999.
- [8] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Transactions on Information Systems*, December 1994.
- [9] R. Bailiot, Y. Behringer, R. Feiner, S. Julier, S. MacIntyre, B. Azuma, "Recent Advances in Augmented Reality," *Computer Graphics and Applications, IEEE*, vol. 21, no. 6, pp. 34-47, Nov/Dec 2001.
- [10] Ivan E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 757-764.
- [11] Ivan E. Sutherland, "The Ultimate Display," in *Proceedings of the IFIP Congress*, 1965, pp. 506-508.
- [12] T. and Mizell, D. Caudell, "Augmented reality: an application of heads-up display technology to manual manufacturing processes," in *Proc. Hawaii International Conf. on Systems Science*, 1992, pp. 659-669.
- [13] H. B. L. Duh, M. Billingham Feng Zhou, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," in *Mixed and Augmented Reality, 2008. ISMAR 2008*, 2008, pp. 193-202.
- [14] Erika Jonietz, "TR10: Augmented Reality," *Technology Review*, March/April 2007.

- [15] F. P. Brooks, "The computer scientist as toolsmith II," *Communications of the ACM*, vol. 39, no. 3, pp. 61-68, March 1996.
- [16] Wikipedia - Augmented Reality. [Online].
http://en.wikipedia.org/wiki/Augmented_reality
- [17] Katharina Pentenrieder and Gudrun Klinker. (2008) Anwendungen der Augmented Reality. [Online]. <http://campar.in.tum.de/Chair/TeachingSs08ArProseminar>
- [18] T. Obst, R. Burgkart, R. Riener, N. Navab T. Sielhorst, "An Augmented Reality Delivery Simulator for Medical Training ," in *International Workshop on Augmented Environments for Medical Imaging - MICCAI Satellite Workshop*, 2004.
- [19] Navigations- und AR - Visualisierungssystem für den Einsatz bei minimalinvasiven, unfallchirurgischen Operationen (NARVIS). [Online]. <http://www.narvis.org/>
- [20] Dieter Schmalstieg et al. Vidente - Handheld Augmented Reality for Technical Infrastructure. [Online]. <http://www.vidente.at/>
- [21] Blair MacIntyre, Dorée Seligmann Steven Feiner. KARMA - Knowledge-based Augmented Reality for Maintenance Assistance. [Online].
<http://graphics.cs.columbia.edu/projects/karma/karma.html>
- [22] The ARQuake Project. [Online]. <http://wearables.unisa.edu.au/arquake/>
- [23] Eye of Judgment. [Online]. <http://www.us.playstation.com/EyeofJudgment/>
- [24] Alessandro Mulloni. (2007) Cow VS Aliens. [Online]. http://studierstube.icg.tu-graz.ac.at/handheld_ar/cows_vs.aliens.php
- [25] Steven Dow et al. Four Angry Men. [Online].
<http://www.cc.gatech.edu/acl/projects/4am.html>
- [26] The MIT Scheller Teacher Education Program. [Online].
<http://education.mit.edu/drupal/ar>
- [27] Wagner Daniel Schmalstieg Dieter, "Experiences with Handheld Augmented Reality," in *The Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, 2007, pp. 3-15.
- [28] V. Vlahakis et al., "Archeoguide: an augmented reality guide for archaeological sites," *Computer Graphics and Applications, IEEE*, vol. 22, no. 5, pp. 52-60, Sept.-Oct. 2002.
- [29] Jay David Bolter, and Maribeth Gandy Blair MacIntyre, "Presence and the Aura of Meaningful Places," in *7th Annual International Workshop on Presence (PRESENCE 2004)*, 2004, pp. 13-15.

- [30] T.S. Perry, "All in the Game," *Spectrum, IEEE*, vol. 40, no. 11, pp. 31-35, Nov. 2003.
- [31] Schall Gerhard, Wagner Daniel, Barakonyi István, Gerhard Reitmayr, Newman Joseph, Florian Ledermann Schmalstieg Dieter, "Managing Complex Augmented Reality Models," *IEEE Computer Graphics and Applications*, vol. 272, no. 1716, pp. 32-41, July/August 2007.
- [32] Wagner Daniel, Barakonyi István, Schmalstieg Dieter Mulloni Alessandro, "Indoor Positioning and Navigation with Camera Phones," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 22-31, April-June 2009.
- [33] B. MacIntyre, T. Höllerer, and T. Webster S. Feiner, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," in *Proc. ISWC '97 (First IEEE Int. Symp. on Wearable Computers)*, 1997, pp. 208-217.
- [34] Harlan Hile et al., "Landmark-based pedestrian navigation from collections of geotagged photos," in *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia*, 2008, pp. 145-152.
- [35] Mark Fiala. ARTag. [Online]. <http://www.artag.net/>
- [36] P. Eisert, J. Rurainsky, and P. Fechteler, "Virtual mirror: Real-time tracking of shoes in augmented reality environments," in *IEEE International Conference on Image Processing, ICIP 2007*, 2007, pp. 16-19.
- [37] Schmalstieg Dieter Wagner Daniel, "History and Future of Tracking for Mobile Phone Augmented Reality," in *International Symposium on Ubiquitous Virtual Reality (ISUVR 2009)*, 2009.
- [38] J.R. Lewis, "In the eye of the beholder," *Spectrum, IEEE*, vol. 41, no. 5, pp. 24-28, May 2004.
- [39] Babak A. Parviz, "Augmented Reality in a Contact Lens," *Spectrum, IEEE*, September 2009.
- [40] P. Maes, L. Chang P. Mistry, "WUW - Wear Ur World - A Wearable Gestural Interface," in *CHI '09 extended abstracts on Human factors in computing systems*, 2009.
- [41] Schmalstieg Dieter Wagner Daniel, "Making Augmented Reality Practical on Mobile Phones, Part 1," *IEEE Computer Graphics and Applications*, vol. 29, no. 3, pp. 12-15, May/June 2009.
- [42] Daniel Wagner. (2009) History of Mobile Augmented Reality. [Online]. <https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/>
- [43] B. Wanstall, "HUD on the Head for Combat Pilots," in *Interavia*, 1989, pp. 334-338.

- [44] M. Billinghurst H. Kato, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, 1999, pp. 85-94.
- [45] MARA - The Mobile Augmented Reality Applications project. [Online]. <http://research.nokia.com/research/projects/mara/index.html>
- [46] Reitmayr Gerhard, Mulloni Alessandro, Drummond Tom, Schmalstieg Dieter Wagner Daniel, "Pose Tracking from Natural Features on Mobile Phones," in *International Symposium on Mixed and Augmented Reality (ISMAR 08)*, 2008.
- [47] Markus Tripp and Robert Rice. ARML Introduction - Augmented Reality (AR) Specification Based on KML with Extensions. [Online]. <http://www.openarm1.org/>
- [48] P. Mistry P. Maes. (2009) Unveiling the "Sixth Sense," game-changing wearable tech.
- [49] Wikipedia - Feature detection (computer vision). [Online]. http://en.wikipedia.org/wiki/Feature_detection_%28computer_vision%29
- [50] S. Carlsson, and A. Zisserman J. Clarke, "Detecting and tracking linear features efficiently," in *Proc. 7th British Machine Vision Conf. (BMVA)*, 1996, pp. 415-424.
- [51] J Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, November 1986.
- [52] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147-151.
- [53] E. and Drummond, T.W Rosten, "Machine learning for high-speed corner detection," in *The 9th European Conference on Computer Vision (ECCV 2006)*, 2006.
- [54] V. Chandrasekhar, N. Gelfand, Y. Xiong, W. C. Chen, T. Bismpiagiannis, R. Grzeszczuk, K. Pulli, and B. Girod G. Takacs, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proceeding of the 1st ACM international conference on Multimedia information retrieval (MIR '08)*, 2008, pp. 427-434.
- [55] David G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004, pp. 91-110.
- [56] David Lowe. Demo Software: SIFT Keypoint Detector. [Online]. <http://people.cs.ubc.ca/~lowe/keypoints/>
- [57] Andreas Ess, Tinne Tuytelaars, Luc Van Gool Herbert Bay, "SURF: Speeded Up Robust Features," in *Computer Vision and Image Understanding (CVIU)*, 2008, pp. 346-359.
- [58] SURF: Speeded Up Robust Features. [Online].

<http://www.vision.ee.ethz.ch/~surf/index.html>

- [59] kooaba. [Online]. <http://www.kooaba.com/>
- [60] P. Fua, V. Lepetit M. Özuysal, "Fast Keypoint Recognition in Ten Lines of Code," in *Conference on Computer Vision and Pattern Recognition*, 2007.
- [61] BazAR: A vision based fast detection library. [Online]. <http://cvlab.epfl.ch/software/bazar/>
- [62] K. Schmid, C. Mikolajczyk, "A Performance Evaluation of Local Descriptors," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, pp. 1615-1630.
- [63] S., Hellwich, O. Sarfraz, "Head Pose Estimation in Face Recognition across Pose Scenarios," in *Proceedings of VISAPP'08*, 2008.
- [64] Engin Tola, Vincent Lepetit, and Pascal Fua, "Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
- [65] Engin Tola. Local Descriptor for Dense Wide-Baseline Stereo Matching. [Online]. <http://cvlab.epfl.ch/~tola/daisy.html>
- [66] Wei-Chao Chen, Natasha Gelfand, Kari Pulli Duy-Nguyen Ta, "SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009.
- [67] H. Tan, K.T. Kato, "Pervasive 2D Barcodes for Camera Phone Applications," *Pervasive Computing, IEEE*, vol. 6, no. 4, pp. 76-85, Oct.-Dec. 2007.
- [68] Wikitude - World Browser. [Online]. http://www.wikitude.org/world_browser
- [69] Michael Rohs, "Visual Code Widgets for Marker-Based Interaction," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (IWSAWC'05)*, 2005.
- [70] J. Rekimoto, "Matrix: A Realtime Object Identification and Registration Method for Augmented Reality," in *Third Asian Pacific Computer and Human Interaction*, 1998.
- [71] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments (DARE '00)*, 2000, pp. 1-10.
- [72] S. Navab, N. Xiang Zhang Fronz, "Visual marker detection and decoding in AR systems: a comparative study," in *International Symposium on Mixed and Augmented Reality*

(ISMAR 2002), 2002, pp. 97-106.

- [73] Mark Fiala, "Comparing ARTag and ARToolkit Plus fiducial marker systems," in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, 2005.
- [74] ARToolKit. [Online]. <http://www.hitl.washington.edu/artoolkit/>
- [75] M. Fiala, "ARTag Revision 1: A Fiducial Marker System Using Digital Techniques," National Research Council Canada, November 2004.
- [76] C.B. Fan Xiao Middlin, P. Owen, "What is the best fiducial?," in *The First IEEE International Augmented Reality Toolkit Workshop*, 2002, pp. 98-105.
- [77] D. Schmalstieg G. Reitmayr, "Location based applications for mobile augmented reality," Vienna University of Technology, Technical Report 2003.
- [78] Schmalstieg Dieter Wagner Daniel, "First Steps Towards Handheld Augmented Reality," in *Proceedings of the 7th International Conference on Wearable Computers (ISWC 2003)*, 2003.
- [79] Schmalstieg Dieter Wagner Daniel, "ARToolKitPlus for Pose Tracking on Mobile Devices," in *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, 2007.
- [80] Jean-Yves Bouguet. (2008 , June) Camera Calibration Toolbox for Matlab. [Online]. http://www.vision.caltech.edu/bouguetj/calib_doc/
- [81] Daniel Wagner. Studierstube Tracker. [Online]. http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbtracker.php
- [82] Daniel Wagner. Studierstube ES. [Online]. http://studierstube.icg.tu-graz.ac.at/handheld_ar/stbes.php
- [83] Wagner Daniel, Tobias Langlotz, and Dieter Schmalstieg, "Robust and Unobtrusive Marker Tracking on Mobile Phones," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 08)*, 2008.
- [84] Huang J. Costanza E., "Designable Visual Markers," in *Proceedings ACM CHI2009*, 2009.
- [85] Schmalstieg Dieter Wagner Daniel, "Making Augmented Reality Practical on Mobile Phones, Part 2," *IEEE Computer Graphics and Applications*, vol. 29, no. 4, pp. 6-9, July/August 2009.
- [86] Wikipedia - Image gradient. [Online]. http://en.wikipedia.org/wiki/Image_gradient

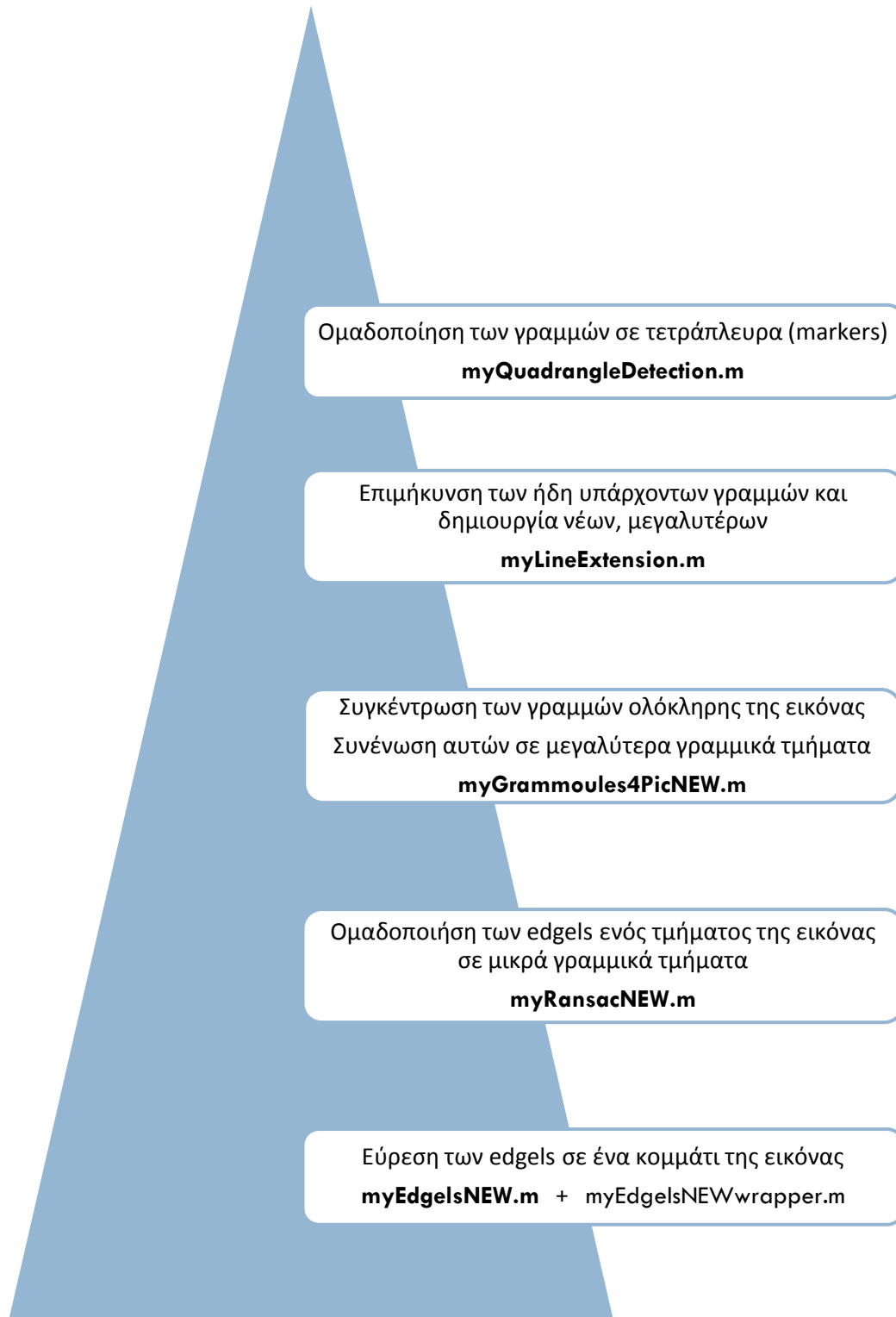
- [87] Paul Heckbert, "Projective Mappings for Image Warping," U.C. Berkley, Master Thesis 1995.
- [88] Wikipedia - Pinhole camera model. [Online].
http://en.wikipedia.org/wiki/Pinhole_camera_model
- [89] How to Make and Use a Pinhole Camera. [Online].
<http://www.kodak.com/global/en/consumer/education/lessonPlans/pinholeCamera/pinholeCanBox.shtml>
- [90] Stephen Cawood and Mark Fiala, *Augmented Reality: A Practical Guide.*: pragprog, 2008.
- [91] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed.: Cambridge University Press, 2004.
- [92] Janne Heikkila and Olli Silven, "A Four-step Camera Calibration Procedure with Implicit Image Correction," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, 1997.
- [93] Jean-Yves Bouguet. Description of the calibration parameters. [Online].
http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html
- [94] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration," Microsoft Research, Technical Report 1998.
- [95] Martin Hirzer, "Marker Detection for Augmented Reality Applications," TU-Graz, 2008.
- [96] B.H. Shekar, P. Nagabhushan D.S.Guru, "A simple and robust line detection algorithm based on small eigenvalue analysis," *Pattern Recognition Letters*, vol. 25, no. 1, pp. 1-13, January 2004.
- [97] G. Pinz, A. Schweighofer, "Robust Pose Estimation from a Planar Target," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2024-2030, Dec 2006.
- [98] Gerald Schweighofer. Gerald Schweighofer. [Online].
<http://www.emt.tugraz.at/~vmg/schweighofer>
- [99] Wei-Chao Chen, Natasha Gelfand, Radek Grzeszczuk, Marius Tico, Ramakrishna Vedantham, Xianglin Wang, Yingen Xiong Kari Pulli, "Mobile Visual Computing," in *International Symposium on Ubiquitous Virtual Reality*, 2009.
- [100] Michael Garland Jonathan Cohen, "Solving Computational Problems with GPU Computing," *Computing in Science and Engineering*, vol. 11, no. 5, pp. 58-63, Sep/Oct

2009.

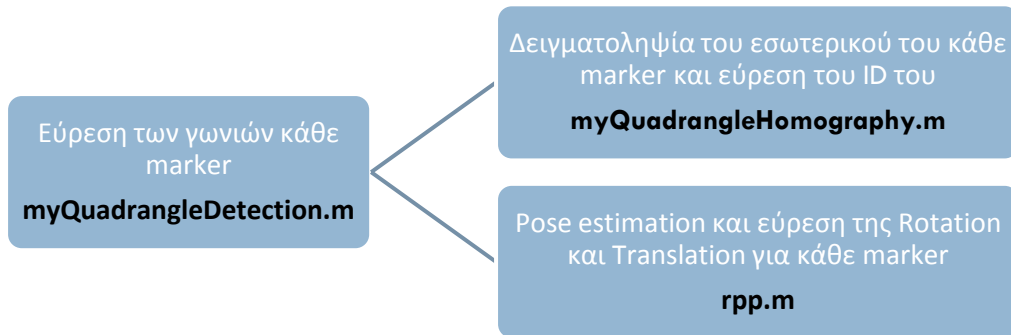
- [101] G. Takacs, V. Chandrasekhar, B. Girod, and R. Grzeszczuk, "Feature Tracking for Mobile Augmented Reality Using Video Coder Motion Vectors," in *Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.
- [102] Gabriel Takacs, Chih-Wei Louis Chen Vijay Chandrasekhar, "Feature Tracking with H.264 Motion Vectors for Mobile Augmented Reality," Stanford, Project Report 2007.
- [103] Wikipedia - Object hyperlinking. [Online].
http://en.wikipedia.org/wiki/Physical_world_hyperlinks
- [104] E., Brombach, B., Zeidler, T. and Bimber, O. Bruns, "Enabling Mobile Phones To Support Large-Scale Museum Guidance," *IEEE Multimedia*, vol. 14, no. 2, pp. 16-25, 2007.
- [105] Wikipedia - TinyURL. [Online]. <http://en.wikipedia.org/wiki/TinyURL>
- [106] Mark Fiala, "ARTag, An Improved Marker System Based on ARToolkit," National Research Council Canada, July 2004.
- [107] Daniel Wagner. (2006, June) ARToolkitplus. [Online]. http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php
- [108] Wikipedia - Vignetting. [Online]. <http://en.wikipedia.org/wiki/Vignetting>
- [109] Wikipedia - ARM architecture. [Online].
http://en.wikipedia.org/wiki/ARM_architecture
- [110] FLARToolKit. [Online]. <http://www.libspark.org/wiki/saqoosha/FLARToolKit/en>
- [111] G Feldman I Sobel, "A 3x3 isotropic gradient operator for image processing," Stanford, Presentation for Stanford Artificial Project 1968. [Online].
http://en.wikipedia.org/wiki/Sobel_operator
- [112] D. F. Dementhon, and L. S. Davis D. Oberkamp, "Iterative pose estimation using coplanar points," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '93)*, 1993, pp. 626-627.
- [113] Σ. Μ. Πανάς, *Ψηφιακή Επεξεργασία Σημάτων*. Θεσσαλονίκη: University Studio Press, 1987.

Παράρτημα - Επεξήγηση κώδικα

Παρακάτω παρουσιάζονται τα ονόματα των .m αρχείων του Matlab που αντιστοιχούν στην αλγοριθμική ιεραρχία που παρουσιάστηκε στις εικόνες 93 και 94



Εικόνα 117 - Τα ονόματα των αρχείων του MATLAB και η αλγοριθμική ιεραρχία



Εικόνα 118 - Τα ονόματα των αρχείων του MATLAB και το ανώτατο επίπεδο της αλγοριθμικής ιεραρχίας

Όπως φαίνεται, τον κεντρικό ρόλο τον παίζει η συνάρτηση **myQuadrangleDetection.m** που βρίσκεται στο ανώτατο επίπεδο της αλγοριθμικής ιεραρχίας.

Η όλη εκτέλεση ρυθμίζεται από αυτή, ενώ στην κορυφή του κώδικά της βρίσκεται ένα είδος “κέντρου ελέγχου”, όπου μπορούν να γίνουν όλες οι ρυθμίσεις για τις υποενότητες κάθε αλγοριθμικού επιπέδου.