

ΤΕΙΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

ΣΕ
ΡΡ
ΕΣ

2013

ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ
ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΙΝΗΣΗΣ ΣΕ
ΕΛΕΓΧΟΜΕΝΟ ΧΩΡΟ ΜΕ ΧΡΗΣΗ
WEBCAMERA ΚΑΙ ΟΡΕNCV

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Στεφανία Παπαρήγα

ΤΕΙΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ
ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΙΝΗΣΗΣ ΣΕ
ΕΛΕΓΧΟΜΕΝΟ ΧΩΡΟ ΜΕ ΧΡΗΣΗ
WEBCAMERA ΚΑΙ ΟΡΕNCV

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Στεφάνια Παπαρήγα, Α.Μ. 1838

ΣΕΡΡΕΣ, 2013

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

ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΙΝΗΣΗΣ
ΣΕ ΕΛΕΓΧΟΜΕΝΟ ΧΩΡΟ ΜΕ ΧΡΗΣΗ WEBCAMERA ΚΑΙ OPENCV

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Στεφανία Παπαρήγα, Α.Μ. 1838

Επιβλέπων: Δρ. Γεώργιος Παυλίδης
Επιστημονικός Συνεργάτης Τ.Ε.Ι. Σερρών

Τ.Ε.Ι. ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Στους γονείς μου

Περίληψη

Η καθημερινότητα ενός σύγχρονου ανθρώπου έχει αλλάξει ριζικά εξαιτίας της ραγδαίας ανάπτυξης της τεχνολογίας. Ο άνθρωπος, στις καθημερινές του δραστηριότητες, δρα υπό τον περιορισμό του χρόνου και του χώρου. Όσο βελτιώνεται η τεχνολογία αυτοί οι περιορισμοί εξαλείφονται. Έτσι μπορούμε να πραγματοποιήσουμε περισσότερες λειτουργίες και να καλύψουμε τις ανάγκες μας σε πολύ λιγότερο χρόνο καθώς και ο παράγοντας της απόστασης τείνει να γίνεται όλο και μικρότερο εμπόδιο. Μια από τις σημαντικότερες ανθρώπινες ανάγκες στη σύγχρονη αυτή εποχή είναι η αίσθηση της ασφάλειας. Η βασική αίσθηση της ασφάλειας ταυτίζεται με την αίσθηση της ασφάλειας του προσωπικού χώρου του κάθε ανθρώπου. Βασικό στοιχείο σε αυτό είναι ο έλεγχος του περιβάλλοντός του μέσω της παρακολούθησης του χώρου αυτού.

Στην επιτήρηση χώρων, η οπτική επαφή είναι ένα σημαντικό μέσο αξιολόγησης του χώρου και της κατάστασης που επικρατεί σε αυτόν. Όταν όμως δεν υπάρχει άνθρωπος παρατηρητής στο χώρο υπό επίβλεψη τότε αυτή η επαφή χάνεται. Το κενό αυτό μπορούν σήμερα να καλύψουν συστήματα που βασίζονται σε απλές κάμερες και υπολογιστές.

Αναγνωρίζοντας την ανάγκη του ανθρώπου για ασφάλεια και την ευκολία που παρέχεται πλέον από τους υπολογιστές για την επιτήρηση χώρων, στην παρούσα εργασία μελετήθηκαν τεχνικές επιτήρησης χώρου και αναπτύχθηκε εφαρμογή βασισμένη στη βιβλιοθήκη επεξεργασίας εικόνας OpenCV, με στόχο την παρακολούθηση ελεγχόμενου χώρου με χρήση υπολογιστή και μιας απλής διαδικτυακής κάμερας (webcamera).

Abstract

The daily life of modern humans has changed dramatically due to the rapid development of technology. Humans, in their daily activities are acting under the limitation of time and space. As technology improves, these restrictions are diminished or even eliminated. This has a significant impact on daily lives, as humans can now accomplish more functions and to meet their needs in less time. The distance factor tends to become a smaller and smaller obstacle. One of the most important human needs in this modern era is the sense of security. The basic sense of security coincides with the same sense of security of personal space of everyone. A key element in this is to control the environment through visual monitoring.

In surveillance, visual contact is an important tool in assessing a site and a situation. In cases where there is no human observer on site under supervision then this visual contact is lost. This gap can now cover systems based on simple cameras and computers.

Recognizing the human need for privacy and security and the convenience provided by computers on surveillance sites, in this thesis surveillance techniques have been studied and a video surveillance system has been developed and implemented based on the image processing library Open CV, to monitor a personal space using a computer and a simple web camera.

Πρόλογος

Η παρούσα εργασία έχει ως αντικείμενο την ανάπτυξη ενός συστήματος αναγνώρισης και παρακολούθησης κίνησης σε ελεγχόμενο χώρο με απλή webcam. Υλοποιήθηκε στο πλαίσιο της ανάθεσης πτυχιακής εργασίας από το τμήμα Μηχανικών Πληροφορικής της Σχολής Τεχνολογικών Εφαρμογών του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Κεντρικής Μακεδονίας στις Σέρρες.

Θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν σε αυτό το διάστημα εκπόνησης της πτυχιακής μου εργασίας.

Πάνω απ' όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ.Γεώργιο Παυλίδη για την άψογη συνεργασία και βοήθεια που μου προσέφερε κατά τη διάρκεια της συγγραφής της εργασίας.

Ταυτόχρονα, θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη συμπαράσταση που μου έδειξε σε όλη τη φοιτητική μου πορεία.

*Στεφανία Παπαρήγα
Σέρρες, 2013*

EΙΚΟΝΕΣ

Εικόνα 1. ISpylogo.....	18
Εικόνα 2. Το Λογισμικό ISpy	18
Εικόνα 3. Microsoft Visual Studio 2010 Express Edition	27
Εικόνα 4. Open CV Logo.....	1
Εικόνα 5. Συστατικά μέρη της OpenCV	29
Εικόνα 6. Διάγραμμα ροής λογισμικού.....	36
Εικόνα 7. Τυπικό παράδειγμα εσωτερικού χώρου.....	37
Εικόνα 8. Τυπικό παράδειγμα εξωτερικού χώρου	37
Εικόνα 9. Παράθυρο FG	39
Εικόνα 10. Παράθυρο Video Surveillance.....	39
Εικόνα 11. Παράθυρο FG όταν γίνεται αντιληπτή κίνηση	41
Εικόνα 12. Παράθυρο FG όταν εισέρχεται αντικείμενο στο πλάνο	43
Εικόνα 13. Παράθυρο Video Surveillance όταν εισέρχεται αντικείμενο στο πλάνο	43
Εικόνα 14. Φάκελος VideoSurveillance	44
Εικόνα 15. Φάκελοι με ημερομηνίες που περιέχονται στον φάκελο VideoSurveillance.....	45
Εικόνα 16. Εικόνες με τίτλο την ώρα που αποθηκεύθηκαν.....	45
Εικόνα 17. Νέα ειδοποίηση μέσω ηλεκτρονικού ταχυδρομείου.....	46
Εικόνα 18. Μήνυμα ηλεκτρονικού ταχυδρομείου από την εφαρμογή.....	46
Εικόνα 19. Παράθυρο VideoSurveillance σε πείραμα εσωτερικού χώρου	49
Εικόνα 20. Παράθυρο FG σε πείραμα εσωτερικού χώρου	49
Εικόνα 21. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα εσωτερικού χώρου	50
Εικόνα 22. Εικόνα που αποθηκεύθηκε σε πείραμα εσωτερικού χώρου ..	50
Εικόνα 23. Παράθυρο VideoSurveillance σε πείραμα εξωτερικού χώρου	51
Εικόνα 24. Παράθυρο FG σε πείραμα εξωτερικού χώρου.....	51
Εικόνα 25. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα εξωτερικού χώρου	52
Εικόνα 26. Εικόνα που αποθηκεύθηκε σε πείραμα εξωτερικού χώρου ...	52
Εικόνα 27. Παράθυρο VideoSurveillance σε πείραμα πολύ χαμηλού φωτισμού.....	53
Εικόνα 28. Παράθυρο FG σε πείραμα πολύ χαμηλού φωτισμού.....	53
Εικόνα 29. Παράθυρο VideoSurveillance σε πείραμα με απότομη μείωση φωτισμού.....	54
Εικόνα 30. Παράθυρο FG σε πείραμα με απότομη μείωση φωτισμού	54
Εικόνα 31. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με απότομη μείωση φωτισμού.....	55
Εικόνα 32. Αποθήκευση εικόνας σε πείραμα με απότομη μείωση φωτισμού.....	55

Εικόνα 33. Παράθυρο VideoSurveillance σε πείραμα με απότομη αύξηση φωτισμού.....	56
Εικόνα 34. Παράθυρο FG σε πείραμα με απότομη αύξηση φωτισμού....	56
Εικόνα 35. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με απότομη αύξηση φωτισμού.....	57
Εικόνα 36. Αποθήκευση εικόνας σε πείραμα με απότομη αύξηση φωτισμού.....	57
Εικόνα 37. Παράθυρο VideoSurveillance σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α).....	58
Εικόνα 38. Παράθυρο FG σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α).....	58
Εικόνα 39. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α).....	59
Εικόνα 40. Αποθήκευση εικόνας σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α).....	59
Εικόνα 41. Παράθυρο VideoSurveillance σε πείραμα με είσοδο σκιάς στο πλάνο– περίπτωση (β).....	60
Εικόνα 42. Παράθυρο FG σε πείραμα με είσοδο σκιάς στο πλάνο– περίπτωση (β).....	60
Εικόνα 43. Έναρξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (α).....	61
Εικόνα 44. Λήξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (α).....	61
Εικόνα 45. Έναρξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (β).....	61
Εικόνα 46. Λήξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (β).....	62
Εικόνα 47. Εικόνες που αποθηκεύθηκαν.....	62
Εικόνα 48. Μήνυμα ηλεκτρονικού ταχυδρομείου από το λογισμικό ενώ ο Η/Υ βρίσκεται σε αδράνεια.....	63
Εικόνα 49. Μήνυμα λάθους από το λογισμικό.....	67
Εικόνα 50. Ιδιότητες Include του cpp αρχείου.....	80
Εικόνα 51. Ιδιότητες Library του cpp αρχείου.....	80
Εικόνα 52. Ιδιότητες Source του cpp αρχείου.....	81
Εικόνα 53. Ιδιότητες AdditionalDependencies του cpp αρχείου.....	81
Εικόνα 54. Η ιστοσελίδα www.emailarchitect.net	82

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	15
1.1 Ασφάλεια και επιτήρηση.....	15
1.2 Τι είναι λογισμικό βιντεοεπιτήρησης	15
1.3 Η ιστορία της βιντεοεπιτήρησης	16
1.4 Τι είναι η ανίχνευση κίνησης;	17
1.5 Ελεύθερα λογισμικά βιντεοεπιτήρησης	18
1.6 Η εφαρμογής που αναπτύχθηκε.....	19
ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ	21
2.1 Κατηγορίες Αλγορίθμων Αναγνώρισης Κίνησης.....	21
2.1.1 Αφαίρεση του φόντου (Background subtraction).....	21
2.1.2. Χρονική διαφορά (Temporal differencing)	21
2.1.3. Οπτική ροή (Optical flow).....	22
2.2. Αλγόριθμος Gauss background/foreground detection.....	22
2.2.1. Περιγραφή Μοντέλου.....	22
2.2.2. Κατάτμηση κίνησης με χρήση στατιστικών μεθόδων	23
2.2.3. Ενημέρωση Παραμέτρων	23
2.2.4. Κώδικας του αλγόριθμου σε c++	25
2.3. Microsoft Visual Studio.....	27
Τα λειτουργικά συστήματα στα οποία μπορεί να εκτελεστεί το Microsoft Visual Studio 2010 Express είναι τα εξής:	27
2.3.3. Ελάχιστες Απαιτήσεις Συστήματος	28
2.4. Τι είναι βιβλιοθήκη.....	28
2.5. Η βιβλιοθήκη OpenCV	28
2.5.1. Δομικά Μέρη της OpenCV	29
2.5.2. Πλεονεκτήματα της OpenCV	30
2.5.3. Εφαρμογές της OpenCV.....	30
2.6. Η βιβλιοθήκη EASendMail.....	31
2.6.1 Βασικά χαρακτηριστικά βιβλιοθήκης EASendMail.....	31
2.7. Τι είναι SMTP	32
ΠΕΡΙΓΡΑΦΗ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΛΥΣΗΣ	33
3.1. Τεκμηρίωση συναρτήσεων που χρησιμοποιήθηκαν	33
3.1.1. CvCapture.....	33
3.1.2. struct IplImage.....	33
3.1.3. CvNamedWindow	33
3.1.4. CvDestroyWindow	34

3.1.5. CvShowImage.....	34
3.1.6. CvSaveImage.....	34
3.1.7. CvQueryFrame	35
3.1.8. CvReleaseCapture.....	35
3.1.9. cvCaptureFromCAM	35
3.1.10. cvCreateImage	35
3.2. Διάγραμμα ροής του λογισμικού.....	36
3.3. Ανάλυση λειτουργίας προγράμματος.....	36
3.3.1. Έναρξη Προγράμματος.....	36
3.3.2. Λήψη Εικόνας Από WebCamera	39
3.3.3. Επεξεργασία με τον αλγόριθμο Gauss Background-Foreground.....	40
3.3.4. Ανίχνευση κίνησης	40
3.3.4.1. Πρότυπο RGB	42
3.3.5. Καταγραφή και αποθήκευση εικόνας	43
3.3.6. Αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου	46
3.4. Πειραματικά αποτελέσματα	47
3.4.1. Εσωτερικός χώρος	48
3.4.2. Εξωτερικός Χώρος	51
3.4.3. Συνθήκες πολύ χαμηλού φωτισμού	52
3.4.4. Απότομη Εναλλαγή Φωτός.....	53
3.4.4.1. Απότομη Μείωση Φωτισμού.....	53
3.4.4.2. Απότομη αύξηση φωτισμού	55
3.4.5. Περιπτώσεις σκιάς	57
3.4.5.1. Πρώτη περίπτωση εισόδου σκιάς	58
3.4.5.2. Δεύτερη Περίπτωση Εισόδου Σκιάς.....	59
3.4.6. Λειτουργία λογισμικού επί 12 συνεχόμενες ώρες	60
3.4.7. Λειτουργία λογισμικού χωρίς σύνδεση στο διαδίκτυο	62
3.4.8. Λειτουργία λογισμικού με τον Η/Υ σε αδράνεια.....	63
3.5. Συμπεράσματα.....	63
3.6. Κυριότερα προβλήματα που παρατηρήθηκαν	66
3.6. Μελλοντικές επεκτάσεις του λογισμικού	67
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	69
ΠΑΡΑΡΤΗΜΑ Α – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ.....	71
ΠΑΡΑΡΤΗΜΑ Β– ΟΔΗΓΙΕΣ ΣΥΝΔΕΣΗΣ OPENCV ΜΕ VISUAL STUDIO	79
B.1 Σύνδεση Visual Studio-OpenCV	79
B.2 Εγκατάσταση και σύνδεση EASendMail με το MicrosoftVisualStudio ..	82

Εισαγωγή

1.1 Ασφάλεια και επιτήρηση

Η καθημερινότητα ενός σύγχρονου ανθρώπου έχει αλλάξει ριζικά εξαιτίας της ραγδαίας ανάπτυξης της τεχνολογίας. Ο άνθρωπος, στις καθημερινές του δραστηριότητες, δρα από τον περιορισμό του χρόνου και του χώρου. Όσο βελτιώνεται η τεχνολογία αυτοί οι περιορισμοί εξαλείφονται. Έτσι μπορούμε να πραγματοποιήσουμε περισσότερες λειτουργίες και να καλύψουμε τις ανάγκες μας σε πολύ λιγότερο χρόνο καθώς και ο παράγοντας της απόστασης τείνει να γίνεται όλο και μικρότερο εμπόδιο.

Μια από τις σημαντικότερες ανθρώπινες ανάγκες είναι η αίσθηση της ασφάλειας. Η βασική αίσθηση της ασφάλειας ταυτίζεται με την αίσθηση της ασφάλειας του προσωπικού χώρου του κάθε ανθρώπου. Βασικό στοιχείο σε αυτό είναι ο έλεγχος του περιβάλλοντός του μέσω της παρακολούθησης του χώρου αυτού. Η οπτική επαφή είναι ένα μέσο αξιολόγησης του χώρου και της κατάστασης που επικρατεί σε αυτόν. Όταν όμως δεν βρισκόμαστε άμεσα στο χώρο δεν έχουμε την επαφή αυτή.

Η βασικότερη αίσθηση του ανθρώπου είναι αναμφισβήτητη η όραση. Για το λόγο αυτό ήταν πολύ σημαντική η ανακάλυψη και η ενσωμάτωση της κάμερας στους ηλεκτρονικούς υπολογιστές. Με τον τρόπο αυτό ο ηλεκτρονικός υπολογιστής αντιλαμβάνεται το περιβάλλον όπως και το ανθρώπινο μάτι.

Αναγνωρίζοντας την ανάγκη του ανθρώπου για ασφάλεια και την ευκολία που παρέχεται πλέον από τους υπολογιστές για την επιτήρηση χώρων, στην παρούσα εργασία μελετήθηκαν τεχνικές επιτήρησης χώρου και αναπτύχθηκε εφαρμογή με στόχο την παρακολούθηση ελεγχόμενου χώρου με χρήση υπολογιστή και μιας απλής διαδικτυακής κάμερας (webcamera).

1.2 Τι είναι λογισμικό βιντεοεπιτήρησης

Ένα λογισμικό βιντεοεπιτήρησης βοηθά στην παρακολούθηση συμπεριφοράς, δραστηριοτήτων, ή άλλων πληροφοριών που αλλάζουν, συνήθως ανθρώπων, με σκοπό τον επηρεασμό, τη διαχείριση, την καθοδήγηση ή την προστασία τους.

Ως σύστημα βιντεοεπιτήρησης ορίζεται ένα σύστημα το οποίο είναι εγκατεστημένο σε ένα χώρο και έχει τη δυνατότητα λήψης και καταγραφής εικόνας ή ακόμη και ήχου. Η βιντεοεπιτήρηση αποτελεί αναμφίβολα ένα μέτρο πρόληψης, ιδιαίτερα αν εφαρμοσθεί με σωστή σχεδίαση. Από την άλλη πλευρά όμως, επιδρά στη συμπεριφορά των προσώπων, καθώς κάποιος που γνωρίζει ότι παρακολουθείται, ενδέχεται να προσαρμόζει τη συμπεριφορά του, αλλά και να υπόκειται σε ψυχολογική πίεση. Είναι λοιπόν αναγκαίο να γίνεται σωστή και όχι καταχρηστική λειτουργία των συστημάτων βιντεοεπιτήρησης.

Η χρήση συστημάτων βιντεοεπιτήρησης είναι πλέον ιδιαίτερα εκτεταμένη. Χρησιμοποιούνται σε δημόσιους χώρους όπως σε τράπεζες, ξενοδοχεία, εμπορικά καταστήματα με σκοπό την προστασία των πελατών, του προσωπικού αλλά και των εγκαταστάσεων. Χρησιμοποιούνται όμως και σε ιδιωτικούς χώρους για προστασία της σωματικής ακεραιότητας και της ιδιωτικής περιουσίας πολιτών.

Η εγκατάσταση και λειτουργία συστημάτων βιντεοεπιτήρησης σε δημόσιους χώρους από δημόσιες αρχές, για τους σκοπούς των αρχών αυτών, ρυθμίζεται από το ν. 3917/2011. Αν όμως το σύστημα βιντεοεπιτήρησης εγκατασταθεί από έναν ιδιώτη με σκοπό τη βιντεοεπιτήρηση του ιδιωτικού του χώρου, όπως για παράδειγμα του χώρου της οικίας του, τότε δεν υπάρχει η υποχρέωση γνωστοποίησης προς την Αρχή [1].

1.3 Η ιστορία της βιντεοεπιτήρησης

Η βιντεοεπιτήρηση, όπως και όλες οι εφαρμογές της τεχνολογίας, παρουσίασε ραγδαία εξέλιξη από τα πρώτα βήματα της εμφάνισής της μέχρι τη σύγχρονη εποχή. Το γεγονός αυτό συμβαδίζει με τη γενικότερη εξέλιξη της επιστήμης της πληροφορικής. Παρακάτω παρατίθενται χρονολογικά κάποια σημεία άξια αναφοράς της πορείας της βιντεοεπιτήρησης [2].

- 1949: Εκδίδεται το βιβλίο του George Owel "1984" που εισάγει την έννοια της βιντεοεπιτήρησης στο κοινό.
- 1951: Για πρώτη φορά καταγράφονται σε βιντεοταινία εικόνες από ζωντανή μετάδοση τηλεοπτικής κάμερας.
- 1960: Πρώτη καταγεγραμμένη χρήση καμερών βιντεοεπιτήρησης στην πλατεία Trafalgar του Λονδίνου για την παρακολούθηση της βασιλικής οικογένειας.
- 1965: Πρώτες αναφορές της νομοθετικής επιβολής της επιτήρησης στις Η.Π.Α.
- 1966: Η NASA χρησιμοποιεί αναλογικά σήματα για να χαρτογραφήσει την επιφάνεια του φεγγαριού στέλνοντας ψηφιακές εικόνες στη γη.

- 1969: Εκδίδεται πατέντα ευρεσιτεχνίας για το πρώτο οικιακό σύστημα ασφαλείας που χρησιμοποιεί τηλεοπτική παρακολούθηση στη MarieBrown.
- 1972: Η εταιρία TexasInstrumentsπατένταρε την πρώτη ηλεκτρονική κάμερα που δεν χρειάζεται το παραδοσιακό φιλμ.
- 1973: Το τσιπ ανίχνευσης εικόνας CCD (ChargedCoupledDevice) εφευρίσκεται.
- 1980: Τα συστήματα παρακολούθησης επεκτείνονται σε εταιρίες για την πρόληψη κλοπών.
- 1986: Οι επιστήμονες της KODAKεφευρίσκουν τον πρώτο αισθητήρα Megapixel, ικανό να καταγράφει 1,4 εκατομμύρια pixels.
- 1992:Εφευρίσκεται η κάμερα «Νταντά». Η βιομηχανία τείνει να αναπτύσσει κάμερες μικρότερες σε μέγεθος και μεγαλύτερες σε ανάλυση.
- 1996: Εφευρίσκεται η πρώτη IPκάμερα, Neteye 200 από την εταιρία AxisCommunications.
- 1998: Εισάγονται στην αγορά τα DVRs και πραγματοποιείται το δεύτερο μεγάλο κύμα της υιοθέτησης των συστημάτων βιντεοεπιτήρησης.
- 2001: Η επίθεση της 11^{ης} Σεπτεμβρίου στη Νέα Υόρκη άλλαξε τη νοοτροπία στη βιντεοεπιτήρηση. Από μαζικές παρακολουθήσεις επικεντρώθηκαν περισσότερο στην ιδιωτική ασφάλεια.
- 2003:Το σχολείοRoyalPalmMiddleSchoolστοPhoenixεγκαθιστάλογισμικό αναγνώρισης προσώπου για τον εντοπισμό αγνοούμενων παιδιών.
- 2006: Η πόλη του Σικάγο ανακοινώνει το πρόγραμμα “OperationVirutalShield” το πιο εκτεταμένο σύστημα παρακολούθησης πόλεως στις Η.Π.Α.
- 2007: Υπολογίζεται ότι περισσότερες από 97% όλων των πληροφοριών από τηλεπικοινωνίες μεταφέρονται μέσω του Internet.
- Σήμερα: Το Internet έχει καταστήσει δυνατή την χρήση της βιντεοεπιτήρησης από παντού και την παρακολούθησή της από οποιοδήποτε σημείο.

1.4 Τι είναι η ανίχνευση κίνησης;

Με τον όρο ανίχνευση κίνησης ορίζουμε τη διαδικασία που επιβεβαιώνει την αλλαγή στη θέση ενός αντικειμένου σε σχέση με το περιβάλλον του ή την αλλαγή στον περιβάλλοντα χώρο σε σχέση με ένα αντικείμενο. Γενικότερα η ανίχνευση μπορεί να επιτευχθεί με ηλεκτρονικές μεθόδους (κά-

μερα, μικρόφωνο, υπέρυθρες, laser), όπως στο λογισμικό που αναπτύχθηκε, αλλά και από μηχανικές μεθόδους (πλήκτρα σε πληκτρολόγιο)[3].

Η παρακολούθηση ενός ελεγχόμενου χώρου θα ήταν πολύ πιο ακριβής αν δεν υπήρχαν τα εξής προβλήματα:

- Μεταβολή φωτεινότητας της εικόνας
- Αργή απόκριση της κάμερας σε γρήγορη κίνηση του αντικειμένου
- Επηρεασμός του παρασκηνίου από την εμφάνιση νέου προσκηνίου

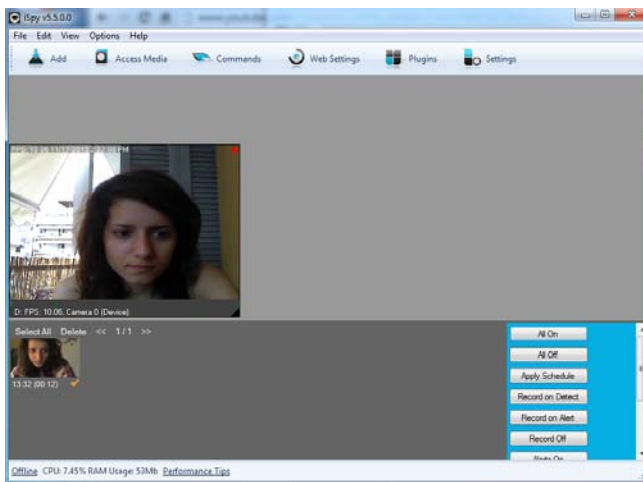
1.5 Ελεύθερα λογισμικά βιντεοεπιτήρησης

Το Ispy είναι μια open source εφαρμογή που επιτρέπει τη σύνδεση κάμερας και μικροφώνου για τον έλεγχο κίνησης και ήχου. Όταν ανιχνευθεί κίνηση τότε το λογισμικό καταγράφει τα δεδομένα σε βίντεο και εικόνες.



Εικόνα 1. ISpylogo

Οι σημαντικότερες υπηρεσίες του λογισμικού είναι η απομακρυσμένη παρακολούθηση μέσω διαδικτύου, η αυτόματη αποστολή βίντεο στην ιστοσελίδα www.youtube.com, πρόσβαση στο λογισμικό μέσω κινητού, εντοπισμός και καταγραφή ήχου, αποστολή ηλεκτρονικού μηνύματος/SMS/μηνύματος στο www.twitter.com κατά την ανίχνευση κίνησης ή την ανίχνευση ήχου, αναγνώριση χρώματος για περιπτώσεις πυρκαγιάς ή καπνού και πολλά άλλα.



Εικόνα 2. Το Λογισμικό I Spy

Οι λειτουργίες αποθήκευσης βίντεο και εικόνας, όταν ανιχνευθεί κίνηση, είναι δωρεάν, όμως δεν ισχύει το ίδιο και για τις υπόλοιπες υπηρεσίες του προγράμματος. Το Ispy είναι ένα λογισμικό που έχει αναπτυχθεί σε c# με Visual Studio 10 και είναι διαθέσιμο σε 17 γλώσσες [4].

1.6 Εφαρμογής που αναπτύχθηκε

Βασικός σκοπός του συστήματος που υλοποιήθηκε είναι να ειδοποιεί το χρήστη σε περίπτωση ανίχνευσης κίνησης στον ελεγχόμενο χώρο. Αυτό επιτυγχάνεται με την υλοποίηση ενός συστήματος παρακολούθησης και ανίχνευσης κίνησης και με την εκμετάλλευση των δυνατοτήτων που παρέχει η βιβλιοθήκη προγραμματισμού OpenCV. Έτσι ξεπερνιέται το εμπόδιο του χώρου και της απόστασης όσον αφορά την άμεσα οπτική επαφή στον επιθυμητό χώρο. Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί σε ιδιωτικά και δημόσια μέρη, όπου υπάρχει η ανάγκη της παρακολούθησης. Γενικά χαρακτηρίζεται ως μια εφαρμογή αυτόματης παρακολούθησης μιας και δεν απαιτεί την ανθρώπινη παρουσία. Περιλαμβάνει την αυτόματη ενημέρωση του χρήστη εφόσον καταγραφεί κάποιο ύποπτο περιστατικό. Επίσης, είναι προσιτό οικονομικά μιας και δε χρειάζεται επιπρόσθετη αγορά εξοπλισμού καθώς χρησιμοποιεί τις συσκευές που υπάρχουν ήδη στους περισσότερους επιτραπέζιους υπολογιστές.

Για την ανάπτυξη της παρούσας εργασίας χρησιμοποιήθηκε ένας φορητός υπολογιστής DELL Inspiron 1564 και η webcam του φορητού από άποψη υλικού. Τα λογισμικά που χρησιμοποιήθηκαν για την ανάπτυξη της εργασίας είναι τα εξής:

- Visual Studio 2010 Express Edition
- OpenCV 2.1.0.
- EASendMail
- Open Office 4.0.0
- Photoshop CS3

Τα βασικότερα συστατικά μέρη της εφαρμογής είναι οι βιβλιοθήκες OpenCV και EASendMail, οι οποίες συνδυάστηκαν με το Visual Studio. Η εφαρμογή εφόσον είναι σε λειτουργία επεξεργάζεται τις εικόνες που παίρνει από την κάμερα με έναν αλγόριθμο αναγνώρισης κίνησης σε πραγματικό χρόνο. Όταν ο αλγόριθμος αναγνωρίσει κίνηση ενεργοποιείται η λειτουργία της αποθήκευσης εικόνων στην κατάλληλη διαδρομή και αποστέλλει μήνυμα στο ηλεκτρονικό ταχυδρομείο.

Περιγραφή προβλήματος

2.1 Κατηγορίες Αλγορίθμων Αναγνώρισης Κίνησης

Κεντρικό σημείο αναφοράς στην παρούσα εργασία είναι η ανίχνευση κίνησης μέσω της κάμερας ενός υπολογιστή. Στον τομέα των αλγορίθμων ανίχνευσης κίνησης υπάρχουν τρεις βασικές κατηγορίες, οι οποίες περιγράφονται συνοπτικά στις παραγράφους που ακολουθούν:

- Κατηγορία αφαίρεσης φόντου
- Κατηγορία χρονικής διαφοράς
- Κατηγορία οπτικής ροής

2.1.1 Αφαίρεση φόντου (Background subtraction)

Η τεχνική αφαίρεσης φόντου είναι μια τεχνική που αναφέρεται στους τομείς της επεξεργασίας εικόνας και υπολογιστικής όρασης όπου το προσκήνιο μιας εικόνας απομονώνεται για περαιτέρω επεξεργασία, όπως για παράδειγμα αναγνώριση αντικειμένου. Σε γενικές γραμμές οι περιοχές που παρουσιάζουν ενδιαφέρον είναι τα αντικείμενα στο προσκήνιο της εικόνας. Η τεχνική αφαίρεσης φόντου είναι μια ευρέως χρησιμοποιούμενη προσέγγιση για την ανίχνευση κινούμενων αντικειμένων σε βίντεο από στατικές κάμερες. Αυτή η μέθοδος όμως, δεν είναι κατάλληλη όταν το φόντο αλλάζει δυναμικά και κινείται, όπως θα μπορούσε να συμβαίνει για παράδειγμα με ένα δέντρο που κινούνται τα φύλλα του λόγω ανέμου ή με μια θάλασσα που έχει κύμα. Το σκεπτικό της προσέγγισης είναι ότι η ανίχνευση κινούμενων αντικειμένων προκύπτει από τη διαφορά μεταξύ της τρέχουσας εικόνας και μιας εικόνας αναφοράς που συχνά αποκαλείται "εικόνα φόντου" ή "μοντέλο φόντου" [5].

2.1.2. Χρονική διαφορά (Temporal differencing)

Η κατηγορία της χρονικής διαφοράς έχει να κάνει με μία διαφορά μεταξύ της χρονικής διάρκειας διαδοχικών καρέ εξετάζοντας εικονοστοιχείο - εικονοστοιχείο. Τα διαδοχικά καρέ μπορεί να είναι είτε δύο, είτε περισσότερα. Η χρονική διαφορά προσαρμόζεται σε περιβάλλοντα που αλλάζουν, αφού το ιστορικό του φόντου ανανεώνεται κάθε λίγα καρέ. Ενώ πρόκειται

για μια μέθοδο που είναι εξαιρετικά προσαρμοστική σε δυναμικά περιβάλλοντα, γενικά κάνει μια φτωχή εργασία στην εξαγωγή των σχημάτων των κινούμενων αντικειμένων [6].

2.1.3. Οπτική ροή (Optical flow)

Πρόκειται για μια προσέγγιση ανίχνευσης κίνησης που καθορίζει κατά πόσο ένα εικονοστοιχείο μιας εικόνας κινείται μεταξύ των γειτονικών του. Η μέθοδος της οπτικής ροής είναι πολύπλοκη, αλλά έχει το πλεονέκτημα της ανίχνευσης κίνησης με ακρίβεια ακόμη και χωρίς να γνωρίζεται το φόντο. Έτσι μπορεί να επιτευχθεί ο σκοπός της αναγνώρισης κίνησης με κινούμενο παρασκήνιο ακόμη και αν η κάμερα κινείται. Η οπτική ροή είναι αρκετά πολύπλοκη υπολογιστικά και απαιτεί η εσωτερική κίνηση των χαρακτηριστικών των καρέ να είναι μικρή. Επίσης, είναι δύσκολη η υλοποίηση σε πραγματικό χρόνο και συχνά απαιτεί εξειδικευμένο υλικό[6].

Για την παρούσα εργασία χρησιμοποιήθηκε αλγόριθμος που κατατάσσεται στην κατηγορία της αφαίρεσης φόντου. Ο παράγοντας της στατικότητας της κάμερας ήταν καταλυτικός στην επιλογή της κατηγορίας αλγορίθμου, καθώς όπως προαναφέρθηκε οι αλγόριθμοι αφαίρεσης φόντου είναι πολύ αποτελεσματικοί σε περιπτώσεις με σταθερό παρασκήνιο.

2.2. Αλγόριθμος Gauss background/foreground detection

2.2.1. Περιγραφή Μοντέλου

Στο μοντέλο μίξης γκαουσιανών κατανομών το κάθε εικονοστοιχείο του παρασκηνίου χαρακτηρίζεται από τη μίξη κατανομών Gauss, όπου το K είναι ένας μικρός αριθμός από το 3 μέχρι το 5. Η κάθε κατανομή Gauss αναπαριστά διαφορετικό χρώμα του παρασκηνίου και οι παράμετροι της μίξης αναπαριστούν τη χρονική αναλογία με την οποία αυτά τα χρώματα παραμένουν στην εικόνα. Τα στοιχεία του background αποφασίζονται υποθέτοντας ότι το φόντο περιέχει τα B χρώματα με την μεγαλύτερη πιθανότητα. Τα πιθανά χρώματα του φόντου είναι αυτά που παραμένουν περισσότερο ώρα και είναι τα πιο στατικά. Σταθερά μονόχρωμα αντικείμενα τείνουν να σχηματίζουν συμπαγείς δέσμες στο χρωματικό χώρο, ενώ τα κινούμενα αντικείμενα σχηματίζουν πιο ευρείς δέσμες εξαιτίας των διαφορετικών ανακλώμενων επιφανειών κατά τη διάρκεια της κίνησης. Η μέτρηση αυτή ονομάζεται fitness value. Για να επιτραπεί στο μοντέλο να προσαρμόζει τις αλλαγές στο φωτισμό και να τις επεξεργάζεται σε πραγματικό χρόνο εφαρμόζεται μια διαδικασία ενημέρωσης (update) η οποία βασίζεται στην επιλεκτική ανανέωση του μοντέλου. Κάθε καινούρια τιμή εικονοστοιχείου συγκρίνεται με τα στοιχεία του υπάρχοντος μοντέλου σε σχέση με το fitness. Στη πρώτη συνιστώσα του μοντέλου που ταιριάζει θα ενημερωθεί. Αν δεν υπάρξει ταιρίασμα, δημιουργείται μια νέα συνιστώσα

Gauss που έχει σαν μέση τιμή την τιμή αυτής της συνιστώσας, πολύ μεγάλη απόκλιση και μια μικρή τιμή σαν παράμετρο βάρους.

2.2.2. Κατάτμηση κίνησης με χρήση στατιστικών μεθόδων

Σχετικά πρόσφατα, κάποιες στατιστικές μέθοδοι για να εξαχθούν οι περι-οχέςαλλαγών από το φόντο έχουν εμπνευστεί από τις βασικές μεθόδους αφαίρεσης φόντου που ήδη περιγράψαμε. Οι στατιστικές αυτές προσεγγί-σεις χρησιμοποιούν χαρακτηριστικά από ατομικά εικονοστοιχεία ή από ομάδες τους για να κατασκευάσουν πιο σύνθετα μοντέλα για το φόντο. Οι στατιστικές αυτών των μοντέλων μάλιστα, αλλάζουν δυναμική κατά τη διάρκεια της επεξεργασίας. Κάθε εικονοστοιχείο στην τρέχουσα εικόνα χαρακτηρίζεται ως εικονοστοιχείο πρώτου πλάνου ή φόντου αφού συ-γκριθεί με τις στατιστικές του ενίοτε μοντέλου φόντου. Αυτή η προσέγγι-ση γίνεται όλο και πιο δημοφιλής λόγω της σθεναρότητας που παρουσιάζει στο θόρυβο, στις σκιές, στις αλλαγές των συνθηκών φωτός κλπ[7].

2.2.3. Ενημέρωση Παραμέτρων

Κάθε pixel της εικόνας μοντελοποιείται από ένα συνδυασμό K κατανο-μών Gauss. Η πιθανότητα ένα συγκεκριμένο pixel να έχει την τιμή x_N τη χρονική στιγμή N μπορεί να γραφεί ως

$$p(x_N) = \sum_{j=1}^K w_j \eta(x_n; \theta_j)$$

όπου w_k είναι η παράμετρος βάρους της k-στης συνιστώσας Gauss και $\eta(x_n; \theta_k)$ είναι η κανονική κατανομή της συνιστώσας αυτής η οποία εκ-φράζεται από τη σχέση :

$$\eta(x; \theta_k) = \eta(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1} (x-\mu_k)}$$

όπου μ_k η μέση τιμή και $\Sigma_k = \sigma_k^2 I$ το γινόμενο απόκλισης από το μέσο όρο της k-στης συνιστώσας. Οι K κατανομές ταξινομούνται με βάση την τιμή $\text{fitness } w_k / \sigma_k$ και οι πρώτες B κατανομές χρησιμοποιούνται σαν το μοντέλο του background της εικόνας αυτής όπου το B υπολογίζεται από τη σχέση

$$B = \arg_b \min \left(\sum_{j=1}^b w_j > T \right)$$

Το T είναι το ελάχιστο κλάσμα του μοντέλου του background στην εικό-να, δηλαδή είναι η ελάχιστη εκ' των προτέρων πιθανότητα εμφάνισης του background στην εικόνα. Η αφαίρεση του background πραγματοποιείται

καθορίζοντας σαν foreground οποιοδήποτε pixel απέχει περισσότερο από 2,5 τυπικές αποκλίσεις από οποιαδήποτε από τις B κατανομές. Η πρώτη συνιστώσα Gauss που ταιριάζει τις τιμές ελέγχου θα ενημερωθεί μέσω των ακόλουθων εξισώσεων :

$$\begin{aligned}\bar{w}_k^{N+1} &= (1-a)\bar{w}_k^N + a\bar{p}(\omega_k|x_{N+1}) \\ \bar{\mu}_k^{N+1} &= (1-a)\bar{\mu}_k^N + \rho x_{N+1} \\ \bar{\Sigma}_k^{N+1} &= (1-a)\bar{\Sigma}_k^N + \rho(x_{N+1} - \bar{\mu}_k^{N+1})(x_{N+1} - \bar{\mu}_k^{N+1})^T \\ \rho &= a\eta(x_{N+1}; \bar{\mu}_k^N, \bar{\Sigma}_k^N) \\ \bar{p}(\omega_k|x_{N+1}) &= 1 \text{ αν } \omega_k \text{ η πρώτη συνιστώσα Gauss που ταιριάζει} \\ \text{αλλιώς } \bar{p}(\omega_k|x_{N+1}) &= 0\end{aligned}$$

όπου ω_k είναι η k-στη συνιστώσα Gauss και $1/a$ ορίζει τη χρονική σταθερά που προσδιορίζει μεταβολές

Αν καμία από τις K κατανομές δεν ταιριάζει με την τιμή του pixel, η ελάχιστη πιθανή συνιστώσα αντικαθίσταται από την συγκεκριμένη τιμή σαν μέση τιμή, μία αρχικά πολύ μεγάλη διακύμανση και μία χαμηλή παράμετρο βάρους. Ουσιαστικά μόνο το a και το T αποτελούν παραμέτρους που πρέπει να οριστούν για την εφαρμογή του αλγορίθμου.

Ωστόσο ο αλγόριθμος όπως εξηγήθηκε μέχρι τώρα έχει δύο πολύ σημαντικά προβλήματα. Καταρχήν, εάν η αρχική τιμή ενός συγκεκριμένου pixel είναι ένα αντικείμενο του foreground, υπάρχει μόνο μία κατανομή Gauss της οποίας το βάρος ισούται με τη μονάδα. Με μόνο ενός χρώματος διαδοχικές τιμές στο background, χρειάζονται $\log_{(1-a)}(T)$ εικόνες μέχρι να αναγνωριστεί το αυθεντικό background και $\log_{(1-a)}(0.5)$ εικόνες μέχρι να βρεθεί η κυρίαρχη συνιστώσα του φόντου. Για παράδειγμα, αν υποθέσουμε ότι τουλάχιστον για 60% του χρόνου το φόντο εμφανίζεται στο βίντεο και $a = 0.002$ (500 τελευταίες εικόνες), θα χρειαστούν 255 και 346 εικόνες για να συμπεριληφθεί η συνιστώσα στο background και για να γίνει η επικρατούσα συνιστώσα φόντου, αντίστοιχα. Η κατάσταση χειροτερεύει ακόμα περισσότερο σε πολύπλοκα περιβάλλοντα όπου ένα στατικό και καθαρό φόντο είναι σπάνιο. Το δεύτερο πρόβλημα που προκύπτει είναι ότι το ρ είναι πάρα πολύ μικρό λόγω του κλάσματος πιθανότητας. Αυτό συνεπάγεται πολύ αργές προσαρμογές στους πίνακες της μέσης τιμής και της διακύμανσης κι έτσι ο αλγόριθμος εκτίμησης του φόντου μπορεί να αποτυγχάνει για αρκετά δευτερόλεπτα μετά την αρχικοποίηση. Μία απλή λύση για το πρόβλημα αυτό είναι να αφαιρέσουμε τον όρο της πιθανότητας από το ρ . Όσο για το πρώτο πρόβλημα, τη λύση τη δίνει ένας EM αλγόριθμος ως εξής:

Αρχικά η εκτίμηση του συνδυαστικού Gauss μοντέλου γίνεται με βάση τα στατιστικά που προκύπτουν από τις ακόλουθες εξισώσεις ενημέρωσης :

$$\begin{aligned}\widehat{w}_k^{N+1} &= \widehat{w}_k^N + \frac{1}{N+1} (\widehat{p}(\omega_k | x_{N+1}) - \widehat{w}_k^N) \\ \widehat{\mu}_k^{N+1} &= \widehat{\mu}_k^N + \frac{\widehat{p}(\omega_k | x_{N+1})}{\sum_{i=1}^{N+1} \widehat{p}(\omega_k | x_i)} (x_{N+1} - \widehat{\mu}_k^N) \\ \widehat{\Sigma}_k^{N+1} &= \widehat{\Sigma}_k^N + \frac{\widehat{p}(\omega_k | x_{N+1})}{\sum_{i=1}^{N+1} \widehat{p}(\omega_k | x_i)} ((x_{N+1} - \widehat{\mu}_k^N)(x_{N+1} - \widehat{\mu}_k^N)^T - \widehat{\Sigma}_k^N)\end{aligned}$$

Η αρχική αυτή εκτίμηση βελτιώνει την ακρίβεια καθώς επίσης και τηναπόδοση του αλγορίθμου επιτρέποντας γρήγορη σύγκλιση σε ένα σταθερό μοντέλο φόντου. Έπειτα, αφού τα πρώτα L δείγματα έχουν επεξεργαστεί, λαμβάνονται υπόψηστις εξισώσεις ενημέρωσης οι οποίες έτσι δίνουν προτεραιότητα στα πιο πρόσφατα δεδομένα με αποτέλεσμα ο αλγόριθμος να προσαρμόζει πιο εύκολα τις διάφορες μεταβολές του περιβάλλοντος. Οι εξισώσεις αυτές, που απαιτούν τα L πιο πρόσφατα δείγματα είναι οι εξής:

$$\begin{aligned}\widehat{w}_k^{N+1} &= \widehat{w}_k^N + \frac{1}{L} (\widehat{p}(\omega_k | x_{N+1}) - \widehat{w}_k^N) \\ \widehat{\mu}_k^{N+1} &= \widehat{\mu}_k^N + \frac{1}{L} \left(\frac{\widehat{p}(\omega_k | x_{N+1}) x_{N+1}}{\widehat{w}_k^{N+1}} - \widehat{\mu}_k^N \right) \\ \widehat{\Sigma}_k^{N+1} &= \widehat{\Sigma}_k^N + \frac{1}{L} \left(\frac{\widehat{p}(\omega_k | x_{N+1}) (x_{N+1} - \widehat{\mu}_k^N)(x_{N+1} - \widehat{\mu}_k^N)^T}{\widehat{w}_k^{N+1}} - \widehat{\Sigma}_k^N \right)\end{aligned}$$

[7]

2.2.4. Κώδικας του αλγόριθμου σε c++

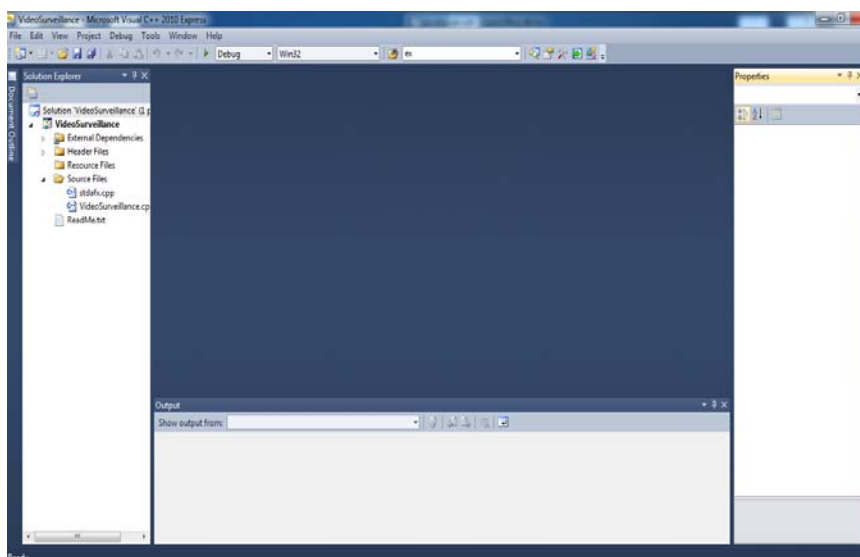
```
/* Demo of the background/foreground detection alg [22] */
/* Author: Francois Cauwe */
#include "cv.h"
#include "cvaux.h"
#include "highgui.h"
#include <ctype.h>
#include <stdio.h>
int main(int argc, char** argv)
{
    /* Start capturing */
    CvCapture* capture = 0;
    if(argc == 1 || (argc == 2 &&strlen(argv[1]) == 1
&&isdigit(argv[1][0])))
    capture = cvCaptureFromCAM(argc == 2 ? argv[1][0] - '0' :
0);
    else if(argc == 2)
    capture = cvCaptureFromAVI(argv[1]);
    if(!capture)
    {
```

```
fprintf(stderr, "Could not initialize...\n");
return -1;
}
/* print a welcome message, and the OpenCV version */
printf ("Demo of the background classification using
CvGaussBGModel %s (%d.%d.%d)\n",
        CV_VERSION,
        CV_MAJOR_VERSION, CV_MINOR_VERSION,
CV_SUBMINOR_VERSION);
/* Capture 1 video frame for initialization */
IplImage* videoFrame = NULL;
videoFrame = cvQueryFrame(capture);
if(!videoFrame)
{
printf("Bad frame \n");
exit(0);
}
// Create windows
cvNamedWindow("BG", 1);
cvNamedWindow("FG", 1);
// Select parameters for Gaussian model.
CvGaussBGStatModelParams* params = new CvGaussBGStat-
ModelParams;
params->win_size=2;
params->n_gauss=5;
params->bg_threshold=0.7;
params->std_threshold=3.5;
params->minArea=15;
params->weight_init=0.05;
params->variance_init=30;
// CreateCvBGStatModel
// cvCreateGaussianBGModel(IplImage* first_frame,
CvGaussBGStatModelParams* parameters)
// or
// cvCreateGaussianBGModel(IplImage* first_frame)
CvBGStatModel* bgModel = cvCreateGaus-
sianBGModel(videoFrame,params);
int key=-1;
while(key != 'q')
{
// Grab a frame
videoFrame = cvQueryFrame(capture);
if(!videoFrame)
break;
// Update model
cvUpdateBGStatModel(videoFrame,bgModel);
// Display results
cvShowImage("BG", bgModel->background);
cvShowImage("FG", bgModel->foreground);
key = cvWaitKey(10);
}
cvDestroyWindow("BG");
cvDestroyWindow("FG");
cvReleaseBGStatModel(&bgModel);
cvReleaseCapture(&capture);
```

```
return 0;  
}
```

2.3. Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE-Integrated Development Environment) από τη Microsoft που βοηθά στη δημιουργία και την ανάπτυξη εφαρμογών και στον εντοπισμό σφαλμάτων. Χρησιμοποιείται για την ανάπτυξη εφαρμογών κονσόλας, Windows Forms εφαρμογών και υπηρεσιών web. Το Microsoft Visual Studio Express είναι μια πιο ελαφριά έκδοση του Microsoft Visual Studio. Η πρώτη έκδοση του Visual Studio Express κυκλοφόρησε τον Οκτώβριο του 2005. Στην παρούσα εργασία χρησιμοποιήθηκε το Microsoft Visual Studio 2010 Express [8].



Εικόνα3. Microsoft Visual Studio 2010 Express Edition

Τα λειτουργικά συστήματα στα οποία μπορεί να εκτελεστεί το Microsoft Visual Studio 2010 Express είναι τα εξής:

- Windows 7 SP1 (x86 και x64)
- Windows 8 (x86 και x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)

Οι υποστηριζόμενες αρχιτεκτονικές με τις οποίες είναι συμβατό το Microsoft Visual 2010 Express είναι είτε 32-bit (x86), είτε 64-bit (x64).

2.3.3. Ελάχιστες Απαιτήσεις Συστήματος

Τα ελάχιστα στοιχεία που απαιτείται να έχει ένας υπολογιστής για να εκτελεστεί το Microsoft Visual 2010 Express είναι τα παρακάτω [9]:

- Επεξεργαστής 1GB ή 2GB
- 10GB ελεύθερος χώρος στον σκληρό δίσκο
- 5400RPM στον σκληρό δίσκο
- DirectX 9 συμβατή κάρτα γραφικών με ανάλυση 1024 x 768 ή υψηλότερη

2.4. Τι είναι βιβλιοθήκη

Στην πληροφορική καλούμε βιβλιοθήκη (library) μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιούνται για την ανάπτυξη λογισμικού. Οι βιβλιοθήκες περιέχουν κώδικα και δεδομένα και βοηθούν υπηρεσίες που εκτελούνται σε προγράμματα. Αυτό επιτρέπει τον διαμοιρασμό και τη χρήση του κώδικα και των δεδομένων με αρθρωτό τρόπο. Τα σύγχρονα λειτουργικά συστήματα παρέχουν βιβλιοθήκες που υλοποιούν την πλειονότητα των υπηρεσιών του συστήματος. Έτσι, ο περισσότερος κώδικας που χρησιμοποιούν οι σύγχρονες εφαρμογές παρέχεται από βιβλιοθήκες και δεν χρειάζεται να γραφτεί για κάθε νέο πρόγραμμα[10].

2.5. Η βιβλιοθήκη OpenCV



Εικόνα 4. Open CV Logo

Η OpenCV είναι μια βιβλιοθήκη μηχανικής όρασης που περιέχει συναρτήσεις που αφορούν την επεξεργασία εικόνας σε πραγματικό χρόνο. Η βιβλιοθήκη αυτή αναπτύχθηκε από την Intel το 1999 με C/C++. Η OpenCV είναι ανοιχτού λογισμικού και είναι ελεύθερη για εμπορική και ερευνητική χρήση υπό την άδεια ανοιχτού λογισμικού BSD. Αυτό σημαίνει ότι μπορεί να γίνει ελεύθερα η χρήση της για εμπορικούς ή ερευνητικούς σκοπούς.

Έχει διεπαφές C, C++, Python, Java και υποστηρίζει Windows, Linux, Mac OS, iOS, και Android. Γράφτηκε αρχικά για compilation σε BorlandC++, MSVC++ και για τους compilers της Intel. Η δημιουργία της OpenCV αποσκοπεί και στη δημιουργία μιας κοινότητας ανοιχτού λογισμικού, σχετική με τη μηχανική όραση, η οποία θα αναπτύσσει σύγχρονες μεθόδους επεξεργασίας εικόνας σε ένα συνεχώς αναπτυσσόμενο τεχνολογικό περιβάλλον.

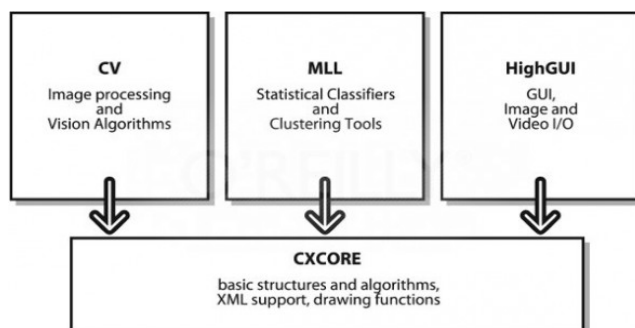
Η OpenCV ξεκίνησε μέσα από μια έρευνα της Intel που αφορούσε την προώθηση εφαρμογών με υψηλές απαιτήσεις από τον επεξεργαστή του συστήματος. Ένας από τους συντάκτες της βιβλιοθήκης που εργαζόταν για την Intel, επισκεπτόταν πανεπιστήμια και παρατήρησε ότι κάποιες από τις κορυφαίες ομάδες πανεπιστημίων, όπως ήταν το MIT Media Lab, είχαν αναπτύξει κώδικα τεχνητής όρασης που μεταβιβαζόταν από φοιτητή σε φοιτητή. Με τον τρόπο του διαμοιρασμού μεταξύ τους, είχαν το πλεονέκτημα της γρηγορότερης και καλύτερης ανάπτυξης εφαρμογών αντί να ανακαλύπτουν συνέχεια βασικές λειτουργίες από το μηδέν.

Η OpenCV σήμερα περιέχει πάνω από 500 συναρτήσεις κατάλληλες για διάφορες εφαρμογές στους τομείς της οπτικής επεξεργασίας εικονών. Η βιβλιοθήκη OpenCV σχεδιάστηκε ώστε να έχει υψηλή υπολογιστική απόδοση και δίνει έμφαση σε εφαρμογές πραγματικού χρόνου. Η βιβλιοθήκη αυτή μπορεί να εκμεταλλευτεί τις δυνατότητες των νέων πολυπύρηνων επεξεργαστών. Οι αλγόριθμοι της OpenCV είναι βελτιστοποιημένοι για επεξεργαστές αρχιτεκτονικής Intel Pentium (MMX, Pro, 3, 4).

Σε περίπτωση που ο χρήστης επιθυμεί τη βελτιστοποίηση της OpenCV μπορεί να αγοράσει τη βιβλιοθήκη IPP (Integrated Performance Primitives Library) της Intel, η οποία αποτελείται από πιστοποιημένες υπορουτίνες χαμηλού επιπέδου που βελτιώνουν την απόδοση των λειτουργιών που προσφέρει η OpenCV. Η βιβλιοθήκη OpenCV δεν εξαρτάται σε καμία περίπτωση από την IPP. Αν η IPP είναι εγκατεστημένη τότε η OpenCV θα τη χρησιμοποιήσει αυτόματα με την εκτέλεση των βιβλιοθηκών δυναμικής σχεδίασης της IPP για να ενισχύσει περαιτέρω την ταχύτητά της [11]

2.5.1. Δομικά Μέρη της OpenCV

Η OpenCV δομείται κυρίως από τέσσερα συστατικά μέρη, το CV, το MLL, το HighGUI και το CxCore που παρουσιάζονται και στην Εικόνα 5.



Εικόνα 5. Συστατικά μέρη της OpenCV

Πιο συγκεκριμένα το πρώτο συστατικό μέρος με την ονομασία CV, αναπαριστά τους βασικούς αλγόριθμους για επεξεργασία εικόνας αλλά και

τους υψηλού επιπέδου αλγόριθμους για τη μηχανική όραση. Το δεύτερο, MLL, αναπαριστά τη βιβλιοθήκη μηχανικής μάθησης που περιλαμβάνει εργαλεία για ομαδοποίηση και για στατιστική ταξινόμηση. Το τρίτο, HighGUI, αναπαριστά τις ρουτίνες εισόδου-εξόδου, τις συναρτήσεις για την προβολή και την αποθήκευση των εικόνων και γενικότερο ότι έχει να κάνει με τη διεπαφή χρήστη-εφαρμογής. Τέλος, το CXCORE περιέχει τις βασικές δομές δεδομένων, όπως το IplImage, τη βασική δομή στην OpenCV που περιγράφει με λεπτομέρεια το κάθε καρέ προς επεξεργασία [12].

2.5.2. Πλεονεκτήματα της OpenCV

Το βασικότερο πλεονέκτημα της OpenCV είναι η ταχύτητα. Αυτός είναι και ο βασικός λόγος που δημιουργήθηκε εξ αρχής γιατί η Intel ήθελε να δείξει πόσο προχωρημένοι ήταν οι επεξεργαστές που μπορούσαν να επεξεργάζονται βίντεο σε πραγματικό χρόνο.

Το δεύτερο είναι ότι ενώ είναι ελεύθερο λογισμικό υποστηρίζεται από εταιρίες όπως η Intel, η IBM και πανεπιστήμια όπως το Standfort, το MIT και πλήθος άλλων οργανισμών και εργαστηρίων που συμβάλλουν συνεχώς στην ανάπτυξή της.

Ένα ακόμη πλεονέκτημα είναι ότι δεν απαιτεί περίπλοκα μηχανήματα και εξοπλισμούς αφού και η πιο απλή webcam είναι αρκετή για την είσοδο εικόνας και ο πιο απλός προσωπικός υπολογιστής είναι ικανός για την επεξεργασία της.

Επίσης, υπάρχουν πολλές έτοιμες μέθοδοι και πολλοί έτοιμοι αλγόριθμοι για το ξεκίνημα κάθε νέου προγραμματιστή στην OpenCV. Η OpenCV έχει περίπου 50.000 χρήστες παγκοσμίως και έχουν γίνει πάνω από επτά εκατομμύρια downloads. Η ανάπτυξή της πλέον δεν αποτελεί αποκλειστική αρμοδιότητα της Intel [11],[12].

2.5.3. Εφαρμογές της OpenCV

Παρακάτω παρουσιάζονται οι κυριότερες εφαρμογές της OpenCV:

- Αναγνώριση προσώπων/αντικειμένων
- Αναγνώριση συναισθημάτων
- Επικοινωνία ανθρώπου-υπολογιστή
- Αυτόματα συστήματα παρακολούθησης
- Συστήματα ελέγχου παραγωγής
- Συρραφή εικόνων από δορυφόρους και διαδικτυακούς χάρτες
- Ευθυγράμμιση σαρωμένων εικόνων

- Μείωση του θορύβου σε ιατρικές εικόνες
- Αντίληψη βάθους (με 2 κάμερες)
- Αναγνώριση ήχων και μουσικής, όπου οι τεχνικές αναγνώρισης εικόνων εφαρμόζονται στο φασματογράφημα του ήχου

Επιπλέον, η OpenCV αποτέλεσε ένα βασικό συστατικό του συστήματος όρασης στο ρομπότ του Stanford “Stanley” το οποίο κέρδισε βραβείο δυο εκατομμυρίων δολαρίων στο διαγωνισμό DARPA Grand Challenge Desert Robot Race [11]. Για την ανάπτυξη της παρούσας εργασίας χρησιμοποιήθηκε η έκδοση 2.1.0. της OpenCV.

2.6. Η βιβλιοθήκη EASendMail

Η βιβλιοθήκη EASendMail είναι μια ελαφριά και γρήγορη υπηρεσία παράδοσης μηνυμάτων ηλεκτρονικού ταχυδρομείου που λειτουργεί με το πρωτόκολλο SMTP. Έχει αναπτυχθεί σε NativeC++ και έχει διεπαφές με C++, C, VB6, Delphi και με άλλες .NET εφαρμογές [13]. Για τη χρήση της βιβλιοθήκης EASendMail πρέπει να γίνει λήψη και εκτέλεση του εγκαταστάτη (installer) από την ιστοσελίδα <http://www.emailarchitect.net/easendmail/>

2.6.1 Βασικά χαρακτηριστικά βιβλιοθήκης EASendMail

Τα βασικότερα χαρακτηριστικά της βιβλιοθήκης είναι τα εξής:

- Αναζήτηση DNS για την αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου χωρίς προκαθορισμένο διακομιστή SMTP
- Εγκυρότητα της δοκιμής της διεύθυνσης ηλεκτρονικού ταχυδρομείου
- Υποστήριξη πολλαπλών συνημμένων αρχείων
- Εισαγωγή κειμένου/html στο περιεχόμενο μηνύματος ηλεκτρονικού ταχυδρομείου
- Επιλογή παραληπτών από βάση δεδομένων
- Ενημέρωση σε περίπτωση αποτυχίας παράδοσης του μηνύματος

Περιορισμοί:

- Δεν είναι δωρεάν
- Δοκιμαστική έκδοση βιβλιοθήκης για διάστημα 30 ημερών [14]

2.7. Τι είναι SMTP

Το πρωτόκολλο SimpleMailTransferProtocol (SMTP) έχει καθιερωθεί για τη μετάδοση μηνυμάτων ηλεκτρονικού ταχυδρομείου στο διαδίκτυο. Το πρωτόκολλο αυτό ξεκίνησε να αναπτύσσεται το 1982 και υποστήριζε κατά βάση μηνύματα απλού κειμένου. Στη συνέχεια όμως αναπτύχθηκαν διάφορα standards που επέτρεπαν την εισαγωγή αρχείων στα μηνύματα ηλεκτρονικού ταχυδρομείου, όπως είναι το MIME (MultipurposeInternetMailExtensions). Για την αποστολή ενός ηλεκτρονικού μηνύματος θα πρέπει ο χρήστης να έχει πρόσβαση σε έναν SMTPServer. Όλα τα προγράμματα ηλεκτρονικής αλληλογραφίας, όπως είναι το MicrosoftOutlook, θα πρέπει να ρυθμιστούν κατάλληλα από το χρήστη για να λειτουργήσουν σωστά. Συγκεκριμένα ο χρήστης θα πρέπει να καθορίσει τον SMTPserver που θα χρησιμοποιήσει για να στείλει και να παραλάβει ηλεκτρονική αλληλογραφία. Οι SMTPservers θα πρέπει να έχουν ανοιχτή μια τουλάχιστον από τις θύρες 25 και 587, έτσι ώστε να μπορούν να επικοινωνήσουν με άλλους SMTPservers για την αποστολή και παραλαβή ηλεκτρονικών μηνυμάτων. Πολλοί SMTPservers χρησιμοποιούν και τις δυο θύρες για λόγους συμβατότητας[15].

Περιγραφή προτεινόμενης λύσης

3.1. Τεκμηρίωση συναρτήσεων που χρησιμοποιήθηκαν

Για την υλοποίηση του λογισμικού ανίχνευσης κίνησης σε πραγματικό χρόνο κύριο συστατικό κομμάτι αποτέλεσαν οι βασικές δομές και οι συναρτήσεις που προσφέρει η βιβλιοθήκη OpenCV. Για την καλύτερη κατανόηση των βασικών δομών και των συναρτήσεων που χρησιμοποιήθηκαν ακολουθεί η ανάλυση και η τεκμηρίωσή τους.

3.1.1. CvCapture

Η δομή `CvCapture`(`typedef struct CvCapture CvCapture`) δεν έχει δημόσια διασύνδεση και χρησιμοποιείται μόνο ως παράμετρος για τις λειτουργίες καταγραφής εικονοσειρών [16].

3.1.2. structIplImage

Το `IplImage` είναι μια δομή που περιέχει μεταδεδομέναεικονών, όπως αριθμός γραμμών, στηλών, καναλιών και τον τύπο δεδομένων που χρησιμοποιείται για κάθε εικονοστοιχείο [17].

3.1.3. CvNamedWindow

Η λειτουργία της συνάρτησης αυτής δημιουργεί ένα παράθυρο το οποίο μπορεί να χρησιμοποιηθεί ως σύμβολο κράτησης θέσης για τις εικόνες. Τα παράθυρα που δημιουργούνται μέσα από τη συνάρτηση αυτή αναφέρονται με τα ονόματά τους. Αν υπάρχει ήδη παράθυρο με το ίδιο όνομα η συνάρτηση δεν κάνει τίποτα.

```
int cvNamedWindow(const char* name, int  
flags=CV_WINDOW_AUTOSIZE)
```

Παράμετρος name

Η μεταβλητή αυτή περιέχει τον τίτλο που θα εμφανίζεται στο παράθυρο που θα δημιουργηθεί.

Παράμετρος flags

Επί του παρόντος η μόνη υποστηριζόμενη τιμή της μεταβλητής είναι `CV_WINDOW_AUTOSIZE`. Αν έχει επισημανθεί τότε το μέγεθος του παραθύρου ρυθμίζεται αυτόματα για να ταιριάζει με την εικόνα που εμφανίζεται. Ο χρήστης δεν μπορεί να αλλάξει το μέγεθος του παραθύρου χειροκίνητα [16].

3.1.4. `CvDestroyWindow`

Η συνάρτηση αυτή καταστρέφει το παράθυρο με το αντίστοιχο όνομα που της δίνεται.

```
void cvDestroyWindow(const char* name)
```

Παράμετρος name

Η μεταβλητή αυτή περιέχει τον τίτλο του παραθύρου που πρόκειται να καταστραφεί [16].

3.1.5. `CvShowImage`

Η συνάρτηση `cvShowImage` εμφανίζει μια εικόνα σε ένα συγκεκριμένο παράθυρο. Αν η συνάρτηση δημιουργίας του παραθύρου περιείχε τη σημαία `CV_WINDOW_AUTOSIZE` τότε η εικόνα θα εμφανιστεί στο αυθεντικό μέγεθός της, αλλιώς θα κλιμακωθεί με βάση το μέγεθος του παραθύρου.

Παράμετρος name

Η μεταβλητή αυτή περιέχει το όνομα του παραθύρου που θα εμφανίσει την εικόνα.

Παράμετρος image

Η μεταβλητή αυτή περιέχει την διεύθυνση μνήμης της εικόνας που θα εμφανιστεί [16].

3.1.6. `CvSaveImage`

Η συνάρτηση `cvSaveImage` αποθηκεύει μια εικόνα σε έναν φάκελο. Η μορφή με την οποία θα αποθηκευθεί η εικόνα εξαρτάται από την επέκταση του αρχείου.

Παράμετρος filename

Είναι η μεταβλητή που περιέχει το όνομα του φακέλου που θα αποθηκευθεί η εικόνα.

Παράμετρος image

Είναι η μεταβλητή που περιέχει τη διεύθυνση μνήμης της εικόνας προς αποθήκευση [16].

3.1.7. CvQueryFrame

Η συνάρτηση αυτή παίρνει ένα στοιχείο μιας εικονοσειράς από την webcamera ή από ένα αρχείο, το αποσυμπιέζει και το επιστρέφει.

```
IplImage* cvQueryFrame(CvCapture* capture)
```

Παράμετρος capture

Η μεταβλητή αυτή περιέχει τη δομή της εικονοσειράς[18].

3.1.8. CvReleaseCapture

```
void cvReleaseCapture(CvCapture** capture);
```

Η συνάρτηση cvReleaseCapture ελευθερώνει τη διεύθυνση μνήμης που δεσμεύει η CvCapture.

Παράμετρος capture

Η μεταβλητή αυτή είναι δείκτης στη δομή καταγραφής εικονοσειράς [16].

3.1.9. cvCaptureFromCAM

Η συνάρτηση αυτή δεσμεύει μνήμη για την ανάγνωση ενός βίντεο από την κάμερα με τη δομή CvCapture.

```
CvCapture* cvCaptureFromCAM(int index)
```

Παράμετρος index

Η μεταβλητή αυτή περιέχει τον δείκτη της κάμερας που θα χρησιμοποιηθεί. Αν υπάρχει μόνο μια κάμερα ή δεν έχει σημασία ποια κάμερα θα χρησιμοποιηθεί τότε η τιμή θα είναι -1. Για την αποδέσμευση της μνήμης χρησιμοποιείται το CvReleaseCapture[18].

3.1.10. cvCreateImage

Η συνάρτηση αυτή δεσμεύει μνήμη και διαθέτει τα δεδομένα της εικόνας

```
IplImage *img = cvCreateImage(CvSize, intdepth, intchannels)
```

Παράμετρος size

Περιέχει τις διαστάσεις της εικόνας (πλάτος, ύψος)

Παράμετρος depth

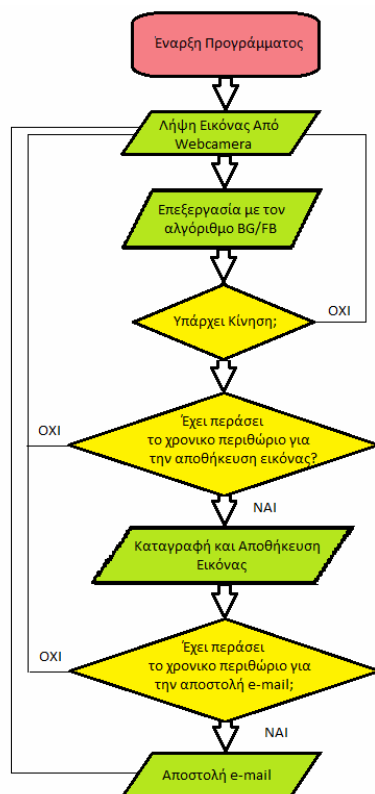
Η μεταβλητή αυτή περιέχει το βάθος των εικονοστοιχείων.

Παράμετρος channels

Η μεταβλητή αυτή περιέχει τον αριθμό καναλιών αναεικονοστοιχείο [19]

3.2. Διάγραμμα ροής του λογισμικού

Στην Εικόνα 6 παρουσιάζεται το διάγραμμα ροής που ακολουθεί το λογισμικό κατά την εκτέλεσή του.



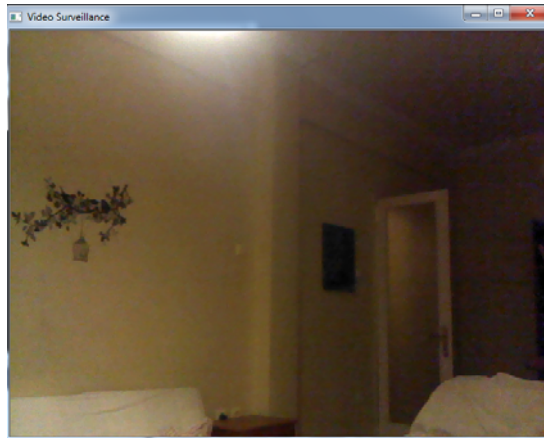
Εικόνα 6. Διάγραμμα ροής λογισμικού

Τα στάδια που ακολουθεί το λογισμικό παρουσιάζονται παρακάτω αναλυτικότερα.

3.3. Ανάλυση λειτουργίας προγράμματος

3.3.1. Έναρξη Προγράμματος

Το λογισμικό ξεκινά τη λειτουργία του με την εκτέλεση του αρχείου VideoSurveillance.exe. Για λόγους εύκολης προσβασιμότητας έχει χρησιμοποιηθεί μια συντόμευση στην επιφάνεια εργασίας του υπολογιστή που αναπτύχθηκε η εργασία. Στη περίπτωση που δεν εκτελούνταν μέσω της συντόμευσης θα χρειαζόταν η εκκίνηση του προγράμματος VisualStudio 2010 Express και μέσα από αυτό να γίνει η εκτέλεση του λογισμικού.



Εικόνα 7. Τυπικό παράδειγμα εσωτερικού χώρου

Ο χρήστης του λογισμικού δε χρειάζεται να κάνει τίποτα άλλο πέρα από την εκτέλεση της εφαρμογής. Παρόλα αυτά δίνεται η δυνατότητα παραμετροποίησης του συστήματος. Το λογισμικό για λόγους ευελιξίας χρησιμοποιεί μεταβλητές που διαβάζει μέσα από ένα αρχείο. Τα δεδομένα που υπάρχουν μέσα στο αρχείο αφορούν κυρίως τις παραμέτρους που χρησιμοποιεί ο αλγόριθμος και τις παραμέτρους που χρησιμοποιούνται για την αποστολή ηλεκτρονικού μηνύματος. Έτσι, εκτός από τον προγραμματιστή που μπορεί να παραμετροποιήσει το σύστημα κατά την εγκατάστασή του, μπορεί και ο χρήστης του προγράμματος αλλάζοντας τιμές στο αρχείο settings.ini.



Εικόνα 8. Τυπικό παράδειγμα εξωτερικού χώρου

Όταν ενεργοποιείται η κάμερα παρατηρείται ότι το πρώτο καρτέ που φαίνεται στο παράθυρο VideoSurveillance είναι μαύρο και σταδιακά τα εικονοστοιχεία παίρνουν τη σωστή τους μορφή. Έτσι, κατά την εκκίνηση του προγράμματος τα δέκα πρώτα καρτέ δεν λαμβάνονται υπόψη από το λογι-

σμικό για να επεξεργαστούν. Αυτό είναι ένα χρονικό περιθώριο που έχει δοθεί στο σύστημα μέχρι να ξεκινήσει να εξάγει σωστά την εικόνα από την κάμερα στο παράθυρο VideoSurveillance.

Κατά την εκκίνηση του προγράμματος αρχικοποιούνται οι μεταβλητές που χρησιμοποιούνται. Όπως προαναφέρθηκε οι τιμές των μεταβλητών που χρησιμοποιεί το πρόγραμμα είναι αποθηκευμένες στο αρχείο settings.ini και είναι οι εξής:

- Save_path
- Win_size
- N_gauss
- Bg_threshold
- Std_threshold
- minArea
- weight_init
- variance_init
- detection_threshold
- detection_value_threshold
- mail_delay
- save_delay
- mail_from
- mail_server_addr
- mail_serverport
- mail_username
- mail_password
- mail_to

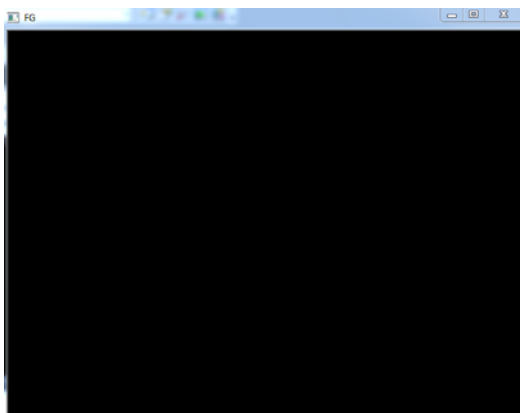
Η πρώτη μεταβλητή είναι το όνομα του φακέλου στον οποίο αποθηκεύονται οι εικόνες. Οι επτά επόμενες μεταβλητές χρησιμοποιούνται για την παραμετροποίηση του αλγόριθμου. Η μεταβλητή detection_threshold αφορά το ποσοστό ανοχής του συστήματος σχετικά με τα λευκά εικονοστοιχεία που παρουσιάζονται και η μεταβλητή detection_value_threshold είναι μια τιμή που συγκρίνεται με το κάθε εικονοστοιχείο για να χαρακτηριστεί λευκό ή μαύρο. Οι μεταβλητές mail_delay αφορά το χρόνο που θα μεσολαβεί από το ένα ηλεκτρονικό μήνυμα στο επόμενο και η μεταβλητή save_delay τον χρόνο που θα μεσολαβεί από την μια αποθήκευση

μιας εικόνας στην επόμενη. Οι έξι τελευταίες μεταβλητές αφορούν παραμέτρους που είναι απαραίτητες για την αποστολή ηλεκτρονικού μηνύματος.

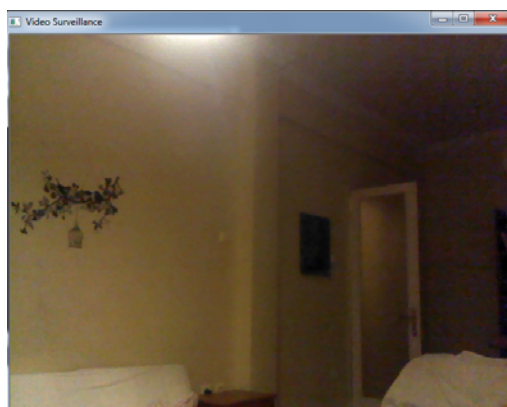
Επίσης, αρχικοποιούνται και δυο μεταβλητές που θα συγκρίνονται με τις `save_delay` και `mail_delay` παίρνοντας πολύ μεγάλες τιμές. Αυτό συμβαίνει ώστε στην πρώτη ανίχνευση κίνησης να αποθηκευθεί η εικόνα και να αποσταλεί με μήνυμα ηλεκτρονικού ταχυδρομείου και να μην χρειαστεί η εξάντληση του χρονικού περιθωρίου.

3.3.2. Λήψη Εικόνας Από WebCamera

Αμέσως μετά την εκκίνηση του προγράμματος ενεργοποιείται η κάμερα του υπολογιστή. Ουσιαστικά το λογισμικό κάθε στιγμή παράγει δυο εικόνες.



Εικόνα 9. Παράθυρο FG



Εικόνα 10. Παράθυρο Video Surveillance

Στην Εικόνα 9 φαίνεται η εικόνα που παράγει ο αλγόριθμος Gauss που βρίσκεται στο παράθυρο με τίτλο FG. Το παράθυρο FG δεν είναι ορατό

στον χρήστη καθώς οι πληροφορίες που παράγονται σε αυτό δεν είναι χρήσιμες σε αυτόν, αλλά μόνο στο λογισμικό. Παρατηρώντας το σχήμα (αριθμός) βλέπουμε ότι ο αλγόριθμος Gauss εξαγάγει μια εικόνα στην οποία φαίνονται άσπρα και μαύρα εικονοστοιχεία. Τα εικονοστοιχεία με το μαύρο χρώμα είναι αυτά στα οποία δεν παρατηρήθηκε κάποια αλλαγή από το προηγούμενο καρέ, ενώ αυτά με το λευκό είναι αυτά που έχουν αλλάξει.

Η δεύτερη εικόνα είναι η εικόνα που λαμβάνει η κάμερα του υπολογιστή και φαίνεται στο παράθυρο που έχει τίτλο VideoSurveillance. Το παράθυρο VideoSurveillance, που φαίνεται στην εικόνα 10, χρησιμοποιείται για την εμφάνιση της εικόνας που λαμβάνει η κάμερα αλλά και για την αποθήκευσή της στην περίπτωση που ανιχνευθεί κίνηση.

3.3.3. Επεξεργασία με τον αλγόριθμο Gauss Background-Foreground

Στο σημείο αυτό το πρόγραμμα λαμβάνει τις εικόνες και τις συγκρίνει με τον αλγόριθμο Gauss, όπως περιγράφεται στο Κεφάλαιο 2. Συγκεκριμένα, ο αλγόριθμος που βοηθά στην ανίχνευση κίνησης στην παρούσα εργασία μπορεί να βρεθεί στην ηλεκτρονική διεύθυνση

<http://neverland.cauwe.org/fcauwe/mocap/files/cvGaussianBGModelDemo.cpp>

3.3.4. Ανίχνευση κίνησης

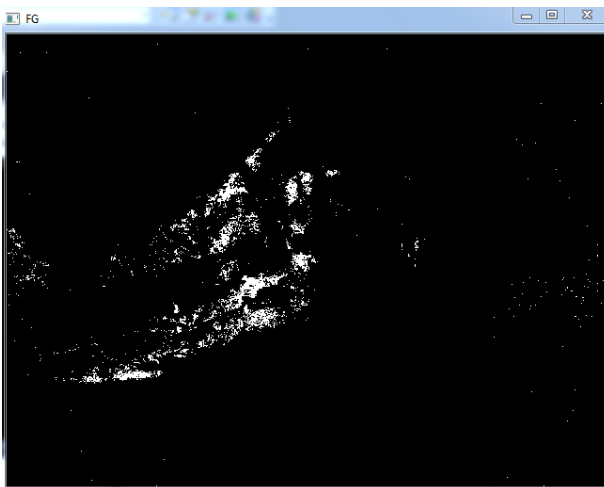
Η αναγνώριση κίνησης στο λογισμικό συμβαίνει όταν υπάρχει μεταβολή των εικονοστοιχείων σε ποσοστό μεγαλύτερο του 1.5% επί του συνόλου των εικονοστοιχείων στο παράθυρο FG όπως φαίνεται στην εικόνα 11. Η τιμή της απόκλισης μπορεί να αλλάξει αντικαθιστώντας την αντίστοιχη τιμή στο αρχείο settings.ini. Αυτή η απόκλιση λαμβάνεται υπόψη γιατί υπάρχει ο παράγοντας του θορύβου.

Στην επιστήμη της πληροφορικής ο θόρυβος (SNR-Signal to Noise Ratio) αφορά ανεπιθύμητες παραμορφώσεις σε σύνολα δεδομένων. Οι παραμορφώσεις αυτές μπορεί είτε να είναι συστηματικές είτε τυχαίες. Εφόσον τα δεδομένα κατά κανόνα εκφράζονται ως αλληλουχίες αριθμών, ο θόρυβος δεν είναι παρά αριθμητικές τιμές οι οποίες προσθαφαιρούνται στα δεδομένα και αλλοιώνουν το επιθυμητό αποτέλεσμα[20].

Στην παρούσα εργασία συναντάται αρκετά συχνά το φαινόμενο του θορύβου. Το αποτέλεσμα του θορύβου είναι φανερό στις εικόνες που απεικονίζουν το παράθυρο FG που παράγει ο αλγόριθμος. Παρατηρούμε πως εμφανίζονται τυχαία στην εικόνα διασκορπισμένα εικονοστοιχεία με λευκό χρώμα, ενώ στην πραγματικότητα δεν θα έπρεπε να ήταν. Για τον λόγο αυτό εισήχθη η παράμετρος που αφορά το ποσοστό των λευκών εικονο-

στοιχείων επί το συνολικό άθροισμα αυτών, ώστε να αποφευχθούν λανθασμένοι συναγερμοί που οφείλονται στον θόρυβο.

Έτσι όταν εντοπιστούν παραπάνω από 1.5% λευκά εικονοστοιχεία αυτό σημαίνει ότι ανιχνεύθηκε κίνηση. Συμπερασματικά, για τη λήψη της απόφασης αν υφίσταται κίνηση ή όχι χρειάζονται δυο δεδομένα, ο αριθμός των συνολικών εικονοστοιχείων ανά καρέ και ο αριθμός των λευκών εικονοστοιχείων μέσα σε αυτά. Το ποσοστό των εικονοστοιχείων που επιτρέπουμε να είναι λευκά μπορεί να αλλάξει μέσα από το αρχείο settings.ini. Κατεβάζοντας το ποσοστό το σύστημα γίνεται πιο «ευαίσθητο» σε αλλαγές, ενώ ανεβάζοντας την τιμή του, το αντίθετο.



Εικόνα 11. Παράθυρο FG όταν γίνεται αντιληπτή κίνηση

Για τον υπολογισμό των συνολικών εικονοστοιχείων ανά καρέ εξάγουμε τα δεδομένα του ύψους και του πλάτους της εικόνας του παραθύρου FG. Με τις δυο αυτές μεταβλητές είναι εύκολο να υπολογίσουμε πως:

$$\text{Σύνολο_Εικονοστοιχείων} = \text{Ύψος} * \text{Πλάτος}$$

Εφόσον ο αριθμός των εικονοστοιχείων της εικόνας είναι πλέον διαθέσιμος είναι εύκολο να υπολογίσουμε το ποσοστό ανοχής του συστήματος. Για τον υπολογισμό της τιμής αυτής χρειάζεται μόνο να εφαρμοσθεί ο μαθηματικός τύπος

$$\text{Ποσό_Ανοχής} = \frac{\text{Σύνολο_Εικονοστοιχείων} * \text{ΕπιθυμητόΠοσοστό}}{100}$$

Για τον υπολογισμό των λευκών εικονοστοιχείων του παραθύρου FG εξετάζουμε κάθε εικονοστοιχείο ξεχωριστά αν πλησιάζει ή όχι στο λευκό. Για την κατηγοριοποίηση αυτή χρησιμοποιήθηκε το πρότυπο χρώματος RGB.

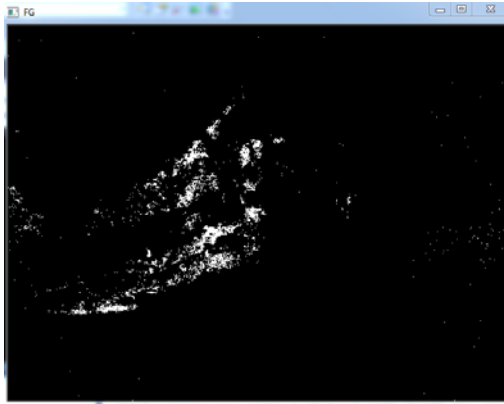
3.3.4.1. Πρότυπο RGB

Το RGBείναι ένα πρότυπο χρώματος στο οποίο περιγράφονται οι τιμές των κόκκινων εικονοστοιχείων (Red)των πράσινων (Green)και των μπλε (Blue). Οι τιμές των χρωμάτων αυτών αναπαρίστανται με τιμές που κυμαίνονται από το 0 μέχρι το 255. Με τον τρόπο αυτό μπορούν να περιγραφούν όλα τα χρώματα, για παράδειγμα το κόκκινο είναι το 255,0,0 [21].

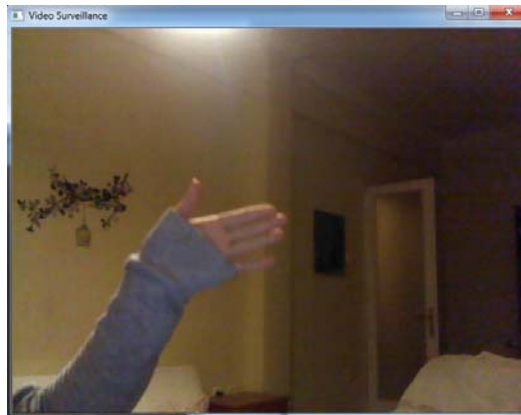
Χρησιμοποιώντας το πρότυπο χρώματος RGBελέγχουμε σε κάθε καρέ τις τιμές όλων των εικονοστοιχείων της εικόνας. Στην περίπτωση που η τιμή είναι μικρότερη του 200 (στο πρότυπο RGBμεταφράζεται ως 200, 200, 200) αυτό σημαίνει οτι το εικονοστοιχείο είναι μαύρο. Αντίστοιχα, αν η τιμή του εικονοστοιχείου είναι μεγαλύτερη του 200 τότε σημαίνει ότι είναι λευκό. Στην περίπτωση που ένα εικονοστοιχείο αναγνωριστεί ως λευκό τότε ο μετρητής των λευκών εικονοστοιχείων, που αρχικοποιείται με την τιμή μηδέν σε κάθε καρέ, αυξάνεται κατά ένα.

Με τη μεθοδολογία αυτή μπορεί πλέον να γίνει σύγκριση μεταξύ των μεταβλητών για να καταλήξουμε στο συμπέρασμα αν υπάρχει κίνηση ή όχι. Αν λοιπόν ο αριθμός των λευκών εικονοστοιχείων είναι μικρότερος από το ποσοστό της ανοχής που δίνουμε στο σύστημα τότε δεν αναγνωρίζεται κίνηση και το λογισμικό θα λάβει νέα εικόνα για σύγκριση. Αν όμως ο αριθμός των λευκών εικονοστοιχείων είναι μεγαλύτερος από την τιμή της ανοχής τότε το λογισμικό προχωρά στην καταγραφή και την αποθήκευση της εικόνας και αν επιτρέπεται από το σύστημα στην αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου. Η εικόνα που θα καταγραφεί είναι αυτή που φαίνεται στο παράθυρο "VideoSurveillance" και όχι αυτή στο "FG" καθώς η πρώτη παρέχει τις πληροφορίες που θέλουμε.

Αξίζει να σημειωθεί πως οι διαστάσεις της εικόνας που παράγει η κάμερα του υπολογιστή στον οποίο αναπτύχθηκε η εργασία είναι 640 x 480 που σημαίνει ότι παράγονται συνολικά 307.200 εικονοστοιχεία σε κάθε καρέ. Έτσι για να ανιχνευθεί κίνηση στο σύστημα αυτό απαιτείται η ανίχνευση τουλάχιστον 4609 λευκών εικονοστοιχείων στο παράθυρο FG.



Εικόνα 12. Παράθυρο FG όταν εισέρχεται αντικείμεμο στο πλάνο

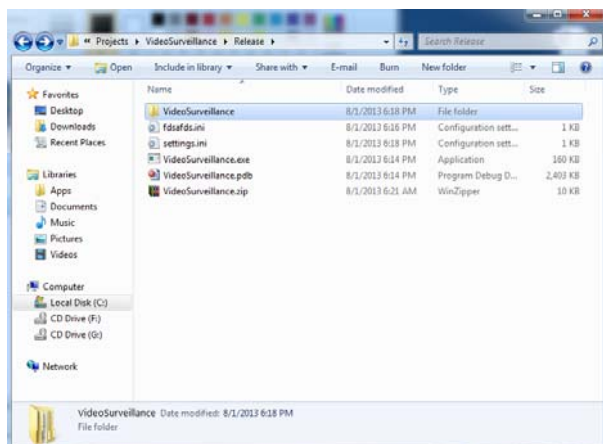


Εικόνα 13. Παράθυρο Video Surveillance όταν εισέρχεται αντικείμεμο στο πλάνο

3.3.5. Καταγραφή και αποθήκευση εικόνας

Όταν ανιχνεύεται κίνηση στο παράθυρο FG με τον τρόπο που περιγράψαμε στο 3.3.4. οι λειτουργίες της καταγραφής και της αποθήκευσης της εικόνας ενεργοποιούνται. Αυτή είναι η βασικότερη λειτουργία για την καταγραφή της δραστηριότητας του περιβάλλοντος που παρακολουθεί το σύστημα. Για την αποθήκευση της εικόνας χρησιμοποιήθηκε η συνάρτηση `cvSaveImage` της οποίας η λειτουργία έχει περιγραφεί προηγουμένως.

Αρχικά το πρόγραμμα ελέγχει αν υπάρχει φάκελος με την ημερομηνία στην οποία πρόκειται να αποθηκεύσει και αν δεν υπάρχει τον δημιουργεί. Στην εικόνα 14 φαίνεται η δομή των αρχείων της εργασίας που αναπτύχθηκε που περιλαμβάνει και τον φάκελο που δημιουργείται.



Εικόνα 14. Φάκελος VideoSurveillance

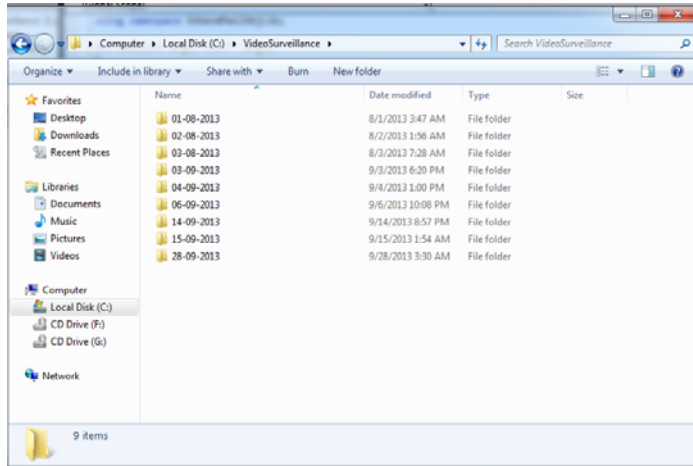
Το λογισμικό χρησιμοποιεί την ημερομηνία και την ώρα για τη σωστή κατάταξη των δεδομένων. Η ημερομηνία χρησιμοποιείται για τη διεύθυνση που αποθηκεύονται τα δεδομένα, ενώ η ώρα χρησιμοποιείται για τον τίτλο της κάθε εικόνας. Για την εξεύρεση της ημερομηνίας και της ώρας χρησιμοποιείται η συνάρτηση `strftime`, η οποία διοχετεύει σε δυο μεταβλητές τις τιμές τους.

Για την περίπτωση δημιουργίας φακέλου χρησιμοποιείται μια μεταβλητή που περιέχει μορφοποιημένα δεδομένα ημερομηνίας. Τα δεδομένα αυτά επιστρέφονται από τη συνάρτηση `strftime`. Στη συνέχεια η μεταβλητή αυτή «προστίθεται» στη διαδρομή που έχει επιλεγεί από τον προγραμματιστή/χρήστη μέσω του αρχείου `settings.ini`. Για την περίπτωση ονομασίας εικόνων χρησιμοποιείται μια μεταβλητή που περιέχει μορφοποιημένα δεδομένα ώρας. Τα δεδομένα αυτά επιστρέφονται από τη συνάρτηση `strftime`. Στη συνέχεια η μεταβλητή αυτή «προστίθεται» ξανά στην προαναφερθείσα επιλεγμένη διαδρομή και στη συνέχεια «προστίθεται» και η κατάληξη “`jpg`” του αρχείου.

Συνοψίζοντας, η αποθήκευση των εικόνων θα γίνεται στη διαδρομή

```
\\VideoSurveillance\[ HH-MM-XXXX ]\
```

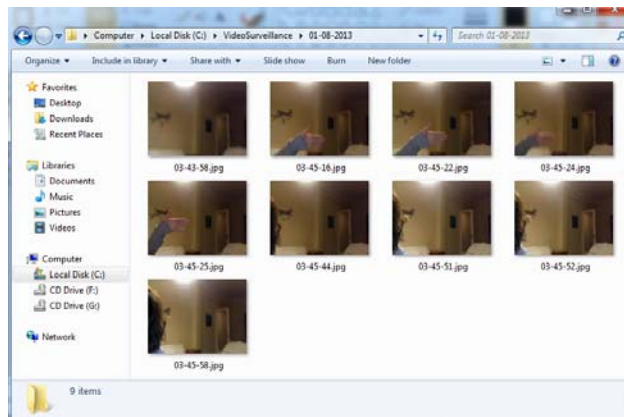
μέσα στον φάκελο στον οποίο αναπτύχθηκε η εργασιάσπως φαίνεται στην εικόνα 15.



Εικόνα 15. Φάκελοι με ημερομηνίες που περιέχονται στον φάκελο VideoSurveillance

Επίσηςόπως φαίνεται και στην εικόνα 16, ο τίτλος της κάθε εικόνας που θα αποθηκεύεται θα είναι η ώρα που αυτή καταγράφηκε, δηλαδή θα είναι

[ΩΩ-ΛΛ-ΔΔ].jpg



Εικόνα 16. Εικόνες με τίτλο την ώρα που αποθηκεύθηκαν

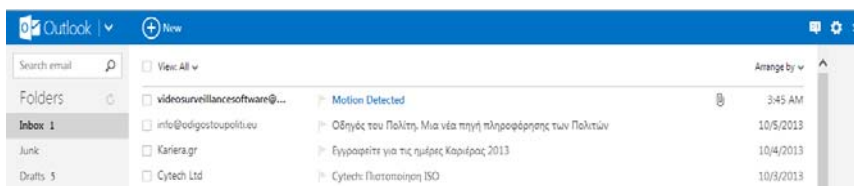
Με τον τρόπο αυτό η κάθε εικόνα έχει διαφορετική ονομασία και θα είναι πιο εύκολα προσβάσιμο σε κάποια αναζήτηση που θα κάνει ο χρήστης. Το μόνο που θα χρειαστεί να κάνει ο χρήστης του προγράμματος είναι να ψάξει στον φάκελο “VideoSurveillance” τον φάκελο με τίτλο την ημερομηνία που θέλει και την εικόνα με τίτλο την ώρα που τον ενδιαφέρει.

Στο σημείο αυτό σκόπιμο είναι να αναφερθεί πως ο λόγος για τον οποίο τα δεδομένα αποθηκεύονται ως εικόνες και όχι ως εικονοσειρές είναι για να μην γεμίσει άσκοπα ο δίσκος δεδομένα. Οι εικόνες που αποθηκεύονται παρατηρήθηκε ότι έχουν μέγεθος από 60KB μέχρι 120KB από ένα δείγμα 450 εικόνων. Η συχνότητα με την οποία το λογισμικό αποθηκεύει εικόνες

βρίσκεται στο αρχείο settings.ini, δηλαδή το πρόγραμμα περιμένει κάποια δευτερόλεπτα για την επόμενη αποθήκευση με την προϋπόθεση πάντα πως έχει ανιχνευθεί κίνηση στο περιβάλλον που παρακολουθεί το σύστημα. Παρόλα αυτά, αν ο χρήστης το επιθυμεί μπορεί να χρησιμοποιηθεί μικρή τιμή στη μεταβλητή που περιέχει το χρόνο που μεσολαβεί από τη μια αποθήκευση στην επόμενη και να αυξήσει το πλήθος των εικόνων που αποθηκεύονται.

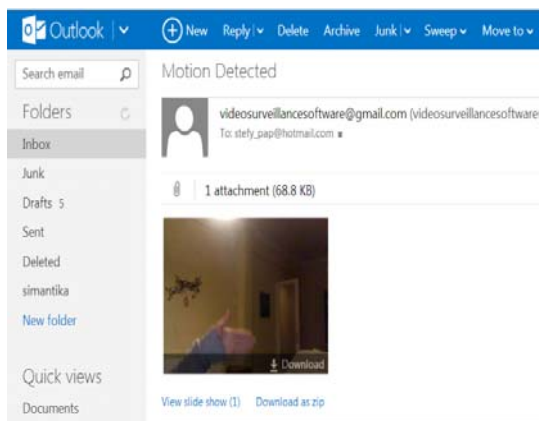
3.3.6. Αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου

Όταν έχει αναγνωριστεί κίνηση στο περιβάλλον που παρακολουθεί το λογισμικό, τότε δίνεται η δυνατότητα ειδοποίησης στον χρήστη μέσω μηνύματος ηλεκτρονικού ταχυδρομείου. Ένα τέτοιο μήνυμα φαίνεται στην εικόνα 17.



Εικόνα 17. Νέα ειδοποίηση μέσω ηλεκτρονικού ταχυδρομείου

Στο ηλεκτρονικό μήνυμα, όπως φαίνεται και στην εικόνα 18, επισυνάπτεται και η λήψη που αποθηκεύθηκε στον ηλεκτρονικό υπολογιστή στον οποίο εκτελείται η εφαρμογή.



Εικόνα 18. Μήνυμα ηλεκτρονικού ταχυδρομείου από την εφαρμογή

Έτσι μόλις ανιχνευθεί κίνηση ο χρήστης του προγράμματος λαμβάνει μήνυμα στο ηλεκτρονικό του ταχυδρομείο από την ηλεκτρονική διεύθυνση που έχει οριστεί στο αρχείο settings.ini. Το θέμα του ηλεκτρονικού μηνύματος θα είναι "MotionDetected" και το μήνυμα θα παρακινήσει τον χρήστη να μην απαντήσει καθώς κάτι τέτοιο θα ήταν άσκοπο.

Η αποστολή ηλεκτρονικού μηνύματος έγινε με τη βοήθεια της βιβλιοθήκης EASendMail, η οποία αναλύθηκε στο προηγούμενο κεφάλαιο. Για την αξιοποίησή της χρειάστηκε μια διεύθυνση ηλεκτρονικού ταχυδρομείου η οποία δημιουργήθηκε γι' αυτόν τον σκοπό και μόνο, η «videosurveillancesoftware@gmail.com».

Χρησιμοποιώντας, λοιπόν, τα στοιχεία της διεύθυνσης αυτής παραμετροποιήθηκαν οι μεταβλητές που σχετίζονται με τον αποστολέα του μηνύματος, όπως το όνομα χρήστη, ο κωδικός χρήστη, και ο τίτλος του αποστολέα. Στη συνέχεια, παραμετροποιήθηκαν οι μεταβλητές που σχετίζονται με το ηλεκτρονικό μήνυμα όπως το θέμα, το περιεχόμενο, και το συνημμένο αρχείο του μηνύματος. Επίσης, χρειάστηκε ο ορισμός της smtpδιεύθυνσης, που στην περίπτωση του gmail είναι το smtp.gmail.com και της θύρας εξυπηρετητή που ορίστηκε σε 465. Τέλος, χρειάστηκε φυσικά να οριστεί και η ηλεκτρονική διεύθυνση παραλήπτη ώστε να λαμβάνει τα ηλεκτρονικά μηνύματα και τα στιγμιότυπα που τον ενδιαφέρουν. Τα στοιχεία αυτά υπάρχουν μέσα στο αρχείο settings.ini και όχι στον πηγαίο κώδικα.

Στο σημείο αυτό σκόπιμο θα ήταν να αναφερθεί ότι το σύστημα παρεμποδίζει το spam στον χρήστη του λογισμικού. Το spam δημιουργεί προβλήματα, όπως απώλεια πόρων και παραγωγικότητας. Για το λόγο αυτό το λογισμικό έχει σχεδιαστεί έτσι ώστε να στέλνει μόνο ένα ηλεκτρονικό μήνυμα ανά δέκα λεπτά. Φυσικά η συχνότητα αποστολής ηλεκτρονικών μηνυμάτων είναι παραμετροποιήσιμη καθώς η τιμή αυτή υπάρχει στο αρχείο settings.ini.

Για την επίτευξη αυτού του σκοπού έχει χρησιμοποιηθεί μια μεταβλητή που υπολογίζει τον χρόνο που μεσολαβεί από την αποστολή ενός ηλεκτρονικού μηνύματος στο επόμενο. Αυτό συμβαίνει με τη σύγκριση της ώρας που στάλθηκε το προηγούμενο μήνυμα με την τωρινή, με τη βοήθεια της συνάρτησης difftime. Όπως αναφέρθηκε και στο Κεφάλαιο 1, με την εκκίνηση του προγράμματος η μεταβλητή της ώρας του προηγούμενου απεσταλμένου μηνύματος παίρνει μια μεγάλη τιμή ώστε να είναι σε ετοιμότητα για την αποστολή νέου ηλεκτρονικού μηνύματος. Μόλις ανιχνευθεί κίνηση, όμως, η μεταβλητή της ώρας που στάλθηκε το προηγούμενο ηλεκτρονικό μήνυμα, ανανεώνεται και παίρνει την τωρινή τιμή της ώρας.

3.4. Πειραματικά αποτελέσματα

Για το πειραματικό μέρος της εργασίας πραγματοποιήθηκαν κάποιες δοκιμές του λογισμικού σε διάφορες καταστάσεις. Αυτές είναι οι εξής:

- σε εσωτερικό χώρο
- σε εξωτερικό χώρο

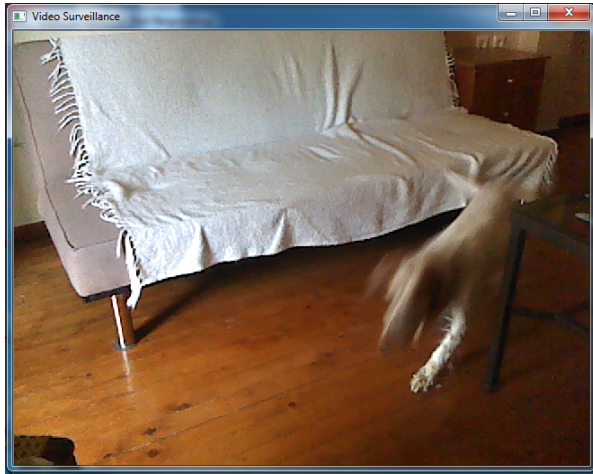
- σε συνθήκες χαμηλού φωτισμού
- σε αυξομειώσεις της έντασης του φωτός
- σε είσοδο σκιάς στο πλάνο
- 12ωρη συνεχόμενη λειτουργία του λογισμικού
- χωρίς σύνδεση στο διαδίκτυο
- με τον υπολογιστή υπό αδράνεια

Για τη δοκιμή του προγράμματος παραμετροποιήθηκε η μεταβλητή του χρόνου που θα μεσολαβεί από το ένα μήνυμα ηλεκτρονικού ταχυδρομείου στο επόμενο στα 20 δευτερόλεπτα για λόγους ευκολίας. Τα πειραματικά αποτελέσματα πραγματοποιήθηκαν σε φορητό υπολογιστή Dell, του οποίου τα βασικά τεχνικά χαρακτηριστικά είναι:

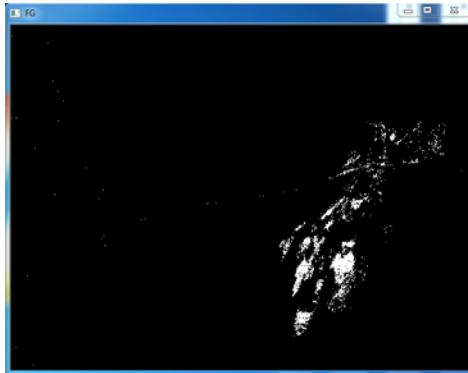
- Επεξεργαστής Intel(R) Core(TM) i3 2.13GHz0
- Μνήμη 3.00 GB
- Αρχιτεκτονική Συστήματος 32-bit
- Λειτουργικό Σύστημα Windows 7 Ultimate

3.4.1. Εσωτερικός χώρος

Στην απλή περίπτωση της ανίχνευσης κίνησης σε εσωτερικό χώρο παρατηρούμε ότι το λογισμικό λειτουργεί σωστά στον εντοπισμό της κίνησης στο πλάνο. Χαρακτηριστικά στο παράθυρο FG της εικόνας 20 απεικονίζονται τα λευκά εικονοστοιχεία που εκλαμβάνονται ως κίνηση από το λογισμικό καθώς υπάρχει μεταβολή στο χρώμα των εικονοστοιχείων. Το λογισμικό, όπως έχει προαναφερθεί, δημιουργηθεί και το παράθυρο “VideoSurveillance” (εικόνα 19) το οποίο είναι το μοναδικό ορατό παράθυρο προς τον χρήστη.

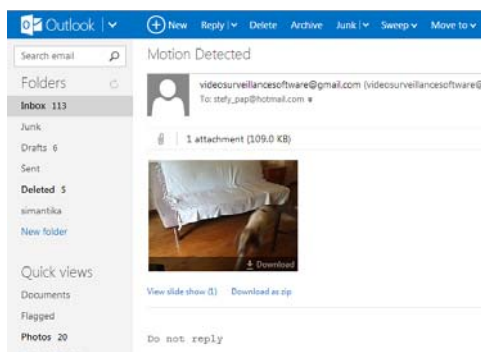


Εικόνα 19. Παράθυρο VideoSurveillance σε πείραμα εσωτερικού χώρου

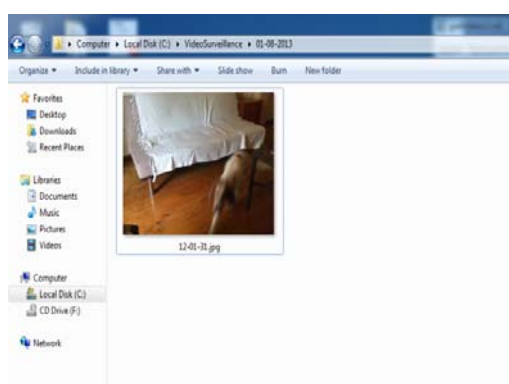


Εικόνα 20. Παράθυρο FG σε πείραμα εσωτερικού χώρου

Έτσι το σύστημα ορθά αναγνωρίζει την κίνηση στον χώρο που παρακολουθεί και προβαίνει στις ενέργειες αποθήκευσης εικόνας και αποστολής αυτής μέσω μηνύματος ηλεκτρονικού ταχυδρομείου, όπως φαίνεται στις εικόνες 21 και 22 αντίστοιχα.



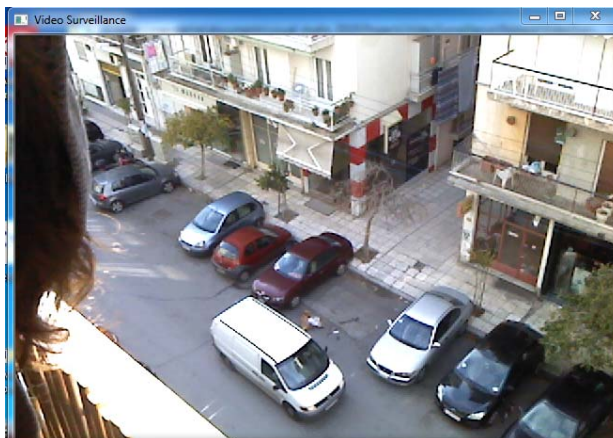
Εικόνα 21. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα εσωτερικού χώρου



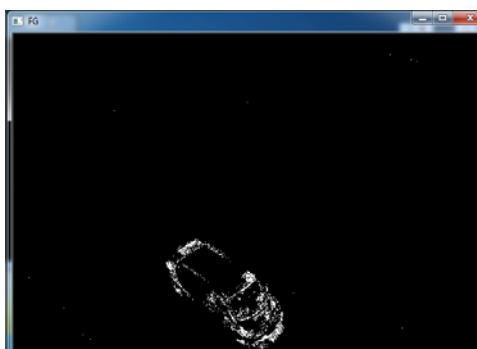
Εικόνα 22. Εικόνα που αποθηκεύθηκε σε πείραμα εσωτερικού χώρου

Στις περιπτώσεις των εσωτερικών χώρων το λογισμικό αντιλαμβάνεται άρτια την κίνηση και δεν είναι απαραίτητη η παραμετροποίηση του συστήματος. Η παραμετροποίηση θα σήμαινε την εισαγωγή μικρότερων τιμών στην μεταβλητή που περιέχει το ποσοστό ανοχής του συστήματος. Η όποια παραμετροποίηση θα ήταν απαραίτητη για να γίνει πιο “ευαίσθητο” το λογισμικό για να αντιληφτεί την κίνηση μικρότερων αντικειμένων. Ο λόγος που συμβαίνει αυτό σε έναν εσωτερικό χώρο οφείλεται στις περιορισμένες διαστάσεις του, όπως για παράδειγμα ένα τυπικό δωμάτιο οικίας. Συνεπώς τα κινούμενα αντικείμενα θα απεικονίζονται στο παράθυρο VideoSurveillance και στο παράθυρο FG με αρκετά εικονοστοιχεία για να γίνει λήψη απόφασης αν υπάρχει κίνηση στο χώρο ή όχι. Έτσι, όταν εισέρχεται ένα αντικείμενο στο πλάνο ο αλγόριθμος δίνει στο σύστημα πολλά στοιχεία για τον σχηματισμό του προσκηνίου και να το χαρακτηρίσει ως κινούμενο ή όχι.

3.4.2. Εξωτερικός Χώρος



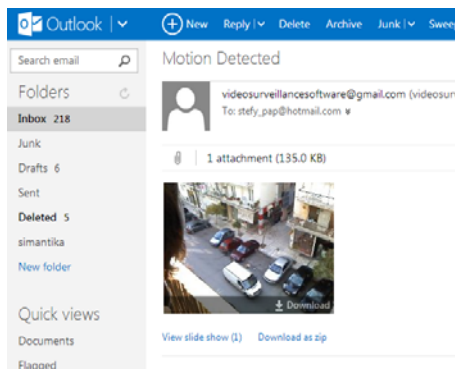
Εικόνα 23. Παράθυρο VideoSurveillance σε πείραμα εξωτερικού χώρου



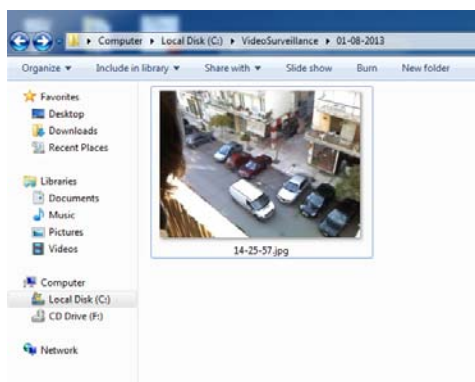
Εικόνα 24. Παράθυρο FG σε πείραμα εξωτερικού χώρου

Στην περίπτωση του εξωτερικού χώρου παρατηρούμε πως το λογισμικό λειτουργεί και πάλι σωστά, αναγνωρίζοντας την ύπαρξη κινούμενων αντικειμένων. Χαρακτηριστικά στο παράθυρο VideoSurveillance στην εικόνα 23 παρατηρούμε την εικόνα ενός αυτοκινήτου σε έναν δρόμο και στο παράθυρο FG της εικόνας 24 την εικόνα που παράγει ο αλγόριθμος. Για την επιτυχή ανίχνευση των αντικειμένων έγινε παραμετροποίηση των δεδομένων που χρησιμοποιεί το πρόγραμμα. Συγκεκριμένα η μεταβλητή που αντιπροσωπεύει το ποσοστό των λευκών εικονοστοιχείων επί του συνόλου αυτών μειώθηκε από 1.5% σε 1.0%. Αυτό συνέβη επειδή η λήψη η οποία πραγματοποιήθηκε σε εξωτερικό χώρο κατέγραφε αντικείμενα από μεγάλη απόσταση. Συνεπώς τα αντικείμενα που εισέρχονταν στο πλάνο είχαν μικρές διαστάσεις λόγω απόστασης και άρα απεικονίζονταν με λίγα εικονοστοιχεία. Στη συνέχεια το λογισμικό προβαίνει στις ενέργειες αποθήκευσης της εικόνας που προβάλλει το παράθυρο VideoSurveillance και αποστολής αυτής μέσω μηνύματος ηλεκτρονικού

ταχυδρομείου, όπως φαίνεται στις εικόνες 25 και 26. Συνεπώς η διαδικασία ακολουθείται με τον τρόπο που επιθυμούμε.



Εικόνα 25. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα εξωτερικού χώρου



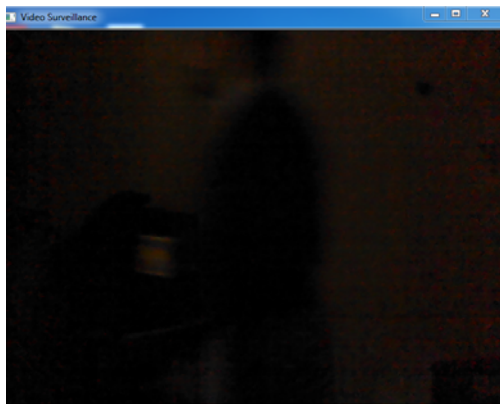
Εικόνα 26. Εικόνα που αποθηκεύθηκε σε πείραμα εξωτερικού χώρου

Αντιπαραθέτοντας τις εικόνες 19 και 23, που λήφθηκαν κατά την πειραματική διαδικασία σε εσωτερικό και εξωτερικό χώρο αντίστοιχα, παρατηρούμε πως δεν παίζουν ρόλο οι πραγματικές διαστάσεις των αντικειμένων, αλλά το σύνολο των εικονοστοιχείων που θα χρησιμοποιήσει το λογισμικό για την απεικόνιση αυτών.

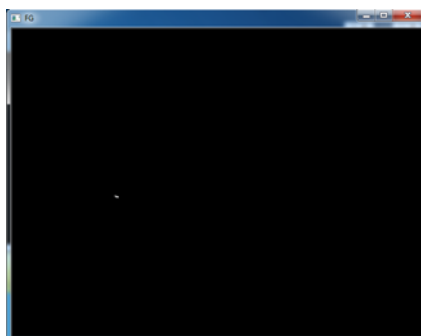
3.4.3. Συνθήκες πολύ χαμηλού φωτισμού

Στην περίπτωση χαμηλού φωτισμού τα πράγματα περιπλέκονται καθώς το λογισμικό εκλαμβάνει πολύ σκοτεινές εικόνες. Αυτό συνεπάγεται πως το αντικείμενο που θα εισχωρήσει στο πλάνο όπως φαίνεται στο παράθυρο VideoSurveillance της εικόνας 27 θα είναι επίσης σκοτεινό με αποτέλεσμα να μην μπορεί να φανεί η διαφορά μεταξύ Background και Foreground. Το λογισμικό δεν εκλαμβάνει την αλλαγή στο προσκήνιο, όπως φαίνεται στο παράθυρο FG της εικόνας 28 και συνεπώς την ύπαρξη κίνησης στο πλάνο. Το αποτέλεσμα είναι το λογισμικό να παραμένει α-

δρανές και λανθασμένα να μην προβαίνει στις λειτουργίες που εκτελεί όταν ανιχνευθεί κίνηση.



Εικόνα 27. Παράθυρο VideoSurveillance σε πείραμα πολύ χαμηλού φωτισμού



Εικόνα 28. Παράθυρο FG σε πείραμα πολύ χαμηλού φωτισμού

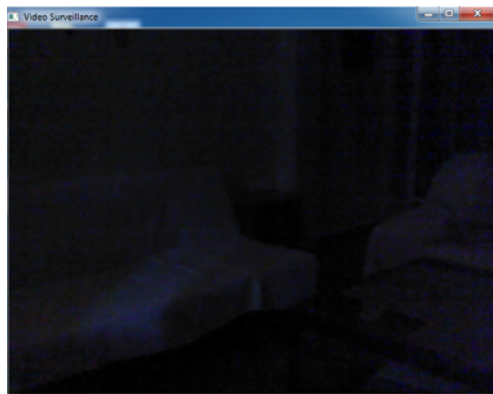
Αξίζει να σημειωθεί πως ενώ παραμετροποιήθηκε η ευαισθησία του προγράμματος, ώστε να χρειάζονται λιγότερα λευκά εικονοστοιχεία για την ανίχνευση κίνησης, το σύστημα δεν κατάφερε να ανταποκριθεί ορθά. Όπως φαίνεται στο παράθυρο FG της εικόνας 28, τα λευκά εικονοστοιχεία που προσμετρούνται και αποτελούν κριτήριο για την ανίχνευση της κίνησης είναι ελάχιστα. Ένα τέτοιο πρόβλημα ίσως θα μπορούσε να βρει λύση με κάμερες νυχτερινής όρασης.

3.4.4. Απότομη Εναλλαγή Φωτός

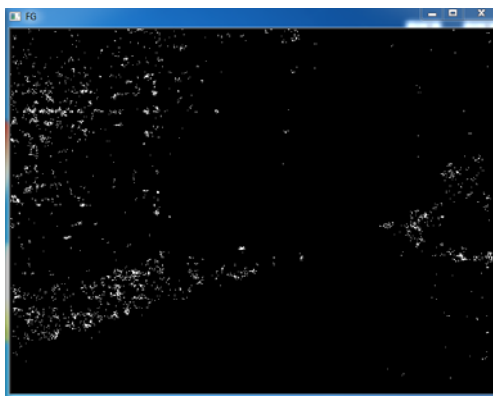
3.4.4.1. Απότομη Μείωση Φωτισμού

Στην περίπτωση της απότομης μείωσης φωτισμού το λογισμικό αντιδρά προβληματικά. Ενώ είναι φανερό στο παράθυρο VideoSurveillance της εικόνας 29 πως δεν υπάρχει αντικείμενο που να εισέρχεται στο προσκήνιο, ο αλγόριθμος εκλαμβάνει την απότομη μείωση του φωτός ως αλλαγή. Αυτό συμβαίνει γιατί τα εικονοστοιχεία από έγχρωμα καταλήγουν απότο-

μα να γίνονται μαύρα και οι δυο αυτές καταστάσεις έχουν μεγάλη διαφορά.

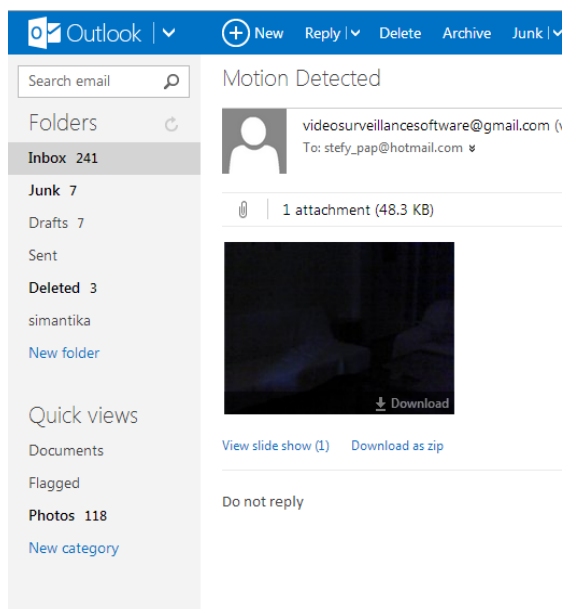


Εικόνα 29. Παράθυρο VideoSurveillance σε πείραμα με απότομη μείωση φωτισμού

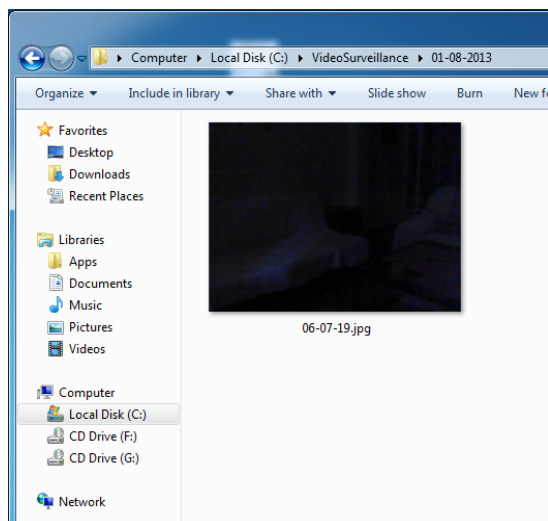


Εικόνα 30. Παράθυρο FG σε πείραμα με απότομη μείωση φωτισμού

Έτσι όπως φαίνεται και στο παράθυρο FG της εικόνας 30, μεγάλο ποσοστό των εικονοστοιχείων γίνονται λευκά σύμφωνα με το πρότυπο του αλγόριθμου. Αυτό έχει ως αποτέλεσμα τη λανθασμένη ενεργοποίηση λειτουργιών αποθήκευσης και αποστολής μηνύματος ηλεκτρονικού ταχυδρομείου, όπως φαίνεται στις εικόνες 31 και 32.



Εικόνα 31. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με απότομη μείωση φωτισμού

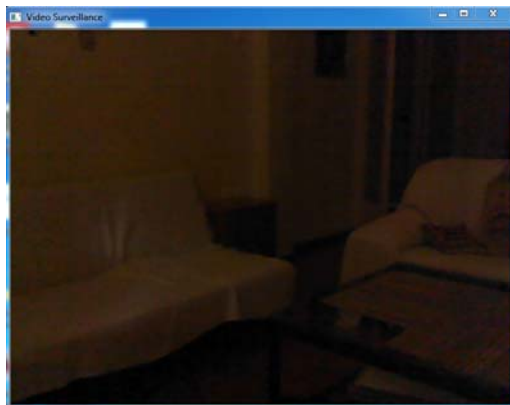


Εικόνα 32. Αποθήκευση εικόνας σε πείραμα με απότομη μείωση φωτισμού

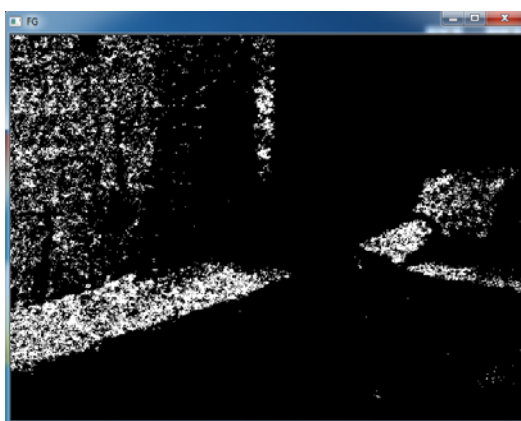
3.4.4.2. Απότομη αύξηση φωτισμού

Στην περίπτωση της απότομης αύξησης της φωτεινότητας της εικόνας υπάρχει ακριβώς το ίδιο πρόβλημα που υπάρχει στην απότομη μείωση της φωτεινότητας του πλάνου. Δηλαδή ο αλγόριθμος αντιλαμβάνεται ως κίνηση την αλλαγή χρώματος των εικονοστοιχείων. Είναι φανερό από το παράθυρο VideoSurveillance που απεικονίζεται στην εικόνα 3 ότι δεν

υπάρχει είσοδος αντικειμένου στο πλάνο. Παρόλα αυτά το παράθυρο FG της εικόνας 34 είναι γεμάτο λευκά εικονοστοιχεία.

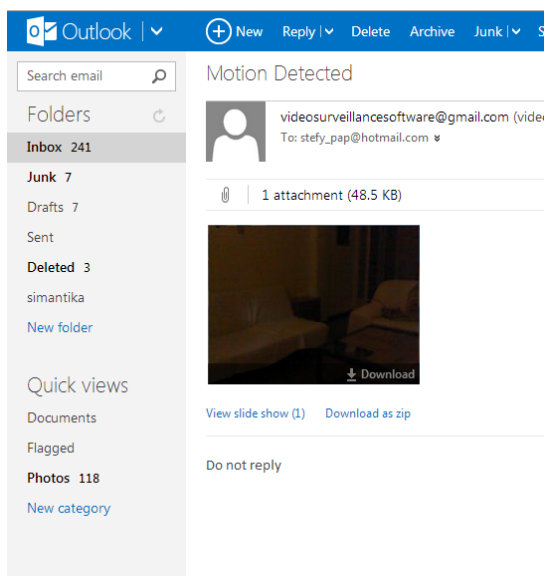


Εικόνα 33. Παράθυρο VideoSurveillance σε πείραμα με απότομη αύξηση φωτισμού

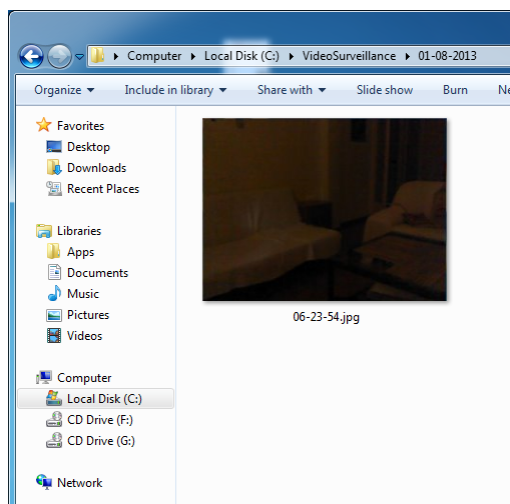


Εικόνα 34. Παράθυρο FG σε πείραμα με απότομη αύξηση φωτισμού

Αυτό έχει ως αποτέλεσμα την ενεργοποίηση των λειτουργιών αποθήκευσης της εικόνας του παραθύρου VideoSurveillance και την αποστολή αυτής μέσω ηλεκτρονικού ταχυδρομείου, όπως φαίνεται στις εικόνες 35 και 36 αντίστοιχα.



Εικόνα 35. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με απότομη αύξηση φωτισμού



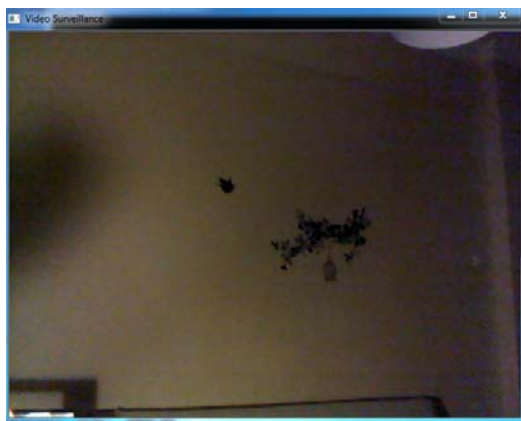
Εικόνα 36. Αποθήκευση εικόνας σε πείραμα με απότομη αύξηση φωτισμού

3.4.5. Περιπτώσεις σκιάς

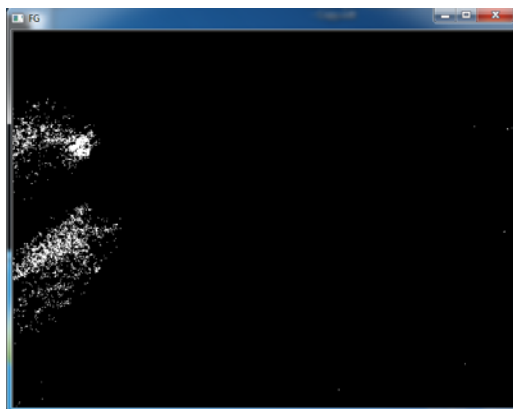
Στην περίπτωση της εισόδου σκιάς στο πλάνο τα πράγματα είναι οριακά. Ουσιαστικά υπάρχουν οι παράμετροι της έντασης και έκτασης της σκιάς. Παρακάτω δίνονται πειραματικά αποτελέσματα μιας περίπτωσης που το λογισμικό ανταποκρίθηκε ορθά και μιας άλλης περίπτωσης που ανταποκρίθηκε λανθασμένα.

3.4.5.1. Πρώτη περίπτωση εισόδου σκιάς

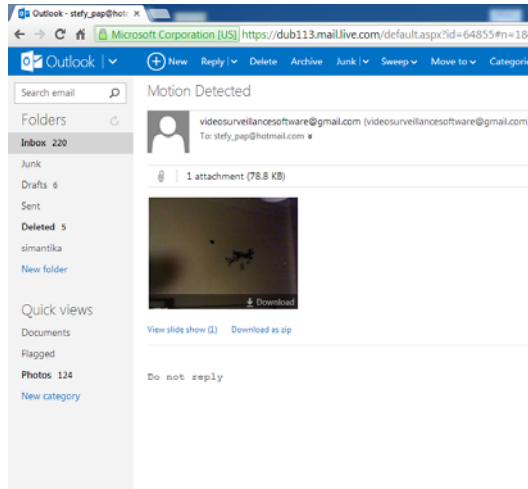
Στην πρώτη περίπτωση, που η σκιά είναι πιο έντονη, το λογισμικό αναγνωρίζει κίνηση στο πλάνο. Ουσιαστικά, τα εικονοστοιχεία που αλλάζουν χρώμα λόγω της σκιάς που εισέρχεται στο πλάνο αντιλαμβάνονται από τον αλγόριθμο ως νέο αντικείμενο στο προσκήνιο, όπως είναι φανερό από το παράθυρο FG της εικόνας 38. Αυτό έχει ως αποτέλεσμα να διεκπεραιώνονται οι λειτουργίες της αποθήκευσης εικόνας και αποστολής αυτής μέσω μηνύματος ηλεκτρονικού ταχυδρομείου. Τα στιγμιότυπα των ενεργειών αυτών φαίνονται στις εικόνες 39 και 40.



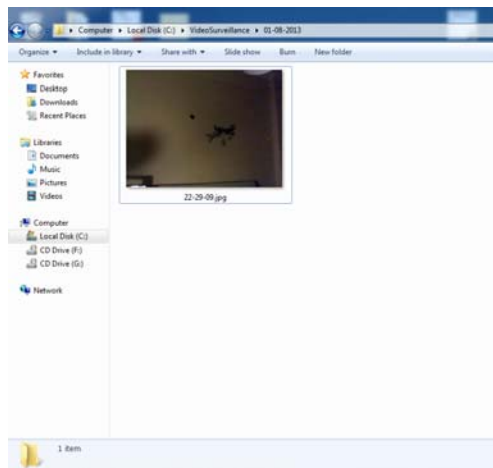
Εικόνα 37. Παράθυρο VideoSurveillance σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α)



Εικόνα 38. Παράθυρο FG σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α)



Εικόνα 39. Μήνυμα ηλεκτρονικού ταχυδρομείου σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α)

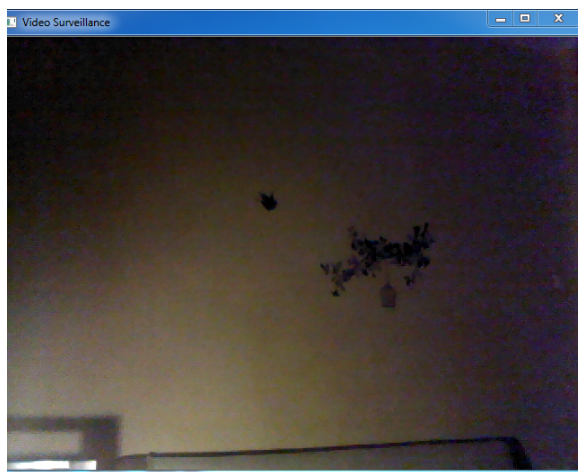


Εικόνα 40. Αποθήκευση εικόνας σε πείραμα με είσοδο σκιάς στο πλάνο – περίπτωση (α)

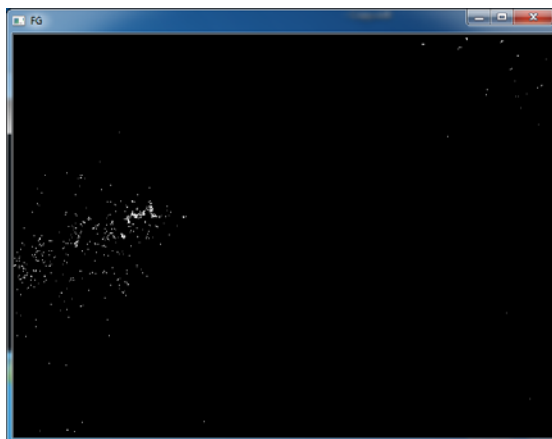
2 Αποθήκευση εικόνας σε πείραμα με είσοδο σκιάς στο πλάνο

3.4.5.2. Δεύτερη Περίπτωση Εισόδου Σκιάς

Στη δεύτερη πειραματική διαδικασία η σκιά που εισέρχεται στο πλάνο είναι λιγότερο έντονη από την πρώτη. Για το λόγο αυτό το λογισμικό ανταποκρίνεται ορθά και δεν αναγνωρίζει κίνηση, όπως φαίνεται και στην εικόνα 41. Ο λόγος που δεν ανιχνεύει κίνηση στο πλάνο είναι επειδή η χρωματική αλλαγή των εικονοστοιχείων με την είσοδο της σκιάς στο πλάνο είναι ανεπαίσθητη, οπότε δεν μεταφράζονται ως λευκά εικονοστοιχεία στο παράθυρο FG που φαίνεται στην εικόνα 42.



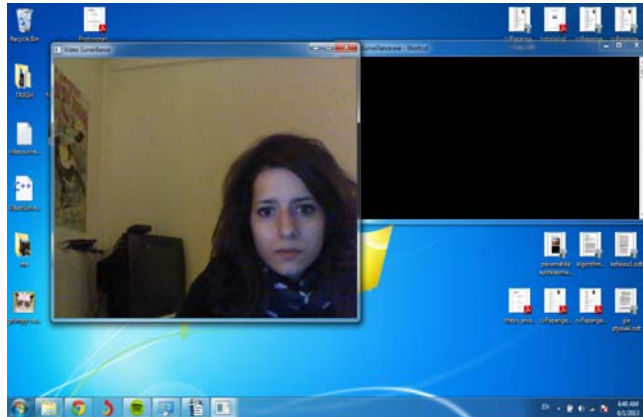
Εικόνα 41. Παράθυρο VideoSurveillance σε πείραμα με είσοδο σκιάς στο πλάνο- περιήπτωση (β)



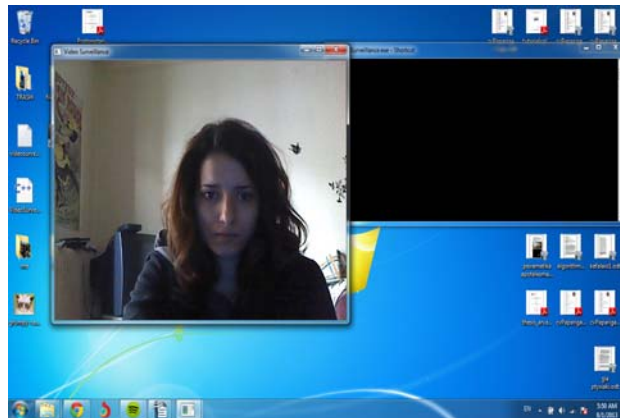
Εικόνα 42. Παράθυρο FG σε πείραμα με είσοδο σκιάς στο πλάνο- περιήπτωση (β)

3.4.6. Λειτουργία λογισμικού επί 12 συνεχόμενες ώρες

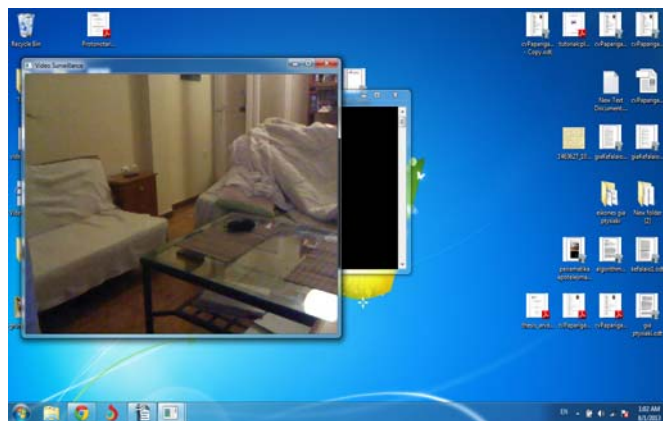
Κατά την πειραματική διαδικασία κρίθηκε απαραίτητη η δοκιμή του λογισμικού για 12 συνεχόμενες ώρες. Η δοκιμή αυτή πραγματοποιήθηκε δυο φορές. Μέσα στο διάστημα αυτό και τις δυο φορές το πρόγραμμα ανταποκρίθηκε πλήρως. Στις εικόνες 43 και 45 φαίνονται τα στιγμιότυπα των καταστάσεων του λογισμικού κατά την έναρξη της πειραματικής διαδικασίας και στις εικόνες 44 και 46 φαίνονται τα στιγμιότυπα κατά τη λήξη αυτών.



Εικόνα 43. Έναρξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (α)



Εικόνα 44. Λήξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (α)



Εικόνα 45. Έναρξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (β)

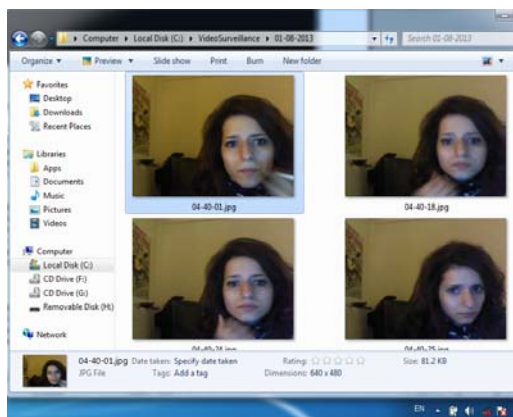


Εικόνα 46. Λήξη πειράματος δωδεκάωρης συνεχόμενης λειτουργίας (β)

Όπως φαίνεται και από τα στιγμιότυπα που απεικονίζονται παραπάνω, το λογισμικό ανταποκρίθηκε στη συνεχόμενη λειτουργία των δώδεκα ωρών χωρίς να παρουσιάσει προβλήματα κατά το διάστημα αυτό. Αξίζει να σημειωθεί πως η εναλλαγή από ημέρα σε νύχτα και το αντίστροφο, δηλαδή από την παρουσία σε απουσία φωτισμού, δεν ενεργοποίησε τους μηχανισμούς που πραγματοποιούνται κατά την ανίχνευση κίνησης. Αυτό συμβαίνει γιατί οι εναλλαγές αυτές γίνονται σταδιακά και όχι απότομα όπως στις πειραματικές διαδικασίες 3.4.4.1 και 3.4.4.2.

3.4.7. Λειτουργία λογισμικού χωρίς σύνδεση στο διαδίκτυο

Η λειτουργία της αποστολής μηνύματος ηλεκτρονικού ταχυδρομείου είναι μια από τις κυριότερες λειτουργίες που προσφέρει το σύστημα καθώς είναι πολύ χρήσιμο για την ενημέρωση του χρήστη του λογισμικού. Όταν όμως δεν υπάρχει σύνδεση στο διαδίκτυο κάτι τέτοιο είναι αδύνατο και η λειτουργία αυτή δεν μπορεί να πραγματοποιηθεί.

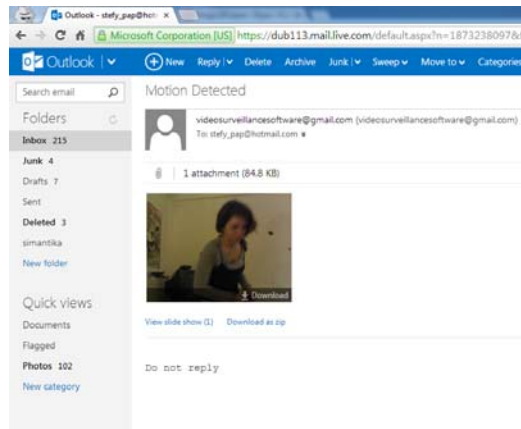


Εικόνα 47. Εικόνες που αποθηκεύθηκαν

Παρόλα αυτά το λογισμικό συνεχίζει να εκτελεί κανονικά τη λειτουργία της αποθήκευσης εικόνων με την προϋπόθεση ότι ανιχνεύεται κίνηση. Γενικά η κατάσταση κατά την οποία ο υπολογιστής δεν είναι συνδεδεμένος στο διαδίκτυο δεν προκαλεί πρόβλημα στις υπόλοιπες λειτουργίες του λογισμικού. Στην εικόνα 47 φαίνεται ο κατάλογος των εικόνων που αποθηκεύθηκαν ενώ ο υπολογιστής δεν ήταν συνδεδεμένος στο διαδίκτυο.

3.4.8. Λειτουργία λογισμικού με τον Η/Υ σε αδράνεια

Για τη δοκιμή του λογισμικού, επίσης, κρίθηκε απαραίτητη η δοκιμασία του σε κατάσταση αδράνειας του ηλεκτρονικού υπολογιστή. Για να πραγματοποιηθεί η πειραματική διαδικασία εκτελέστηκε το πρόγραμμα ενώ είχε οριστεί χρόνος ενός λεπτού για τη μετάβαση του υπολογιστή σε κατάσταση της αδράνειας. Το λογισμικό ανταποκρίθηκε πλήρως. Στην εικόνα 48 φαίνεται το μήνυμα ηλεκτρονικού ταχυδρομείου που λήφθηκε ενώ ο ηλεκτρονικός υπολογιστής ήταν σε αδράνεια.



Εικόνα 48. Μήνυμα ηλεκτρονικού ταχυδρομείου από το λογισμικό ενώ ο Η/Υ βρίσκεται σε αδράνεια

3.5. Συμπεράσματα

Τα αποτελέσματα της συγκεκριμένης εφαρμογής εξαρτώνται σε μεγάλο βαθμό από το ποσοστό επιτυχίας της διαδικασίας ανίχνευσης. Η διαδικασία ανίχνευσης δοκιμάστηκε στις ακόλουθες συνθήκες :

- σε εσωτερικό χώρο,
- σε εξωτερικό χώρο,
- σε συνθήκες χαμηλού φωτισμού,
- σε αυξομειώσεις της έντασης του φωτός
- σε είσοδο σκιάς στο πλάνο,

- σε 12ωρη συνεχόμενη λειτουργία του λογισμικού,
- σε κατάσταση χωρίς σύνδεση στο διαδίκτυο και
- με τον υπολογιστή σε κατάσταση αδράνειας.

Στον εσωτερικό χώρο η εφαρμογή λειτούργησε άρτια και αναγνώρισε την κίνηση σωστά με τον προκαθορισμένο τρόπο. Μπορούμε να πούμε ότι οι ιδανικότερες συνθήκες για να λειτουργήσει το παρόν λογισμικό είναι αυτές που εφαρμόστηκαν κατά την πρώτη πειραματική διαδικασία, δηλαδή μέσα σε ένα τυπικό δωμάτιο οικίας, χωρίς έντονο μεταβαλλόμενο φόντο. Γίνεται αντιληπτό ότι με αυτή τη διαπίστωση προσεγγίζουμε τον στόχο της παρούσας εργασίας, δηλαδή την παροχή ενός συστήματος ανίχνευσης κίνησης σε έναν οικιακό χρηστή, που κατά μεγάλο ποσοστό βρίσκεται και λειτουργεί σε ένα περιβάλλον όπως αυτό της πρώτης πειραματικής διαδικασίας.

Στον εξωτερικό χώρο η ανίχνευση κίνησης πραγματοποιήθηκε, όμως χρειάστηκε να γίνει παραμετροποίηση του συστήματος, ώστε να γίνουν αντιληπτά τα κινούμενα αντικείμενα που βρίσκονταν μακριά από το σημείο λήψης. Λόγω της απόστασης τα αντικείμενα δεν εμφανίζονταν στο προσκήνιο με τις πραγματικές τους διαστάσεις οπότε χρειάστηκε να μειωθεί το ποσοστό από 1,5% σε 1,0% ώστε να γίνουν αντιληπτά από το λογισμικό. Αυτό δείχνει την ευελιξία του λογισμικού που μπορεί να προσαρμοστεί ανάλογα με τις συνθήκες και τη χρήση που θέλουμε εμείς να του ορίσουμε είτε είναι παρατήρηση σε κλειστό οικιακό χώρο είτε σε έναν εξωτερικό με ανοιχτό ορίζοντα.

Ένας από τους πιο σημαντικούς παράγοντες που επηρεάζουν τη λειτουργία του λογισμικού είναι οι επιδράσεις του φωτός. Όπως είδαμε στις περιπτώσεις με πολύ χαμηλό φωτισμό το λογισμικό δε λειτούργησε όπως θα έπρεπε. Λόγω της χαμηλής φωτεινότητας δε γίνεται δυνατή η παρατήρηση των διαφορών μεταξύ παρασκηνίου και προσκήνιου και τα εκλαμβάνει ως μια συνεχής και αδρανή εικόνα. Συνεπώς το λογισμικό δεν αντιλαμβάνεται τις κινήσεις που γίνονται στο προσκήνιο και λανθασμένα δεν προβαίνει στις ενέργειες που θα έπρεπε κατά την ανίχνευση κίνησης.

Παρεμφερές φαινόμενο παρατηρήθηκε και στις περιπτώσεις απότομης αλλαγής φωτισμού, δηλαδή την απότομη μετάβαση από συνθήκη με επαρκή φωτισμό στο σκοτάδι και αντίστροφα. Στις περιπτώσεις αυτές το λογισμικό λειτούργησε με μια ιδιαιτερότητα, ενώ δεν υπήρχε κάποιο κινούμενο αντικείμενο, ανίχνευσε κίνηση. Αυτό οφείλεται στην απότομη εναλλαγή μεταξύ συνθηκών φωτισμού την οποία το λογισμικό κατέγραψε ως «κίνηση». Επειδή υπήρξε μεγάλη αλλαγή μεταξύ παρασκηνίου και προσκήνιου σαν γενική εικόνα στην κάμερα ο αλγόριθμος εντοπισμού κίνησης ενεργοποιήθηκε από αυτήν την αλλαγή χαρακτηρίζοντας την ως «κίνηση». Αυτή η ιδιαιτερότητα φαινομενικά μπορεί να παρουσιαστεί ως

προβληματική \,όμως, έχει πρακτική αξία για τον χρήστη. Εφόσον το ζητούμενό μας είναι ένα λογισμικό που ασκεί βίντεοεπιτήρηση,κατά κύριο λόγο σε οικιακό χώρο, η αντίληψη και η ενημέρωση του χρήστη για την όποια αλλαγή σε αυτό το περιβάλλον είναι χρήσιμη. Για παράδειγμα, κάποιος μπορεί να εισέρθει στον χώρο που θέλουμε να επιτηρήσουμε, αλλά υπάρχει περίπτωση να μην κινηθεί στα όρια λήψης της κάμερας,όμως, εάν ανάψει τα φωτά στο δωμάτιο ο χρήστης θα ειδοποιηθεί για αυτήν την αλλαγή στο χώρο από την εφαρμογή μας. Αυτή η πρακτική εφαρμογή ουσιαστικά σημαίνει ότι ενώ το λογισμικό δεν μπορεί να καταγράψει το κινούμενο αντικείμενο, μπορεί να αντιληφθεί την ανθρώπινη παρουσία.

Κλείνοντας όσον αφορά την επίδραση του φωτός πρέπει να αναφερθούν οι επιδράσεις της σκίασης. Στην περίπτωση της εισόδου σκιάς στο πλάνο,η λειτουργία σαφώς επηρεάζεται από τις παραμέτρους της έντασης και της έκτασης της σκιάς. Σε δύο πειραματικές διαδικασίες είχαμε διαφορετικά αποτελέσματα άμεσα εξαρτώμενα από τις παραμέτρους που αναφέραμε πιο πριν. Στην πρώτη περίπτωση, όπου η σκιά ήταν πιο έντονη, είχαμε λανθασμένη ανίχνευση κίνησης επειδή η σκιασμένη επιφάνεια είχε μεγάλη διαφορά στην απόχρωση με το προηγούμενο πλάνο. Έτσι, τα εικονοστοιχεία άλλαξαν χρώμα και χαρακτηρίστηκαν ως κινούμενο αντικείμενο. Στη δεύτερη περίπτωση, όπου η σκίαση ήταν πιο ομαλή και σταδιακή, το λογισμικό ανταποκρίθηκε σωστά και δεν ανίχνευσε κίνηση. Ο λόγος που δεν ανιχνεύει κίνηση στο πλάνο είναι επειδή η χρωματική αλλαγή των εικονοστοιχείων με τη σταδιακή είσοδο της σκιάς στο πλάνο είναι ανεπαίσθητη, οπότε δεν μεταφράζονται ως λευκά εικονοστοιχεία στο παράθυρο FG.

Παράλληλα με την ανίχνευση κίνησης, η εφαρμογή έχει και τη λειτουργία της ενημέρωσης του χρήστη μέσω ηλεκτρονικού ταχυδρομείου. Αυτή η λειτουργία πραγματοποιήθηκε άρτια σε όλες τις προαναφερθείσες περιπτώσεις όταν υπήρξε ανίχνευση κίνησης. Για να επεκτείνουμε το φάσμα της λειτουργίας της εφαρμογής κρίθηκε απαραίτητο η εκτέλεση και άλλων πειραματικών διαδικασιών για να ελέγχουν τα τεχνικά στοιχεία του όλου προγράμματος.

Έτσι, στην περίπτωση της δόκιμης της συνεχόμενης λειτουργίας του προγράμματος για 12 ώρες αυτό ανταποκρίθηκε σε 2 αντίστοιχες περιπτώσεις άρτια λειτουργώντας απρόσκοπτα και χωρίς αναφορά κάποιου προβλήματος. Σε μια παρόμοια πειραματική διαδικασία δοκιμάσαμε τη λειτουργία της εφαρμογής όταν ο H/Y βρισκόταν σε αδράνεια. Τα αποτελέσματα ήταν εφάμιλλα με αυτά της συνεχούς λειτουργίας,δεν παρατηρήθηκε κάποιο πρόβλημα άξιο αναφοράς και η λειτουργία ήταν ομαλή σε όλους τους τομείς της εφαρμογής.

Τέλος, στην περίπτωση λειτουργίας του λογισμικού χωρίς σύνδεση στο διαδίκτυο,όπως είναι φυσικό, δεν υπήρχε η δυνατότητα αποστολής των

φωτογραφιών μέσω ηλεκτρονικού ταχυδρομείου. Πέραν αυτού, η λειτουργία ήταν σωστή και η εφαρμογή εξακολουθεί την αποθήκευση εικόνων όταν ανιχνεύσει κίνηση.

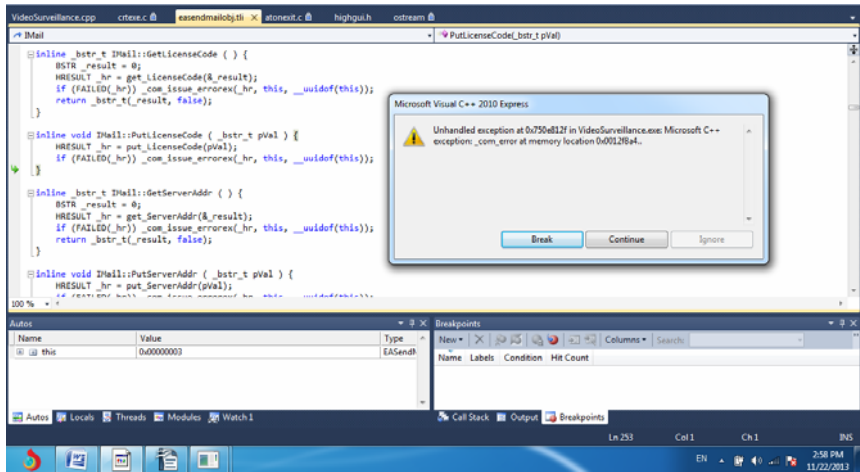
Εν κατακλείδι, η εφαρμογή λειτούργησε σωστά στις περισσότερες των περιπτώσεων ανταποκρινόμενη στις παραμέτρους που είχαμε ορίσει. Σαφώς, τα όποια προβλήματα ή ανεπάρκειες του λογισμικού χρειάζονται περαιτέρω ανάλυση και έρευνα για να ομαλοποιηθούν και να καταστεί η εφαρμογή πλήρως λειτουργική και εύχρηστη.

3.6. Κυριότερα προβλήματα που παρατηρήθηκαν

Κατά την εκτέλεση του λογισμικού της παρούσας εργασίας παρατηρήθηκαν κάποια προβλήματα που σχετίζονται κυρίως με τη βιβλιοθήκη EASendMail.

Το κυριότερο πρόβλημα είναι πως η διαδικασία αποστολής μηνύματος ηλεκτρονικού ταχυδρομείου είναι υπολογιστικά πολύπλοκη και για το λόγο αυτό το λογισμικό αργεί να ανταποκριθεί όταν χρειαστεί να εκτελέσει τη λειτουργία αυτή. Χαρακτηριστικά αναφέρεται πως ο χρόνος που μεσολαβεί από την έναρξη της λειτουργίας της αποστολής, μέχρι την ολοκλήρωσή της μεσολαβούν τουλάχιστον 5 δευτερόλεπτα. Κάτι τέτοιο αποτελεί πρόβλημα καθώς παρεμποδίζει τη διαδικασία αποθήκευσης εικόνων που συμπίπτουν χρονικά με το διάστημα που χρειάζεται το λογισμικό για την αποστολή της εικόνας μέσω ηλεκτρονικού ταχυδρομείου στον χρήστη. Ένα τέτοιο πρόβλημα θα μπορούσε να αντιμετωπιστεί με τη χρήση μιας άλλης βιβλιοθήκης αυτόματης αποστολής ηλεκτρονικών μηνυμάτων, πέραν της EASendMail. Ίσως θα μπορούσε να αντιμετωπιστεί και με τη χρήση ενός υπολογιστή νεότερης τεχνολογίας που να έχει μεγαλύτερη επεξεργαστική ισχύ.

Ένα ακόμη πρόβλημα που αντιμετωπίζει το λογισμικό υπάγεται και πάλι στη βιβλιοθήκη EASendMail και πρόκειται για την άδεια χρήσης. Η βιβλιοθήκη αυτή δεν είναι ελεύθερη για χρήση και έτσι απαιτείται χρηματικό ποσό για την αγορά της. Για την ανάπτυξη της εργασίας ήταν αναγκαίο να ρυθμίζεται η ημερομηνία του συστήματος στις 1 Αυγούστου 2013 ώστε να μπορεί να εκτελεστεί η λειτουργία αποστολής μηνύματος ηλεκτρονικού ταχυδρομείου. Αυτή η ρύθμιση πρέπει να γίνεται γιατί η άδεια της βιβλιοθήκης την ημερομηνία αυτή δεν είχε λήξει και μπορούσε το λογισμικό να χρησιμοποιήσει τις συναρτήσεις της EASendMail. Όταν γίνεται η εκτέλεση του λογισμικού χωρίς τη ρύθμιση της κατάλληλης ημερομηνίας το πρόγραμμα αποτυγχάνει να ανταποκριθεί, όπως φαίνεται στην Εικόνα 49.



Εικόνα 49. Μήνυμα λάθους από το λογισμικό

3.6. Μελλοντικές επεκτάσεις του λογισμικού

Γενικότερα το λογισμικό που αναπτύχθηκε για τους σκοπούς της εργασίας είναι άρτιο ως προς τις λειτουργίες που προσφέρει. Παρόλα αυτά, θα μπορούσαν να προστεθούν σε αυτό διάφορες λειτουργίες και χαρακτηριστικά. Μερικά από αυτά είναι:

- Αποθήκευση σε μορφή βίντεο
- Αποστολή SMS για την ειδοποίηση χρήστη
- Απομακρυσμένη προσβασιμότητα μέσω διαδικτύου
- Πρόσβαση στο λογισμικό μέσω κινητού

Βιβλιογραφία

- [1] Αρχή Προστασίας Δεδομένων Προσωπικού Χαρακτήρα:
www.dpa.gr, 24/11/2013
- [2] History of Video Security Cameras:
<http://www.supercircuits.com/resources/blog/the-history-of-video-security-cameras> , 24/11/2013
- [3] Ανίχνευση Κίνησης - Βικιπαιδεία
http://el.wikipedia.org/wiki/%CE%91%CE%BD%CE%AF%CF%87%CE%BD%CE%B5%CF%85%CF%83%CE%B7_%CE%BA%CE%AF%CE%BD%CE%B7%CF%83%CE%B7%CF%82, 16/11/2013
- [4] iSpy: Open Source Camera Security Software:
<http://www.ispyconnect.com/>, 16/11/2013
- [5] Background Substraction:
http://en.wikipedia.org/wiki/Background_subtraction, 16/11/2013
- [6] Nan Lu, Jihong Wang, Q.H. Wu and Li Yang, “An Improved Motion Detection Method for Real-Time Surveillance”, IAENG International Journal of Computer Science, February 2008
- [7] Ζουρνής-Καρούζος Νικόλαος, «Τεχνικές ανίχνευσης κίνησης σε προκαθορισμένες περιοχές της εικόνας σε πραγματικό χρόνο», Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ιούνιος 2007
- [8] Microsoft Visual Studio Express:
http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Express, 16/11/2013
- [9] VC++ 2010 Express System Requirements
<http://www.cplusplus.com/forum/lounge/89682/>, 16/11/2013
- [10] Βιβλιοθήκη (υπολογιστές):,
[http://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%B2%CE%BB%CE%B9%CE%BF%CE%B8%CE%AE%CE%BA%CE%B7_\(%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AD%CF%82\)](http://el.wikipedia.org/wiki/%CE%92%CE%B9%CE%B2%CE%BB%CE%B9%CE%BF%CE%B8%CE%AE%CE%BA%CE%B7_(%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CE%AD%CF%82)), 8/11/2013

- [11] Gary Bradski and Adrian Kaehler, “Learning OpenCV: Computer Vision with the OpenCV Library”, O’Reilly Press, ISBN: 978-0596516134, 2008
- [12] Ιωάννης Πρωτονοτάριος, «Ανάπτυξη συστήματος ενσωματωμένων αισθητήρων για ασύρματη μετάδοση εικόνας και δεδομένων», Πανεπιστήμιο Πατρών, Φεβρουάριος 2011
- [13] SMTP Component: <http://www.emailarchitect.net/easendmail>, 16/11/2013
- [14] EASendMail SMTP Component: <http://www.softpedia.com/get/Programming/Components-Libraries/EASendMail-SMTP-Component-NET-Edition.shtml>, 16/11/2013
- [15] SMTP: <http://el.wikipedia.org/wiki/SMTP>, 16/11/2013
- [16] OpenCV HighGUI Reference Manual: http://www.cognotics.com/opencv/docs/1.0/ref/opencvref_highgui.htm, 10/08/2013
- [17] C++ OpenCV IplImage: <http://stackoverflow.com/questions/5192578/opencv-iplimage>, 10/08/2013
- [18] Reading and writing images and video: http://opencv.jp/opencv-2svn_org/c/highgui_reading_and_writing_images_and_video.html, 10/08/2013
- [19] Basic c structures and operations: [http://docs.opencv.org/modules/core/doc/old_basic_structures.html?highlight=cvcreateimage#IplImage* cvCreateImage\(CvSize size, int depth, int channels\)](http://docs.opencv.org/modules/core/doc/old_basic_structures.html?highlight=cvcreateimage#IplImage* cvCreateImage(CvSize size, int depth, int channels))], 10/08/2013
- [20] Θόρυβος: <http://el.wikipedia.org/wiki/%CE%98%CF%8C%CF%81%CF%85%CE%B2%CE%BF%CF%82>, 20/11/2013
- [21] RGB: <http://el.wikipedia.org/wiki/RGB>, 28/11/2013
- [22] Gaussian Model: <http://neverland.cauwe.org/fcauwe/mocap/files/cvGaussianBGModelDemo.cpp>, 24/11/2013
- [23] OpenCV : http://opencv.willowgarage.com/wiki/VisualC%2B%2B_VS2010, 02/07/2013
- [24] SMTP Component for VB, ASP, C++, Delphi - Using EASendMail ActiveX Object: http://www.emailarchitect.net/easendmail/sdk/html/object_usage_a.htm, 02/07/2013

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
#include "stdafx.h"
#include "cv.h"
#include "cvaux.h"
#include "highgui.h"
#include <ctype.h>
#include <stdio.h>
#include <iostream>
#include <io.h>
#include <windows.h>
#include <time.h>
#include <string.h>
#include <direct.h>
#include "easendmailobj.tlh"
#include <fstream>
#include <cstdio>

using namespace std;
using namespace EASendMailObjLib;

    struct settings {
        char save_path[255];
        int win_size;
        int n_gauss;
        float bg_threshold;
        float std_threshold;
        int minArea;
        float weight_init;
        int variance_init;
        int initialization_captures;
        float detection_threshold;
        int detection_value_threshold;
        int mail_delay;
        int save_delay;
        char mail_from[255];
        char mail_serveraddr[255];
        int mail_serverport;
        char mail_username[255];
        char mail_password[255];
        char mail_to[255];
    } typedef sets;
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
int main(int argc, char** argv)
{
    sets vars;
    FILE *fp;
    fp=fopen("settings.ini","r");
    fscanf(fp,"%s",&vars.save_path);
    fscanf(fp,"%d",&vars.win_size);
    fscanf(fp,"%d",&vars.n_gauss);
    fscanf(fp,"%f",&vars.bg_threshold);
    fscanf(fp,"%f",&vars.std_threshold);
    fscanf(fp,"%d",&vars.minArea);
    fscanf(fp,"%f",&vars.weight_init);
    fscanf(fp,"%d",&vars.variance_init);
    fscanf(fp,"%d",&vars.initialization_captures);
    fscanf(fp,"%f",&vars.detection_threshold);
    fscanf(fp,"%d",&vars.detection_value_threshold);
    fscanf(fp,"%d",&vars.mail_delay);
    fscanf(fp,"%d",&vars.save_delay);
    fscanf(fp,"%s",&vars.mail_from);
    fscanf(fp,"%s",&vars.mail_serveraddr);
    fscanf(fp,"%d",&vars.mail_serverport);
    fscanf(fp,"%s",&vars.mail_username);
    fscanf(fp,"%s",&vars.mail_password);
    fscanf(fp,"%s",&vars.mail_to);
    fclose(fp);
    time_t prevMailTime=99999999;
    time_t prevSaveTime=99999999;

    mkdir(vars.save_path);
    /*
cout<<"\n"<<vars.save_delay;
cout<<"\nvars.winsize "<<vars.win_size;
cout<<"\nvars.savepath "<<vars.save_path;
cout<<"\nvars.ngauss "<<vars.n_gauss;
cout<<"\nvars.bgthreshold "<<vars.bg_threshold;
cout<<"\nvars.stdthreshold "<<vars.std_threshold;
cout<<"\nvars.minarea "<<vars.minArea;
cout<<"\nvars.wight_init "<<vars.weight_init;
cout<<"\nvars.variance "<<vars.variance_init;
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
cout<<"\nvars.detection threshold
"<<vars.detection_threshold;
cout<<"\nvars.detection_value_threshold
"<<vars.detection_value_threshold;
cout<<"\nvars.mail_delay "<<vars.mail_delay;
cout<<"\nvars.save_delay "<<vars.save_delay;
cout<<"\nvars.mailfrom "<<vars.mail_from;
cout<<"\nvars.mailServerAddr "<<vars.mail_serveraddr;
cout<<"\nvars.serverport "<<vars.mail_serverport;
cout<<"\nvars.mailusername "<<vars.mail_username;
cout<<"\nvars.mailpassword "<<vars.mail_password;
cout<<"\nvars.mailto "<<vars.mail_to;*/

    CvCapture* capture=cvCaptureFromCAM(0);

if(!capture)
{
    fprintf(stderr,"Could not initialize...\n");
    return -1;
}

    IplImage* videoFrame = NULL;
videoFrame = cvQueryFrame(capture);

if(!videoFrame)
{
    printf("Bad frame \n");
    exit(0);
}

    cvNamedWindow("Video Surveillance",
CV_WINDOW_AUTOSIZE);
    //cvNamedWindow("BG", CV_WINDOW_AUTOSIZE);
    //cvNamedWindow("FG", CV_WINDOW_AUTOSIZE);

    // Select parameters for Gaussian model.
    CvGaussBGStatModelParams* params = new CvGaussBGStat-
ModelParams;
    params->win_size=vars.win_size;
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
params->n_gauss=vars.n_gauss;
params->bg_threshold=vars.bg_threshold;
params->std_threshold=vars.std_threshold;
params->minArea=vars.minArea;
params->weight_init=vars.weight_init;
params->variance_init=30;

CvBGStatModel* bgModel = cvCreateGaussianBGModel(videoFrame, params);

int key=-1;

int cols=bgModel->foreground->width;
int rows=bgModel->foreground->height;
float percent=(cols*rows)*vars.detection_threshold;
int frameCount=0;
/*for (int i=0; i<10; i++)
{
    cvShowImage("Video Surveillance", videoFrame);
    videoFrame=cvQueryFrame(capture);
    if (!videoFrame)
        break;
    key=cvWaitKey(2);

}*/
while(key != 'q')
{
    cvShowImage("Video Surveillance", videoFrame);

    //Grab a frame
    videoFrame = cvQueryFrame(capture);
    if(!videoFrame)
        break;

    // Update model
    cvUpdateBGStatModel(videoFrame, bgModel);
if(frameCount<10)
    frameCount++;
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
else
{
    cv::Mat imgMat(bgModel->foreground);
    int value,sum=0;

    for(int i=0;i<rows;i++)
    {
        for (int j=0;j<cols;j++)
        {

            value=imgMat.at<uchar>(i,j);
            if
            (value>=vars.detection_value_threshold)
            {
                sum++;
            }
        }
    }

    //cout<<"\nsum="<<sum;
    // cout<<"\npercent="<<percent;
    if (sum>=percent)
    {
        char date[11],tim[9];
        time_t now = time(NULL);
        strftime(date, 11, "%d-%m-%Y", local-
time(&now));

        string dateStr=vars.save_path;
        dateStr+=date;

        if((_access(dateStr.c_str(), 0)) == -1)
        {
            _mkdir(dateStr.c_str());
        }
        strftime(tim, 9, "%H-%M-%S", local-
time(&now));

        string timeStr=dateStr+"\\";
        timeStr+=tim;
        timeStr+=" .jpg";
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

```
double saveCheck=difftime(now,
prevSaveTime);

    if (saveCheck>vars.save_delay)
    {

        cvSaveImage(timeStr.c_str(),videoFrame,0);
        prevSaveTime=time(NULL);
    }

double spamCheck=difftime(now,
prevMailTime);

    if (spamCheck>vars.mail_delay)

    {cvSaveImage(timeStr.c_str(),videoFrame,0);
      ::CoInitialize(NULL);
      IMailPtr oSmtip = NULL;
      //set mail parameters

      oSmtip.CreateInstance("EASendMailObj.Mail");
      oSmtip->LicenseCode= _T("TryIt");
      oSmtip->FromAddr = (vars.mail_from);
      oSmtip->AddRecipientEx(vars.mail_to, 0);
      oSmtip->Subject=_T("Motion Detected");
      oSmtip->BodyText=_T("Do not reply");
      oSmtip->AddAttachment((timeStr.c_str()));
      oSmtip->ServerAddr =
(vars.mail_serveraddr);
      oSmtip->ServerPort =
vars.mail_serverport;
      oSmtip->UserName=vars.mail_username;
      oSmtip->Password=vars.mail_password;
      //establish SSL connection
      oSmtip->SSL_init();
      oSmtip->SendMail();
      oSmtip.Release();
      prevMailTime=time(NULL);
```

Παράρτημα Α – Πηγαίος Κώδικας Εφαρμογής

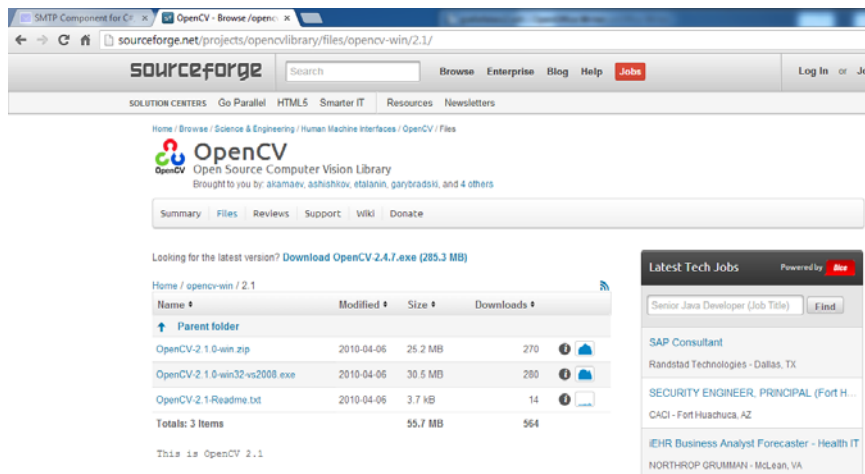
```
        }  
    }  
    key = cvWaitKey(2);  
sum=0;  
    }  
    }  
    cvReleaseBGStatModel(&bgModel);  
cvReleaseCapture(&capture);  
  
return 0;  
}
```

Παράρτημα Β– Οδηγίες σύνδεσης OpenCV με Visual Studio

Β.1 Σύνδεση Visual Studio-OpenCV

Για την εγκατάσταση και σύνδεση της OpenCV με το Microsoft Visual Studio 2010 Express Edition ακολουθούμε τα εξής βήματα:

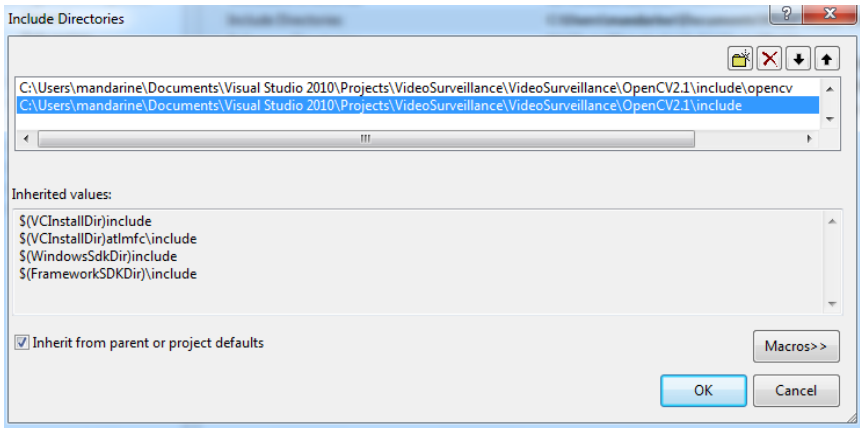
- 1) Download του εκτελέσιμου αρχείου από τη σελίδα <http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.1/>.



3 Σελίδα Sourceforge

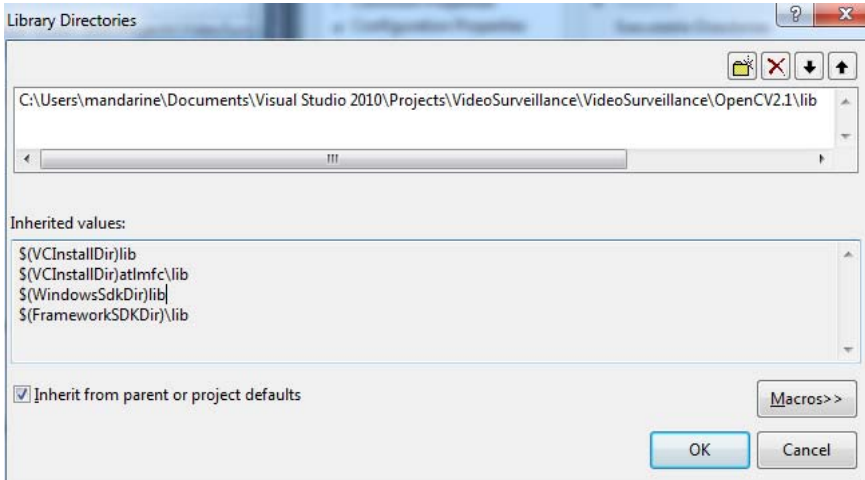
- 2) Download του Microsoft Visual C++ 2010 Express Edition από τη <http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express> σελίδα
- 3) Στις ιδιότητες του cpp αρχείου θα πρέπει να προστεθούν. Στον κατάλογο VC++ Directories->Include Files οι διαδρομές

OpenCV2.1\include\opencv
OpenCV2.1\include



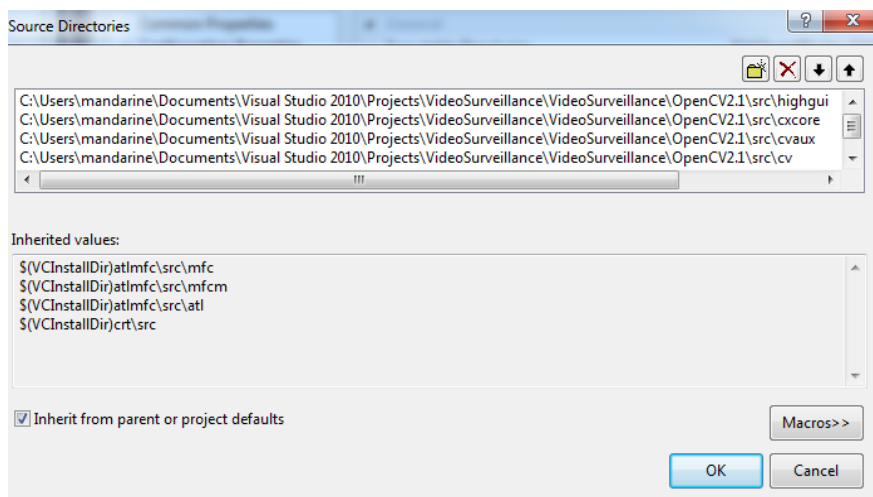
Εικόνα 50. Ιδιότητες Include του cpp αρχείου

- 4) Στον κατάλογο VC++ Directories->Library Directories τη διαδρομή C:\OpenCV2.1\lib



Εικόνα 51. Ιδιότητες Library του cpp αρχείου

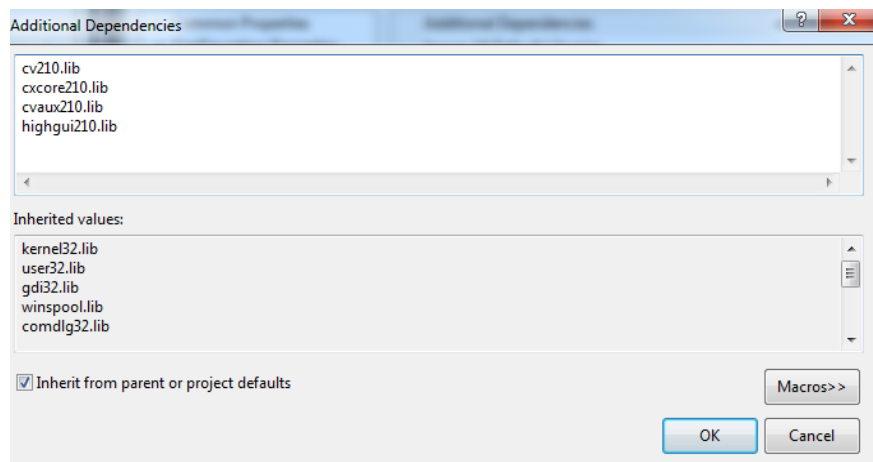
- 5) Στον κατάλογο VC++ Directories-> Source Files οι διαδρομές
- OpenCV2.1\src\cv
 - OpenCV2.1\src\cvaux
 - OpenCV2.1\src\cxcore
 - OpenCV2.1\src\highgui



Εικόνα 52. Ιδιότητες Source του cpp αρχείου

6) Στον κατάλογο Configuration Properties->Linker->Input->Additional Dependencies θα πρέπει να προστεθούν:

- cv210.lib
- cxcore210.lib
- cvaux210.lib
- highgui210.lib

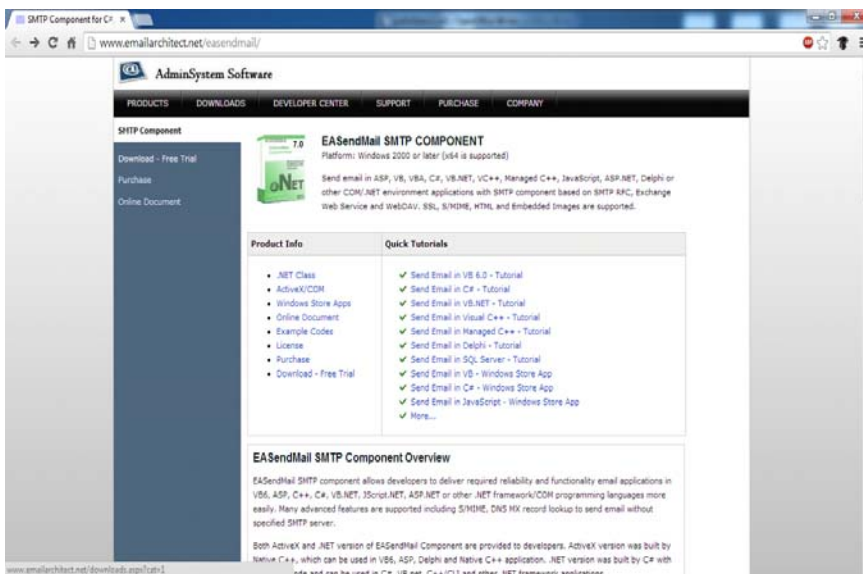


Εικόνα Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε..
Ιδιότητες Additional Dependencies του cpp αρχείου

Η εφαρμογή πλέον είναι έτοιμη να χρησιμοποιήσει τις συναρτήσεις της OpenCV [23].

B.2 Εγκατάσταση και σύνδεση EASendMail με το Microsoft Visual Studio

- 1) Download και εκτέλεση του εγκαταστάτη EASendMail από τη σελίδα www.emailarchitect.net.



Εικόνα 53. Η ιστοσελίδα www.emailarchitect.net

- 2) Αντιγραφή και επικόλληση των αρχείων στον φάκελο που αναπτύσσεται η εργασία
 - easendmailobj.tlh
 - easendmailobj.tli

Η εφαρμογή πλέον είναι έτοιμη να χρησιμοποιήσει τις συναρτήσεις της EASendMail και να εκτελέσει τις επιθυμητές λειτουργίες της [23].

[Αυτή η σελίδα είναι εσκεμμένα λευκή]

ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ ΚΑΙ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΙΝΗΣΗΣ ΣΕ
ΕΛΕΓΧΟΜΕΝΟ ΧΩΡΟ ΜΕ ΧΡΗΣΗ WEBCAMERA ΚΑΙ OPENCV

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Στεφάνια Παπαρήγα

Η καθημερινότητα ενός σύγχρονου ανθρώπου έχει αλλάξει ριζικά εξαιτίας της ραγδαίας ανάπτυξης της τεχνολογίας. Ο άνθρωπος, στις καθημερινές του δραστηριότητες, δρα υπό τον περιορισμό του χρόνου και του χώρου. Όσο βελτιώνεται η τεχνολογία αυτοί οι περιορισμοί εξαλείφονται. Έτσι μπορούμε να πραγματοποιήσουμε περισσότερες λειτουργίες και να καλύψουμε τις ανάγκες μας σε πολύ λιγότερο χρόνο καθώς και ο παράγοντας της απόστασης τείνει να γίνεται όλο και μικρότερο εμπόδιο. Μια από τις σημαντικότερες ανθρώπινες ανάγκες στη σύγχρονη αυτή εποχή είναι η αίσθηση της ασφάλειας. Η βασική αίσθηση της ασφάλειας ταυτίζεται με την αίσθηση της ασφάλειας του προσωπικού χώρου του κάθε ανθρώπου. Βασικό στοιχείο σε αυτό είναι ο έλεγχος του περιβάλλοντός του μέσω της παρακολούθησης του χώρου αυτού. Στην επιτήρηση χώρων, η οπτική επαφή είναι ένα σημαντικό μέσο αξιολόγησης του χώρου και της κατάστασης που επικρατεί σε αυτόν. Όταν όμως δεν υπάρχει άνθρωπος παρατηρητής στο χώρο υπό επίβλεψη τότε αυτή η επαφή χάνεται. Το κενό αυτό μπορούν σήμερα να καλύψουν συστήματα που βασίζονται σε απλές κάμερες και υπολογιστές. Αναγνωρίζοντας την ανάγκη του ανθρώπου για ασφάλεια και την ευκολία που παρέχεται πλέον από τους υπολογιστές για την επιτήρηση χώρων, στην παρούσα εργασία μελετήθηκαν τεχνικές επιτήρησης χώρου και αναπτύχθηκε εφαρμογή βασισμένη στη βιβλιοθήκη επεξεργασίας εικόνας Open CV, με στόχο την παρακολούθηση ελεγχόμενου χώρου με χρήση υπολογιστή και μιας απλής διαδικτυακής κάμερας (web camera).

MOTION DETECTION FOR SURVEILLANCE USING A WEBCAM AND
OPENCV

THESIS

Stefania Pappari

The daily life of modern humans has changed dramatically due to the rapid development of technology. Humans, in their daily activities are acting under the limitation of time and space. As technology improves, these restrictions are diminished or even eliminated. This has a significant impact on daily lives, as humans can now accomplish more functions and to meet their needs in less time. The distance factor tends to become a smaller and smaller obstacle. One of the most important human needs in this modern era is the sense of security. The basic sense of security coincides with the same sense of security of personal space of everyone. A key element in this is to control the environment through visual monitoring. In surveillance, visual contact is an important tool in assessing a site and a situation. In cases where there is no human observer on site under supervision then this visual contact is lost. This gap can now cover systems based on simple cameras and computers. Recognizing the human need for privacy and security and the convenience provided by computers on surveillance sites, in this thesis surveillance techniques have been studied and a video surveillance system has been developed and implemented based on the image processing library Open CV, to monitor a personal space using a computer and a simple web camera.