



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ. "ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΥΠΗΡΕΣΙΕΣ"
ΚΑΤΕΥΘΥΝΣΗ: ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ



ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΥΦΥΟΥΣ ΔΙΚΤΥΟΚΕΝΤΡΙΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΚΡΙΤΙΚΗΣ ΒΙΝΤΕΟΕΠΙΤΗΡΗΣΗΣ
ΚΑΤΑΣΤΑΣΗΣ ΑΣΘΕΝΟΥΣ ΚΑΙ ΑΝΙΧΝΕΥΣΗΣ ΣΥΝΑΓΕΡΜΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εισηγητής: Κοκκοβός Σίμος Α.Μ. : ΜΕ 11081

Επιβλέπων: Απόστολος Μηλιώνης - Επίκουρος Καθηγητής

Ακαδημαϊκό Έτος 2014-2015

*Μόνο μέσα από σκληρή δουλειά και
επώδυνη προσπάθεια, με ασίγαστη
ενεργητικότητα και αποφασιστικό θάρρος
μπορούμε να προχωρήσουμε σε καλύτερα πράγματα.*

Θεόδωρος Ρούζβελτ

Ευχαριστίες:

Ευχαριστώ θερμά τους ανθρώπους του στενού συγγενικού και επαγγελματικού μου κύκλου για την ηθική υποστήριξη και την υπομονή τους τόσο κατά την διάρκεια των σπουδών μου όσο και κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Ευχαριστώ τον επόπτη καθηγητή μου κ. Μηλιώνη Απόστολο για τις πολύ σημαντικές συμβουλές που μου έδωσε κατά την διάρκεια της προετοιμασίας της διπλωματικής εργασίας. Θα ήθελα επίσης, να ευχαριστήσω όλους τους διδάσκοντες καθηγητές μου, που μου έδωσαν τις γνώσεις τους και με μύησαν στα επιστημονικά μονοπάτια κατά την φοίτηση μου στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς.

Τέλος, ένα μεγάλο ευχαριστώ σε όλους όσους πήραν μέρος στην διπλωματική μου εργασία, γιατί χωρίς την πολύτιμη βοήθεια τους δεν θα μπορούσε να διεκπεραιωθεί το τελικό αποτέλεσμα.

ΕΙΣΑΓΩΓΗ

Παρά το γεγονός ότι οι σχετικές εξελίξεις συνεχίζονται με αμείωτη ένταση και με επιταχυνόμενους ρυθμούς, είναι πλέον κοινοτυπία να διατυπώσει κανείς σήμερα την άποψη ότι το κοινωνικό περιβάλλον και η πραγματικότητα βρίσκονται σε μια διαδικασία ραγδαίας και ριζικής διαφοροποίησης της βασικής δομής και των κυρίαρχων διαδικασιών. Είναι γνωστό ότι η διαφοροποίηση αυτή αποτελεί συνέπεια και αποτέλεσμα των καθημερινών εξελίξεων στον τομέα της πληροφορικής και των ψηφιακών συστημάτων, ούτως ώστε τα πράγματα να οδεύουν αδήριτα προς την συνολική επικράτηση νέων δραστηριοτήτων και τεχνολογιών.

Έτσι λοιπόν, ολοκληρώνοντας σε μεταπτυχιακό επίπεδο την πανεπιστημιακή μας εκπαίδευση καλούμαστε να γίνουμε επίσημα μέλη της ακαδημαϊκής κοινότητας, πραγματοποιώντας την πρώτη αληθινά μεγάλη ερευνητική προσπάθεια μας. Δεν είναι άλλη από την διπλωματική εργασία την οποία οφείλει να περατώσει ο φοιτητής στο τελευταίο έτος των σπουδών του. Για να είναι αποτελεσματική κυρίως για το φοιτητή, θα πρέπει η επιλογή του θέματος να γίνει προσεκτικά και με σοβαρότητα ώστε να έγκειται του άμεσου ενδιαφέροντος του.

Το δικό μου θέμα, το οποίο έγινε αποδεκτό με την πρώτη και μόνο αναφορά από τον κ. Μηλιώνη Απόστολο, αφορά την ανίχνευση κίνησης με χρήση του λογισμικού ανοικτού κώδικα Motion και τις συναφείς τεχνολογίες που χρησιμοποιούνται για την υλοποίηση και βελτιστοποίηση του με τελικό στόχο την ανάπτυξη ενός ευφυούς δικτυοκεντρικού συστήματος για τη διακριτική βιντεοεπιτήρηση της κατάστασης ασθενών. Κατά συνέπεια, είναι βασισμένη στην ευρεία ανάλυση και ανάπτυξη των σχετικών χαρακτηριστικών του λογισμικού έχοντας ένα τέτοιο εύρος και βάθος, ώστε να δημιουργηθεί ένα ολοκληρωμένο αποτέλεσμα προσαρμοσμένο στις ανάγκες ενός προχωρημένου χρήστη και να εξασφαλίσει συνάμα την ιδιωτικότητα των παρακολουθούμενων. Στοχεύει επίσης, να διευρύνει τα ποιοτικά χαρακτηριστικά της διαδικασίας αυτής υπό την επίδραση νέων προσθηκών.

ΠΕΡΙΛΗΨΗ

Η όλη εργασία δομείται από έξι κεφάλαια. Το πρώτο κεφάλαιο επιχειρεί να θέσει τις βασικές αρχές οι οποίες διέπουν την ψηφιακή εικόνα, την υπολογιστική όραση καθώς και την δυνατότητα της ανίχνευσης κίνησης, ώστε να μπορούν να είναι αυτά κατανοητά από τον μέσο αναγνώστη. Το δεύτερο κεφάλαιο κάνει εκτενή αναφορά στο λογισμικό που χρησιμοποιήθηκε κατά τη διάρκεια εκπόνησης της εν λόγω διπλωματικής εργασίας (Motion), μεταφέροντας πληροφορίες για την εγκατάσταση, τον τρόπο λειτουργίας και την βασική παραμετροποίηση του. Στα επόμενα κεφάλαια, (τρία, τέσσερα και πέντε) υλοποιούνται προχωρημένες ρυθμίσεις και παραμετροποιήσεις ώστε να εκμεταλλευτούν στο έπακρο οι δυνατότητες του Motion και να προσαρμοστούν στις ανάγκες της συγκεκριμένης διπλωματικής εργασίας. Συγκεκριμένα, στο τρίτο κεφάλαιο παρουσιάζονται οι προσαρμοσμένες ρυθμίσεις για τον εντοπισμό κίνησης προσθέτοντας χρήσιμα χαρακτηριστικά όπως οι ειδοποιήσεις του χρήστη με ηλεκτρονικό τρόπο (εισαγωγή βάσης δεδομένων, μήνυμα ηλεκτρονικού ταχυδρομείου e-mail, γραπτό μήνυμα SMS), ενώ στο τέταρτο γίνεται παρουσίαση των γραφικών διεπαφών επικοινωνίας με το χρήστη και της απομακρυσμένης πρόσβασης και διαχείρισης. Στο πέμπτο κεφάλαιο σχεδιάζεται και υλοποιείται η προσθήκη ευφών χαρακτηριστικών στο σύστημα με δυνατότητα πλήρους διαχείρισης από τον χρήστη του συστήματος για την ελαχιστοποίηση ή ακόμη και την αποφυγή των ψευδών συναγερμών (false alarms), δίνοντας ελαστικότητα και προσαρμοστικότητα για χρήση πολλών διαφορετικών σεναρίων βιντεοεπιτήρησης. Ολοκληρώνοντας, στο έκτο κεφάλαιο παρατίθενται τα συμπεράσματα που αναδεικνύουν τη χρησιμότητα όλων των αλλαγών που έγιναν και οι προτάσεις για μελλοντικές επεκτάσεις και εφαρμογές σε άλλους κλάδους.

Μια σύνδεση λοιπόν, που επιχειρείται αναλυτικά στα παρακάτω κεφάλαια έχει ως επακόλουθο να παρουσιάσει την όλη διαδικασία και τα μέσα που χρησιμοποιήθηκαν για την προσαρμογή του λογισμικού Motion στη διασφάλιση της διακριτικότητας, τη βελτιστοποίησή του σε χαρακτηριστικά που δεν υποστηρίζονται στη βασική έκδοση, την εφαρμογή του στο χώρο της υγείας καθώς και στη φιλικότητα προς το χρήστη.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

ανίχνευση κίνησης, ευφύης βιντεοεπιτήρηση, διασφάλιση διακριτικότητας

ABSTRACT

The current thesis is structured by six chapters. The first chapter attempts to explain the basic principles of digital image, computer vision and motion detection's ability, in order to be understood by the average reader. The second chapter makes extensive reference to the software used during the preparation of this thesis (Motion), giving information about the installation, operation and basic configuration. In the following chapters, (three, four and five) advanced settings and customization are taking place to exploit the full potential of Motion and adapt to the needs of this thesis. Specifically, in the third chapter, the reader can find all the custom settings for motion detection in addition to some useful features like alarms by electronic means (import database, e-mail message, text message SMS), while the fourth chapter is a presentation of the basic and the advanced user's graphics communication interface, remote access and management. The fifth chapter presents the design and implementation of the attached intelligent features in a fully managed interface by the system's operator in order to minimize or even avoid false alarms, while having flexibility and adaptability to use in several different video surveillance scenarios. Finally, the sixth chapter presents the findings, highlights the usefulness of all the changes which have been made and suggests extensions and applications in other industries.

A link, therefore, undertaken with many details in the following chapters is able to present the whole process and the means used to adapt the software Motion to ensure privacy and optimization in many points that are not supported in the basic version. Furthermore, it consists on the application in healthcare's sector as well as the user friendliness.

KEY WORDS

Motion detection, intelligent video surveillance, ensuring privacy

Περιεχόμενα:

1. Κεφάλαιο 1 –Γενικές αρχές	9
1.1 : Ψηφιακή παράσταση εικόνας.....	9
1.2 : Ψηφιακή παράσταση video.....	10
1.3 : Αλγόριθμος ανίχνευσης κίνησης.....	11
1.3.2 : Αλγόριθμοι αφαίρεσης φόντου.....	11
1.3.2.1 : Αλγόριθμος διαφοράς διαδοχικών frames.....	12
1.3.2.2 : Αλγόριθμος χρονικής διαφοράς.....	14
1.3.2.3 : Στατιστικές μέθοδοι.....	14
1.3.2.4 : Οπτική ροή.....	16
1.3.2.5 : Συχνά προβλήματα κατά την εφαρμογή των αλγορίθμων.....	16
1.4: Περί της ιδιωτικότητας.....	18
1.4.1 : Ιδιωτικότητα των ασθενών.....	19
1.5 : Προηγούμενες ανάλογες προσεγγίσεις.....	20
2. Κεφάλαιο 2 –Motion.....	22
2.1 : Περιγραφή.....	22
2.2 : Απόκτηση Motion.....	23
2.3 : Χαρακτηριστικά του Motion.....	24
2.4 : Εγκατάσταση και εκτέλεση του Motion.....	29
2.5 : Motion configuration.....	31
2.5.1 : Configuration αρχεία Motion.....	32
2.5.2 : Επιλογές configuration στη γραμμή εντολών.....	33
2.5.3 : Επιλογές αρχείου configuration.....	34
2.6 : Μεταγλώττιση του Motion.....	35
2.7 : Πηγαία αρχεία Motion.....	37
3. Κεφάλαιο 3 –Βασικές ρυθμίσεις εντοπισμού κίνησης.....	46
3.1 : Παραμετροποίηση σχετικά με την εικόνα που λαμβάνεται.....	46
3.1.1 : Προσθέτοντας περισσότερες από μία κάμερες.....	52
3.2 : Παραμετροποιήσεις για την ανίχνευση κίνησης.....	55

3.3 : Χρησιμοποιώντας βάσεις δεδομένων.....	56
3.3.1 : Ρυθμίσεις στην MySQL.....	57
3.4 : Αποστολή e-mail κατά τη δημιουργία ενός γεγονότος.....	59
3.5 : Αποστολή γραπτού μηνύματος SMS κατά τη δημιουργία ενός γεγονότος.....	64
4. Κεφάλαιο 4 –Προχωρημένες ρυθμίσεις επικοινωνίας και αλληλεπίδρασης με το χρήστη.....	67
4.1 : Ρυθμίσεις για το βασικό web interface.....	67
4.1.1 : Δυνατότητες και επιλογές.....	69
4.2 : Απομακρυσμένος έλεγχος και διαχείριση παραμέτρων.....	73
4.2.1 Από υπολογιστές του ίδιου τοπικού δικτύου.....	73
4.2.2 Από οπουδήποτε μέσω εικονικού ιδιωτικού δικτύου (VPN).....	73
4.3 : Εξέλιξη του βασικού web interface.....	76
5. Κεφάλαιο 5 – Προσθήκη ευφών χαρακτηριστικών στο σύστημα με στόχο τη διακριτική παρακολούθηση του ασθενούς.....	88
5.1: Αυτόματη παύση/επανεκκίνηση του Motion.....	88
5.1.1 : Κατά τη διάρκεια προγραμματισμένων επισκέψεων στο χώρο βιντεοεπιτήρησης.....	90
5.1.2 : Σε περίπτωση έκτακτης εισόδου στο χώρο βιντεοεπιτήρησης.....	91
5.2 : Συνδυασμός και συνεκτίμηση αποτελέσματος δεδομένων από πολλαπλές κάμερες.....	92
5.3 : Εκμάθηση από το σύστημα της συνηθισμένης συμπεριφοράς στις κινήσεις του ασθενούς και δημιουργία συναγερμού μόνο σε διαφορετική περίπτωση.....	98
5.3.1 : Σενάριο 1 - Παρακολούθηση ασθενούς με πολλαπλές κάμερες σε ένα δωμάτιο νοσοκομείου.....	108
5.3.2 : Σενάριο 2- Κατ' οίκον παρακολούθηση ασθενούς με πολλαπλές κάμερες σε πολλαπλά δωμάτια.....	112

6. Κεφάλαιο 6 –Συμπεράσματα.....	127
6.1 : Αξιολόγηση των προσθηκών.....	127
6.2 Αξιολόγηση απόδοσης του συστήματος.....	128
6.2 : Μελλοντικές επεκτάσεις- προτάσεις.....	132
Βιβλιογραφία και χρήσιμα internet sites.....	134

Πανεπιστήμιο Πειραιώς

ΚΕΦΑΛΑΙΟ 1-ΓΕΝΙΚΕΣ ΑΡΧΕΣ

1.1 Ψηφιακή παράσταση εικόνας

Είναι γνωστό, ότι έπειτα από τη μετάβαση από τον αναλογικό στον ψηφιακό κόσμο τα σήματα μετατρέπονται από αναλογικά σε ψηφιακά. Βάση του γεγονότος αυτού, μια πραγματική εικόνα μπορεί να μετατραπεί σε ψηφιακή έχοντας υποστεί πρωτίστως δειγματοληψία και κβαντισμό των εντάσεων των σημείων σε διακριτές τιμές, λαμβάνοντας τη μορφή διακεκριμένου σήματος το οποίο έχει τη μορφή ψηφιακών πινάκων. Συγκεκριμένα, μια ψηφιακή εικόνα μπορεί να είναι δυαδική (binary image), μονοχρωματική αποχρώσεων του γκρι (gray level ή grayscale image) ή έγχρωμη (color image). Η δομή της αποτελείται από ένα σύνολο μικρών κουκίδων που ονομάζονται εικονοστοιχεία (pixels). Στην απλούστερη περίπτωση της ασπρόμαυρης ψηφιακής εικόνας κάθε εικονοστοιχείο μπορεί να είναι είτε άσπρο ή μαύρο. Στην παρακάτω εικόνα (Εικόνα 1.1-1) αναπαρίσταται μια μορφή αντικειμένου που αποτελείται από 256 γραμμές και 256 στήλες, περιέχει δηλαδή 65.536 διαφορετικά εικονοστοιχεία. Επειδή η φωτογραφία είναι ασπρόμαυρη, κάθε εικονοστοιχείο κωδικοποιείται με ένα bit. Το 0 συμβολίζει το μαύρο χρώμα και το 1 το άσπρο.



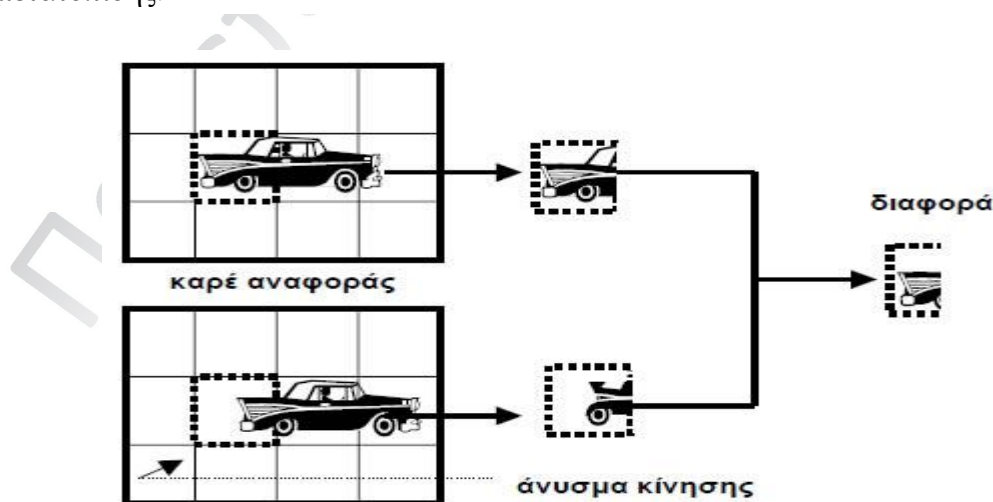
Εικόνα (1.1-1): Ασπρόμαυρη ψηφιακή εικόνα 256 x 256

Για έγχρωμες εικόνες, κάθε εικονοστοιχείο απαιτεί περισσότερα από 1 bit για την αναπαράσταση του αντίστοιχου χρώματος. Για παράδειγμα, αν υπάρχουν 16 διαφορετικά χρώματα και κάθε χρώμα εμφανίζει 16 αποχρώσεις, τότε απαιτούνται $16 \times 16 = 256$ διαφορετικές τιμές.

1.2 Ψηφιακή παράσταση video

Η αρχή πάνω στην οποία στηρίζεται η αναπαραγωγή video είναι ίδια με αυτή του κινηματογράφου. Σε κάθε δευτερόλεπτο εναλλάσσονται αρκετά καρέ εικόνας (frames) με ταχύτητα και μικροδιαφορές μεταξύ τους, δίνοντας στον άνθρωπο την ψευδαίσθηση της κίνησης. Το ανθρώπινο μάτι δεν μπορεί να αντιληφθεί εύκολα αλλαγές με ρυθμό πάνω από 20 καρέ/δευτερόλεπτο και επομένως νομίζει πως βλέπει μια συνεχόμενη σκηνή. Στον κινηματογράφο έχουμε 24 καρέ ανά δευτερόλεπτο, ενώ στην τηλεόραση κυμαίνονται από 25 καρέ/δευτερόλεπτο (σύστημα PAL στην Ευρώπη) έως και 30 καρέ/δευτερόλεπτο (NTSC στην Αμερική).

Οι συχνότητες αυτές αναφέρονται στο ρυθμό εναλλαγής (frame rate) των ακίνητων εικόνων-frames για να σχηματιστεί το video. Η αίσθηση της κίνησης βασίζεται στην ταχύτερη εναλλαγή εικόνων που διαφέρουν πολύ λίγο μεταξύ τους, ώστε το μάτι να μην βλέπει ξαφνικά μεγάλες διαφορές. Η τεχνολογία συμπίεσης σε κινούμενη εικόνα βασίζεται στην απομάκρυνση της πλεονάζουσας πληροφορίας που επαναλαμβάνεται στα διαδοχικά πλαίσια. Η τεχνική που χρησιμοποιείται για την ανίχνευση της πλεονάζουσας πληροφορίας, βασίζεται στην σύγκριση διαδοχικών πλαισίων και στην εύρεση των διαφορών μεταξύ τους. Στη συνέχεια αντί να κωδικοποιηθεί το πλαίσιο, κωδικοποιούνται μόνο οι διαφορές. Το τρέχον πλαίσιο ανακατασκευάζεται από το προηγούμενο και τη διαφορά του με αυτό. Η διαδικασία εύρεσης των διαφορών λέγεται αλγόριθμος εκτίμησης κίνησης (motion estimation algorithm). Υπάρχουν δύο διαδοχικά καρέ, που ονομάζονται καρέ αναφοράς και τρέχον. Τα δύο καρέ διαιρούνται σε στοιχειώδεις μικρές τετραγωνικές περιοχές (blocks). Η κάθε περιοχή του τρέχοντος καρέ αναζητείται στο αναφοράς. Όταν βρεθεί, κωδικοποιείται το διάνυσμα που δείχνει τις συντεταγμένες του σημείου μετατόπισης.



Εικόνα (1.2-1): Διαδικασία εκτίμησης κίνησης video

1.3 Αλγόριθμος ανίχνευσης κίνησης (Motion detection algorithm)

Η ανίχνευση κίνησης βασίζεται κυρίως στην αφαίρεση της μεταβαλλόμενης περιοχής από σταθερά φόντα (background) σε διαδοχικές εικόνες. Είναι μια τεχνολογία κλειδί για την επεξεργασία και ανάλυση εικόνας. Το περιβάλλον στο οποίο γίνεται η εφαρμογή της επιτήρησης έχει αρκετά σημαντικό ρόλο, αφού ανάλογα με τις μεταβολές του καιρού ή του φωτισμού υπάρχουν αρκετές επιπτώσεις στη φωτεινότητα και στους χρωματισμούς της κανονικής εικόνας.

Το στάδιο της ανίχνευσης χωρίζεται στα στάδια της κατάτμησης της κίνησης (Motion Segmentation) και της αναγνώρισης του αντικειμένου (Object Classification). Η κύρια μέθοδος που χρησιμοποιείται είναι αυτή της κατάτμησης της κίνησης, βασίζεται κυρίως στο κατώφλι (threshold) και οι κατηγορίες στις όποιες χωρίζεται είναι οι: Αφαίρεση του φόντου (Background Subtraction), Χρονική διαφορά (Temporal differencing), Στατιστικές μέθοδοι (Statistical-Method) και Οπτική ροή (Optical flow). Η κατάτμηση κίνησης έχει να κάνει με τον διαχωρισμό του κινούμενου αντικειμένου για το οποίο ενδιαφερόμαστε, από την εικόνα του φόντου. Ένας αλγόριθμος κατάτμησης πρέπει να είναι εύρωστος στον θόρυβο και στις αλλαγές του φόντου και του φωτισμού. Στις επόμενες παραγράφους, αναλύονται οι τέσσερις κατηγορίες της κατάτμησης κίνησης.

1.3.2 Αλγόριθμοι αφαίρεσης φόντου (Background subtraction algorithms)

Η αφαίρεση του φόντου είναι μια απλή λύση στην κατάτμηση εικόνας. Μια στατική εικόνα η οποία δεν περιέχει το αντικείμενο ενδιαφέροντος θεωρείται το μοντέλο του φόντου και η εικόνα της κίνησης βρίσκεται από μια pixel προς pixel διαφορά μεταξύ διαδοχικών καρέ και του μοντέλου του φόντου. Αυτή η μέθοδος όμως, δεν είναι κατάλληλη όταν το φόντο αλλάζει δυναμικά και κινείται. Παραλλαγές αυτής της τεχνικής περιλαμβάνουν διαφορετικούς τρόπους υπολογισμού του μοντέλου του φόντου. Το απλούστερο μοντέλο φόντου μπορεί να δημιουργηθεί από τη μέση τιμή του χρόνου των στατικών καρέ. Αντί της μέσης τιμής του χρόνου, η μέση τιμή της τιμής του κάθε pixel (είτε αυτή είναι σε χρωματική, είτε σε γκρι κλίμακα) μπορεί επίσης να χρησιμοποιηθεί για τη δημιουργία του μοντέλου του φόντου. Ο υπολογισμός της μέσης τιμής είναι περισσότερο εύρωστος σε αλλαγές του φωτισμού του φόντου.

Σε αυτή τη μέθοδο αρχικά αφαιρείται το κάθε τρέχον καρέ από την background εικόνα. Έπειτα είναι δυνατόν να συγκριθούν οι αλλαγές στις αποχρώσεις του γκρι και άλλα χαρακτηριστικά. Εάν η απόλυτη τιμή της απόκλισης είναι πέρα από το ορισμένο threshold που έχει προσδιορίσει ο χρήστης, τότε μπορεί να επιβεβαιωθεί η κινούμενη περιοχή.

Η μαθηματική διατύπωση ορίζεται ως εξής:

$$d = |I_t(x, y) - B_L(x, y)|$$
$$M_L(x, y) = \begin{cases} d, & \text{if } d \geq \text{threshold} \\ 0, & \text{if } d < \text{threshold} \end{cases}$$

όπου $B_L(x, y)$ είναι η τιμή της φωτεινότητας της background εικόνας, $M_L(x, y)$ είναι η δυαδική εικόνα από την αφαίρεση της τρέχουσας εικόνας $I_t(x, y)$ και της εικόνας background $B_L(x, y)$ και threshold είναι το αλλαγμένο κατώφλι ανίχνευσης. Η αρχή αυτής της μεθόδου είναι απλή και εύκολα κατανοήσιμη. Ο σχεδιασμός του αλγορίθμου είναι επίσης σχετικά εύκολος. Μπορεί να περιγράψει το μέγεθος, το σχήμα ή το μέρος του κινούμενου στόχου με ακρίβεια. Γενικότερα, όλοι οι κινούμενοι στόχοι μπορούν να προσδιοριστούν με ακρίβεια. Το πλεονέκτημα τέτοιων προσαρμόσιμων μεθόδων είναι ότι εξάγουν το φόντο από τις εικόνες. Συνήθως, το φόντο ορίζεται ως το πιο συχνό χρώμα στην πάροδο του χρόνου. Αυτό σημαίνει ότι τα αντικείμενα που ήταν στην σκηνή κατά την εξαγωγή του φόντου βαθμιαία θα αντικατασταθούν από το φόντο. Αυτό εξαλείφει την ανάγκη της αρχικοποίησης ενός άδειου φόντου. Επίσης, παρέχει ένα μοντέλο φόντου που προσαρμόζεται αυτόματα στις αλλαγές όπως για παράδειγμα, όταν αλλάζει ο καιρός ή όταν ένα παρκαρισμένο αυτοκίνητο φεύγει από τη σκηνή. Το μειονέκτημα αυτής της μεθόδου είναι ότι μπορεί να επηρεαστεί από το φως, τον καιρό και άλλους εξωτερικούς παράγοντες. Ένα ακόμη σημαντικό μειονέκτημα είναι η αλληλεξάρτηση μεταξύ δύο αντιτιθέμενων απαιτήσεων. Από τη μία, η ενημέρωση του φόντου θα πρέπει να γίνεται γρήγορα ώστε να μπορεί να αντιμετωπίσει τις αλλαγές φωτισμού και τις αλλαγές στο φόντο, όπως αλλαγές του φωτισμού λόγω αλλαγής της ώρας ή λόγω ανοίγματος του διακόπτη φωτισμού σε κλειστό χώρο, αλλαγές λόγω ανθρώπων που περπατούν ή λόγω αντικειμένων που φεύγουν από τη σκηνή. Από την άλλη, η ενημέρωση θα πρέπει να γίνεται αργά, ώστε τα αντικείμενα που κινούνται αργά να μην περιλαμβάνονται στο φόντο, όπως ένας άνθρωπος που κοιμάται. Η ταχύτητα λοιπόν, της ενημέρωσης εξαρτάται από την εφαρμογή.

1.3.2.1 Αλγόριθμος διαφοράς διαδοχικών frames

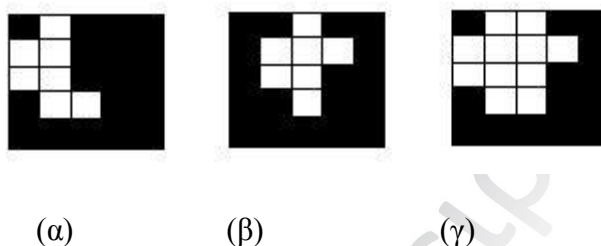
Ο πιο απλός ίσως τρόπος για τον διαχωρισμό του φόντου από το κινούμενο αντικείμενο σε ασπρόμαυρο βίντεο με στατική κάμερα είναι να συγκρίνουμε ανά δύο τα frames του βίντεο παίρνοντας τη διαφορά τους. Αυτό γίνεται ως εξής :

1. Ορίζουμε έναν πίνακα με διαστάσεις ίδιες με αυτές των frames τον οποίο αρχικοποιούμε.
2. Για κάθε δύο διαδοχικά frames παίρνουμε κάθε pixel και βρίσκουμε τη διαφορά της τιμής του μεταξύ των δύο frames.

3. Ορίζουμε ένα κατώφλι (threshold) ($0 < \tau < 1$).

4. Αν η διαφορά των τιμών του pixel είναι μικρότερη της τιμής του κατωφλίου, τότε βάζουμε στην αντίστοιχη θέση του πίνακα την τιμή 1, ενώ αν η διαφορά είναι μεγαλύτερη του κατωφλίου, τότε στην αντίστοιχη θέση βάζουμε την τιμή 0.

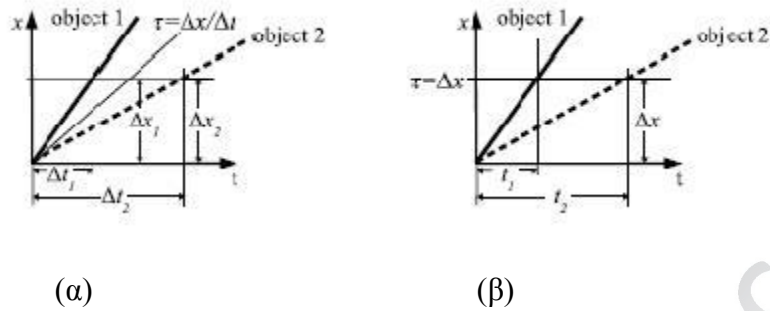
Έτσι, από κάθε δύο διαδοχικά frames παίρνουμε μία ασπρόμαυρη εικόνα όπου τα μαύρα pixel αντιστοιχούν στα pixel που δεν μετακινήθηκαν, δηλαδή στο φόντο, ενώ τα άσπρα pixel αντιστοιχούν στα pixel που άλλαξαν, δηλαδή στο κινούμενο αντικείμενο.



Εικόνα (1.3.2.1-1): Τρία διαδοχικά frames α , β , γ όπου τα μαύρα pixels αντιστοιχούν στα pixels που δεν μετακινήθηκαν, δηλαδή στο φόντο, ενώ τα άσπρα pixels αντιστοιχούν στα pixels που άλλαξαν, δηλαδή στο κινούμενο αντικείμενο.

Με αυτόν τον τρόπο όμως, προκύπτουν διάφορα προβλήματα. Καταρχήν, χρειάζεται μεγάλη προσοχή στην τιμή που θα βάλουμε στο κατώφλι. Αν η τιμή είναι πολύ μικρή, τότε η παραμικρή κίνηση θα ανιχνευθεί, πράγμα που μπορεί να οδηγήσει σε εσφαλμένες εκτιμήσεις, διότι μπορεί να έχουμε μικρές αλλαγές στο φόντο, οι οποίες δε θέλουμε να θεωρηθούν ως κινούμενα αντικείμενα. Τέτοιες αλλαγές μπορεί να οφείλονται σε θόρυβο λόγω της κάμερας, σε αλλαγές φωτισμού, σε σκιές, σε αέρα αν είμαστε σε εξωτερικό περιβάλλον, κτλ. Απ' την άλλη, αν η τιμή του κατωφλίου είναι πολύ μεγάλη, τότε κάποιες αλλαγές μπορεί να μην ανιχνευθούν καθόλου, όπως για παράδειγμα ένα αντικείμενο που κινείται με πολύ μικρή ταχύτητα.

Στα παρακάτω διαγράμματα (Εικόνα 1.3.2.1-2), βλέπουμε ένα αντικείμενο που κινείται γρήγορα (object 1) και ένα που κινείται αργά (object 2). Στο διάγραμμα που βρίσκεται αριστερά, το κατώφλι που επιλέχτηκε, καθορίστηκε με βάση την ταχύτητα των σωμάτων που κινούνται, με αποτέλεσμα το αντικείμενο που κινείται αργά να μη μπορεί ποτέ να ανιχνευθεί ($\tau = x/t$). Αντίθετα, στο διάγραμμα που βρίσκεται στα δεξιά, το κατώφλι τ εξαρτάται από ένα προκαθορισμένο frame αναφοράς ($\tau = \Delta x$). Αυτό οδηγεί στην ανίχνευση και των δύο αντικειμένων, ανεξάρτητα από την ταχύτητά τους.



Εικόνα (1.3.2.1-2): Δύο αντικείμενα που κινούνται. α. Το κατώφλι καθορίστηκε με βάση την ταχύτητα των σωμάτων που κινούνται, με αποτέλεσμα το αντικείμενο που κινείται αργά να μη μπορεί ποτέ να ανιχνευθεί. β. Το κατώφλι εξαρτάται από ένα προκαθορισμένο frame αναφοράς ανεξάρτητα από την ταχύτητα των αντικειμένων.

1.3.2.2 Αλγόριθμος χρονικής διαφοράς (Temporal difference algorithm)

Η συγκεκριμένη μέθοδος μπορεί να εξάγει την κινούμενη περιοχή από δύο ή τρία διαδοχικά καρέ μέσω του threshold της απόλυτης τιμής. Εάν η τιμή είναι μεγαλύτερη από το ορισμένο κατώφλι αποφασίζεται ότι υπάρχουν κινούμενοι στόχοι στην προς μελέτη περιοχή. Η εξίσωση είναι:

$$d = |I_n(x, y) - I_{n-1}(x, y)|$$

$$M_n(x, y) = \begin{cases} d, & \text{if } d \geq \text{threshold} \\ 0, & \text{if } d < \text{threshold} \end{cases}$$

όπου $M_n(x, y)$ είναι η δυαδική εικόνα από και την αφαίρεση της τρέχουσας εικόνας $I_n(x, y)$ και της προηγούμενης εικόνας $I_{n-1}(x, y)$, ενώ το threshold είναι το αλλαγμένο κατώφλι ανίχνευσης. Το πλεονέκτημα αυτής της μεθόδου είναι ότι είναι ικανή να προσαρμοστεί σε κάθε δυναμικό περιβάλλον. Επιπρόσθετα, δεν είναι καθόλου πολύπλοκη και είναι αρκετά σταθερή. Το μειονέκτημα είναι ότι δεν μπορεί να εξάγει ολόκληρες περιοχές κινούμενων στόχων, αλλά μονό το περιμετρικό μέρος, επομένως δεν είναι κατάλληλη για ανίχνευση αντικειμένων που κινούνται πολύ γρήγορα.

1.3.2.3 Στατιστικές μέθοδοι (Statistical methods)

Όταν απαιτείται ανίχνευση κίνησης σε ένα σύστημα βιντεοεπιτήρησης, μπορούν να εφαρμοσθούν πολλές μέθοδοι. Όπως προτυπώθηκε, ο πιο διαδεδομένος τρόπος να αναπτυχτεί ένα τέτοιο σύστημα είναι ο υπολογισμός της απολυτής τιμής της διαφοράς μεταξύ του τρέχοντος καρέ και ενός καρέ αναφοράς. Έπειτα, ο πίνακας διαφορών κατωφλιώνεται. Τα pixels εκείνα τα οποία έχουν τιμή υψηλότερη από το επιλεγμένο κατώφλι θεωρούνται ως μέρος ενός κινούμενου αντικειμένου στη σκηνή, ενώ εκείνα που έχουν μικρότερη τιμή από το κατώφλι ανήκουν στο background.

Το καρέ αναφοράς μπορεί να είναι είτε το προηγούμενο καρέ ή μια εκτίμηση του background υπολογισμένη από τα προηγούμενα επεξεργασμένα καρέ. Το κρίσιμο σημείο σε αυτού του είδους τα συστήματα είναι η τιμή του κατώφλιου που θα πρέπει να επιλεγεί, και να εφαρμοστεί σε όλη την εικόνα. Αν το επιλεγμένο κατώφλι είναι πολύ μικρό, θα αυξηθούν οι πιθανότητες για ψευδής συναγερούς εξαιτίας των βασικών θορύβων που ανιχνεύονται από την κάμερα καθώς και το θόρυβο που γεννάται από ένα μη στατικό background όπως ένα κλαδί δέντρου που κινείται κατά τη διάρκεια μιας ημέρας με άνεμο. Από την άλλη, όταν η τιμή του κατώφλιου οριστεί να είναι πολύ υψηλή, υπάρχει πολύ αυξημένη πιθανότητα να χαθεί πληροφορία καθώς η απολυτή διάφορα μεταξύ του επίπεδου του γκρι στο background και του αντικείμενου είναι πιθανό να είναι χαμηλότερη από το τρέχον κατώφλι. Επιπλέον, σε πραγματικές βίντεο εικόνες, η στάθμη του θορύβου και οι συνθήκες φωτισμού ενδέχεται να είναι διαφορετικές σε κάθε κομμάτι καρέ.

Για την επίλυση του προαναφερθέντος προβλήματος, σε πολλές περιπτώσεις χρησιμοποιείται μια στατιστική μέθοδος που έχει ως σκοπό την προσπάθεια πρόβλεψης των χαρακτηριστικών του κάθε pixel της εικόνας, συνάρτηση του χρόνου. Για να επιτευχθεί αυτό, ο αλγόριθμος εγκαθιδρύει μια περίοδο εκπαίδευσης κατά την οποία υποθετικά δεν υπάρχει κίνηση σε όλη την αλληλουχία και η διακύμανση του κάθε pixel συμβαίνει εξαιτίας του θορύβου. Ο υπολογισμός της στατιστικής των pixels είναι πιθανόν να προσδιορίσει τη συμπεριφορά κάθε τμήματος της εικόνας και με αυτόν τον τρόπο να αποφασίσει εάν το επόμενο δείγμα ενός συγκεκριμένου pixel ανταποκρίνεται ξανά στο background, εάν η τιμή του είναι κοντά στην αναμενόμενη σύμφωνα με τα υπολογισθέντα στατιστικά, ή ανήκει σε ένα νέο αντικείμενο που κινείται. Οι πιο συνηθισμένοι αλγόριθμοι που χρησιμοποιούνται σε τέτοιες περιπτώσεις, υποθέτουν ότι η probability density function (pdf) είναι γκαουσιαννή και αφού υπολογίσουν την μέση (μ) και την τυπική απόκλιση (σ) το επόμενο pixel μπορεί να ληφθεί ως τμήμα του background εφόσον η τιμή του είναι μέσα στα όρια ($\mu - k\sigma$, $\mu + k\sigma$) ή διαφορετικά ως κίνηση όπου το k είναι μια αυθαίρετη σταθερά που επιλέγεται από τον χρήστη.

Παρόλο που η συγκεκριμένη μέθοδος εργάζεται σωστά εάν το pixel έχει χαμηλή στάθμη θορύβου, όταν υπάρχει μεγαλύτερη ένταση του θορύβου στο background εξαιτίας, για παράδειγμα, ενός κλαδιού δέντρου κατά τη διάρκεια μιας ημέρας με έντονο αέρα ή σε ένα αυτοκινητόδρομο υψηλής κίνησης στον οποίο η κίνηση του αυτοκινήτου δεν είναι επιθυμητό να λαμβάνεται υπόψη από το σύστημα, η διακύμανση αυτής της ζώνης θα είναι πολύ υψηλή με αποτέλεσμα το εσωτερικό επίπεδο του γκρι στον οποίο το νέο δείγμα μπορεί να ληφθεί ως κίνηση είναι πολύ μικρό, αυξάνοντας τις πιθανότητες να χαθεί πληροφορία σε αυτές τις ζώνες.

Επιπλέον, εάν σε μια συγκεκριμένη στιγμή υπάρξει μια ξαφνική αλλαγή στον φωτισμό χωρίς να υπάρχει ταυτόχρονα κανένα άλλο είδος κίνησης στη σκηνή, οι αλγόριθμοι που αναφέρθηκαν παραπάνω αποτυγχάνουν επειδή όταν γίνεται ο έλεγχος για την απολυτή διαφορά ή την τυπική απόκλιση, οι αλγόριθμοι συνήθως μαρκάρουν αυτό το pixel ως pixel κίνησης ενώ θα έπρεπε να είχε μαρκαριστεί ως background.

Εάν η αλλαγή φωτισμού αφορά όλη την σκηνή, τότε σχεδόν όλη η επιφάνεια της εικόνας θα ληφθεί ως κίνηση.

1.3.2.4 Οπτική ροή (Optical flow)

Οι τεχνικές της οπτικής ροής βασίζονται στην υπόθεση ότι η ένταση των pixels σε μία ακολουθία εικόνων δεν αλλάζει. Με την οπτική ροή όμως, είναι αδύνατον να προσδιορίσουμε την ταχύτητα της εικόνας στην διεύθυνση κάθετη με την βάθμωση της έντασης της εικόνας. Αυτή η αδυναμία αναφέρεται ως το πρόβλημα του ανοίγματος (aperture problem). Η οπτική ροή είναι πολύ πολύπλοκη υπολογιστικά και απαιτεί η εσωτερική κίνηση των χαρακτηριστικών των καρέ να είναι μικρή. Απ' την άλλη, έχει το πλεονέκτημα ότι μπορεί να διαχωρίσει την κίνηση μεταξύ ενός σώματος στερεού που δεν αλλάζει το σχήμα του και ενός που αλλάζει, διότι η κίνηση του σώματος που δεν αλλάζει παρουσιάζει μικρή συνεχή ροή.

1.3.2.5 Συχνά προβλήματα κατά την εφαρμογή των αλγορίθμων

Κάθε ανιχνευμένη αλλαγή pixel θα πρέπει να λογίζεται ως μέρος του αντικειμένου που βρίσκεται στο προσκήνιο. Για το λόγο αυτό, μπορούν να εφαρμοστούν οι τεχνικές που βασίζονται σε χρονική πληροφορία χρησιμοποιώντας μια καταφλιωμένη διαφορά καρέ. Ανάλογα με την χρονική σχέση μεταξύ των καρέ που βασίζονται στη διαφορά, μπορούν να οριστούν δυο διαφορετικές προσεγγίσεις.

Στην πρώτη προσέγγιση, η τεχνική background subtraction χρησιμοποιεί ένα καρέ αναφοράς ώστε να αναπαραστήσει το background. Το εν λόγω καρέ είναι συνήθως ορισμένο από την πρώτη ληφθείσα εικόνα. Επομένως, ένα pixel μπορεί να κατηγοριοποιηθεί ως προσκήνιο εάν διαφέρει αρκετά από την αντίστοιχη τιμή του στο καρέ αναφοράς. Παρόλο που η λύση αυτή φαίνεται ιδανική, αξίζει να σημειωθεί ότι σε πραγματικά περιβάλλοντα μπορούν να λάβουν χώρα δυο διαφορετικές περιπτώσεις:

- **Ιδανική περίπτωση:** Δεν υπάρχουν αντικείμενα προσκηνίου στο καρέ αναφοράς. Στην περίπτωση αυτή, η προκύπτουσα εικόνα θα είναι η ίδια με το επιθυμητό αποτέλεσμα τμηματοποίησης.
- **Γενική περίπτωση:** Τα αντικείμενα του προσκηνίου μπορεί να εμφανιστούν στο καρέ αναφοράς. Η παρουσία τους καθιστά αποτυχημένη την μέθοδο του background subtraction, αφού παράγει ψευδείς συναγερμούς εξαιτίας της θέσης τους στο καρέ αναφοράς.

	ΚΑΡΕ ΑΝΑΦΟΡΑΣ	ΤΡΕΧΟΝ ΚΑΡΕ	ΑΠΟΤΕΛΕΣΜΑ ΑΦΑΙΡΕΣΗΣ ΦΟΝΤΟΥ
ΙΔΑΝΙΚΗ ΠΕΡΙΠΤΩΣΗ			Το αντικείμενο στο τρέχον καρέ 
ΓΕΝΙΚΗ ΠΕΡΙΠΤΩΣΗ			Το αντικείμενο στο καρέ αναφοράς (ghosting)  Το αντικείμενο στο τρέχον καρέ 

Εικόνα (1.3.2.5.-1): Διαφορετικές συνθήκες-περιπτώσεις κατά τη σύγκριση καρέ αναφοράς και τρέχοντος καρέ

Στη δεύτερη προσέγγιση, μπορούν να χρησιμοποιηθούν οι τεχνικές που βασίζονται στα γειτονικά χρονικά καρέ. Κατά κύριο λόγο, αυτές οι μέθοδοι που βασίζονται στην χρονική διαφορά στηρίζονται στο γεγονός ότι ένα pixel κινείται, όταν η ένταση του έχει μεταβληθεί σημαντικά μεταξύ του τρέχοντος και του αμέσως προηγούμενου καρέ. Παρά το γεγονός ότι η μέθοδος αυτή παρέχει μια εύκολη και γρήγορη ανίχνευση του κινούμενου αντικείμενου, λειτουργεί μόνο σε συγκεκριμένες συνθήκες ταχύτητας και framerate αντικειμένου, διότι δημιουργούν δύο μεγάλα μειονεκτήματα: το ghosting και το foreground aperture (διαπλάτωση-άνοιγμα προσκηνίου). Όπως είναι προφανές στην παρακάτω εικόνα (Εικόνα 1.3.2.5-2), η παρουσία ενός αντικειμένου στο προηγούμενο καρέ παράγει ψευδείς συναγερμούς (ghosting) ενώ η ομοιότητα μεταξύ των pixels όταν η ταχύτητα του αντικειμένου είναι πολύ χαμηλή ή όταν αυτό ακινητοποιείται, δημιουργεί το πρόβλημα της διαπλάτωσης προσκηνίου (foreground aperture).



Εικόνα (1.3.2.5-2): Απεικόνιση των δύο κυριότερων μειονεκτημάτων

1.4 Περί της ιδιωτικότητας

Η ιδιωτικότητα αποτελεί ανθρώπινο δικαίωμα, το οποίο αναγνωρίστηκε στην Διακήρυξη των Ανθρωπίνων δικαιωμάτων των Ηνωμένων Εθνών και πρέπει να προστατεύεται από τους νόμους, τους κώδικες δεοντολογίας, την τεχνολογία καθώς και να υπάρχει εκπαίδευση των ατόμων σχετικά με αυτή.

Ο πρώτος ορισμός της ιδιωτικότητας δόθηκε από τον Samuel D. Warren και τον Luis D. Brandeis και αναφέρεται στο δικαίωμα να απολαμβάνεις μόνος χωρίς παρακολούθηση συγκεκριμένες στιγμές της ζωής. Με το πέρασμα των χρόνων διατυπώθηκε και ένας άλλος ορισμός από τον Alan Westin, ο οποίος σήμερα είναι ο επικρατέστερος και ορίζει την ιδιωτικότητα ως την αξίωση των ατόμων, των ομάδων και των ιδρυμάτων, να αποφασίζουν από μόνοι τους για το πότε και μέχρι ποιο σημείο οι πληροφορίες που αφορούν αυτούς, θα διαβιβάζονται σε άλλους.

Αξίζει επίσης να σημειωθεί, ότι η έννοια της ιδιωτικότητας χωρίζεται σε 3 μορφές: στην εδαφική ιδιωτικότητα που σχετίζεται με την προστασία της φυσικής περιοχής που περιβάλλει ένα πρόσωπο, στην ιδιωτικότητα του ατόμου που αφορά την προστασία ενός ατόμου από την αδικαιολόγητη παρέμβαση και στην πληροφοριακή ιδιωτικότητα που αφορά τον έλεγχο του αν και πως τα προσωπικά στοιχεία μπορούν να χρησιμοποιηθούν με οποιοδήποτε τρόπο.

Επιπρόσθετα, η προσωπική πληροφοριακή ιδιωτικότητα αποτελεί το δικαίωμα της πληροφοριακής αυτοδιάθεσης. Ωστόσο, η ιδιωτικότητα δεν είναι ένα απεριόριστο δικαίωμα, καθώς μπορεί να έρθει σε ρήξη με άλλες νομικές αξίες, αλλά και επειδή τα άτομα σε πολλές περιπτώσεις είναι υποχρεωμένα να αποκαλύψουν κάποια προσωπικά τους στοιχεία στην κοινωνία. Υπάρχουν, όμως, νόμοι Προστασίας της Ιδιωτικότητας και των Δεδομένων προκειμένου να προστατεύονται τα δικαιώματα της Ιδιωτικότητας.

Για να προστατευθεί το δικαίωμα της πληροφοριακής αυτοδιάθεσης είναι απαραίτητη η εξασφάλιση κάποιων βασικών αρχών της ιδιωτικότητας, όταν πρόκειται να γίνει συλλογή ή επεξεργασία προσωπικών δεδομένων. Αυτές οι αρχές έχουν επιβληθεί από τον Γερμανικό Ομοσπονδιακό Νόμο Προστασίας των Δεδομένων. Επιγραμματικά, κάποιες από τις σημαντικότερες αυτές αρχές είναι αυτές της νομιμότητας και της δικαιοσύνης, της προδιαγραφής και της αντιστοίχισης σκοπού, της αναγκαιότητας συλλογής και επεξεργασίας δεδομένων, της πληροφόρησης και των δικαιωμάτων πρόσβασης στα άτομα στα οποία ανήκουν τα δεδομένα, της ασφάλειας και της ακρίβειας και τέλος της επίβλεψης και των κυρώσεων.

Το σημαντικότερο πρόβλημα που απειλεί την ιδιωτικότητα είναι η αυξανόμενη ανάπτυξη του παγκόσμιου δικτύου και ειδικότερα του διαδικτύου όπου δεν υπάρχει δυνατότητα κεντρικής διαχείρισης. Συνεπώς, ο λόγος ύπαρξης των κινδύνων της ιδιωτικότητας οφείλεται στο γεγονός ότι τα προσωπικά δεδομένα που αφορούν τους χρήστες είναι διαθέσιμα σε διαφορετικούς ιστότοπους σε όλον τον κόσμο. Η ταχύτατη εξέλιξη της παγκόσμιας κοινωνίας της πληροφορίας συντελεί στην ανάπτυξη νέων εφαρμογών πληροφορικής σε διάφορους τομείς. Επομένως, υπάρχει ένας συνεχώς αυξανόμενος όγκος προσωπικών δεδομένων, τα οποία επεξεργάζονται και διατίθενται μέσω δικτύων σε όλα τα κράτη.

1.4.1 Ιδιωτικότητα των ασθενών

Έστω ένα υγειονομικό δίκτυο στο οποίο συνδέονται γιατροί, νοσοκομεία και κοινωνικά κέντρα σε ευρωπαϊκή κλίμακα. Μέσα από αυτό, εκφράζεται η χρησιμότητα ενός ηλεκτρονικού ιατρικού φακέλου των ασθενών ο οποίος μπορεί να βοηθήσει σε πολλούς τομείς, όπως η διάγνωση ασθενειών ή ο σχεδιασμός αποτελεσματικών θεραπειών.

Ωστόσο, τα ιατρικά αρχεία των ασθενών μπορεί να περιέχουν προσωπικές και ευαίσθητες πληροφορίες, οι οποίες δίχως τη σωστή προστασία μπορεί να επιφέρουν άσχημα αποτελέσματα, όπως κοινωνική αμηχανία, επιπτώσεις στη δυνατότητα κοινωνικής ασφάλισης κ.α. Είναι προφανές, πως χωρίς εμπιστοσύνη στην ιατρική ιδιωτικότητα, οι ασθενείς μπορεί να αποφεύγουν την πολύ σημαντική για εκείνους υγειονομική περίθαλψη.

Εξαιτίας της ανάπτυξης των νέων δικτύων υγειονομικής περίθαλψης σε παγκόσμιο επίπεδο, τα ιατρικά δεδομένα που θα συλλέγονται και θα μεταφέρονται σε διάφορους τόπους ολοένα και θα αυξάνονται. Είναι σημαντικό, λοιπόν, να υπάρχει πλήρης μυστικότητα και προστασία των εν λόγω δεδομένων. Αυτό μπορεί να επιτευχθεί μόνο εάν ένα εξουσιοδοτημένο πρόσωπο έχει πρόσβαση στις ιατρικές πληροφορίες και πρέπει επίσης να υπάρχει διαφάνεια με σκοπό οι ασθενείς να ξέρουν ποιος και για ποιο λόγο έχει πρόσβαση στα δεδομένα τους.

Ειδικότερα, στην περίπτωση της βιντεοεπιτήρησης ασθενών που λαμβάνει χώρα στην τρέχουσα διπλωματική εργασία, παρουσιάζονται τα αντίστοιχα προβλήματα σχετικά με την ιδιωτικότητα. Είναι προφανές ότι οι ασθενείς που παρακολουθούνται δεν επιθυμούν να είναι ορατή η εικόνα τους από οποιονδήποτε σε όποια κατάσταση και αν βρίσκονται είτε λόγω ντροπής εξαιτίας της κατάστασης τους ή της ενδυμασίας τους αλλά και λόγω διαμόρφωσης καλύτερης ψυχολογίας. Για το λόγο αυτό προστίθενται και αξιοποιούνται όσο το δυνατόν λιγότερες πληροφορίες σχετικά με την εικόνα που λαμβάνεται οι οποίες όμως είναι επαρκέστατες. Συγκεκριμένα, από το πρόγραμμα μεταφέρεται αρχικά μόνο η πληροφορία της κίνησης και όχι η εικόνα του ασθενή που κινείται. Φυσικά, σε περίπτωση έκτακτης ανάγκης, υπάρχει η δυνατότητα επαναφοράς της πραγματικής εικόνας και πλήρους απεικόνισης της κατάστασης του ασθενούς.

Συμπερασματικά, είναι σαφές το γεγονός ότι υπάρχουν σοβαροί κίνδυνοι για την ιδιωτικότητα σε ένα υγειονομικό δίκτυο. Επομένως, είναι αναγκαία η προστασία των προσωπικών δεδομένων, τα οποία μπορούν εύκολα να δρομολογηθούν σε διάφορες χώρες και ύστερα να διαβαστούν ή ακόμα και να τροποποιηθούν από άλλους, είτε με τη χρήση των νόμων περί προστασίας είτε με τη χρήση της τεχνολογίας, όπως προσπαθεί να επιτύχει και η συγκεκριμένη διπλωματική εργασία.

1.5 Προηγούμενες ανάλογες προσεγγίσεις

Η συνεχής ανάπτυξη αυτοματοποιημένων και ευφυών συστημάτων δικτύων βιντεοεπιτήρησης, έχει κεντρίσει το ενδιαφέρον αρκετών ερευνητών ώστε να εφαρμόσουν νέες τεχνικές διασφάλισης της ιδιωτικότητας κατά τη βιντεοπαρακολούθηση χώρων και ανθρώπων γενικότερα και ασθενών ειδικότερα.

Στην γενικότερη περίπτωση, είναι φυσιολογικό να υπάρχει αρκετά περισσότερο ερευνητικό υλικό. Πολύ αξιόλογη ερευνητική προσπάθεια αποτελεί η δημιουργία ενός δικτύου πολλαπλών καμερών με τοποθέτηση ενός κεντρικού συστήματος RFID στις εισόδους των χώρων βιντεοεπιτήρησης ώστε να γίνεται αναγνώριση των προς παρακολούθηση οντοτήτων (οι οποίες με τη σειρά τους διαθέτουν μια μοναδική ετικέτα ID) και στη συνέχεια απόδοση από τις κάμερες, ανάλογα με το επίπεδο πρόσβασης του διαχειριστή, ορισμένων μόνο χαρακτηριστικών της εμφάνισης του αντικειμένου ή του ανθρώπου που εισέρχεται στο χώρο βιντεοεπιτήρησης ύστερα από σύγκριση με καταχωρημένα χαρακτηριστικά που απεικονίζουν τη φυσιολογική κατάσταση και που ήδη έχουν καταχωρηθεί σε βάσεις δεδομένων [11]. Επιπρόσθετα, έχει ήδη ερευνηθεί και η περίπτωση ενός συστήματος βιντεοεπιτήρησης εντός οικίας χωρίς καταγραφές, όπου αυτό είναι ενημερωμένο με αρχεία βίντεο που εμπεριέχουν διαφορετικά σενάρια φυσιολογικής ή μη κατάστασης και χρησιμοποιώντας αλγορίθμους σύγκρισης σε περίπτωση μη φυσιολογικής κατάστασης εξάγει εικόνες με φίλτρα παραμόρφωσης ώστε να μην υπάρχει αναγνώριση του παρακολουθούμενου [13]. Η προσπάθεια αυτή έχει επεκταθεί και σε τεχνική με χρήση γραφικής διεπαφής ώστε παρόλο που η πληροφορία από τη βάση δεδομένων μπορεί να φτάσει ήδη με επικάλυψη στοιχείων, ο τελικός χρήστης να μπορεί να κάνει περαιτέρω ανάλυση του βίντεο από το γραφικό περιβάλλον ώστε να παραμορφώσει περισσότερο την κανονική εικόνα του παρακολουθούμενου παραποιώντας την ταυτότητά του και έτσι να αυξήσει την ιδιωτικότητα [10]. Ακόμη, στη βιβλιογραφία υπάρχουν τεχνικές όπου γίνεται κρυπτογράφηση της πληροφορίας του κινούμενου αντικειμένου και δυνατότητα απεικόνισης αυτού μόνο στο διαχειριστή του συστήματος [12]. Ακολουθεί και σε αυτή την περίπτωση χρήση φίλτρου για δημιουργία παραμόρφωσης της εικόνας του παρακολουθούμενου, δημιουργία αλγορίθμου για αναγνώριση ανθρώπου μέσω προσώπου, ματιών, φωνής κτλ. και επικάλυψη ή ακόμα και αφαίρεση του προσώπου ώστε να μην είναι ορατό στις καταγραφές. Σε περίπτωση βιντεοπαρακολούθησης σε εξωτερικές συνθήκες αξίζει να αναφερθεί η τεχνική ενημέρωσης του συστήματος για τη φυσιολογική εικόνα παρασκηνίου που θα πρέπει να λαμβάνεται κανονικά και της πιθανής πορείας που θα ακολουθήσει ο παρακολουθούμενος και σε αντίθετη περίπτωση ανάλογα με το επίπεδο προστασίας της ιδιωτικότητας που χρειάζεται, γίνεται εφαρμογή φίλτρων για απόκρυψη χαρακτηριστικών και εμφάνιση μόνο ανυσμάτων και απροσδιόριστων φιγούρων σε διαφορετικά χρώματα ανάλογα πάντα με το επίπεδο προστασίας [8]. Σε αντίστοιχη δημοσίευση παρουσιάζεται η δυνατότητα παρακολούθησης με εξελιγμένους αλγορίθμους της πορείας του παρακολουθούμενου και απεικόνιση της με άνυσμα, προσπάθεια ταύτισης από πολλαπλές κάμερες του ίδιου ατόμου σε διαφορετικά σημεία επιτήρησης και απόκρυψη προσώπου με φίλτρα παραμόρφωσης εικόνας [14]. Τέλος, έχει παρουσιαστεί μια διαφορετική μέθοδος και πάλι σε εξωτερικές συνθήκες βιντεοεπιτήρησης όπου το προτεινόμενο σύστημα αποτελείται από κάμερες καθοδηγούμενες με μικροεπεξεργαστή [9, 15] οι οποίες έχουν δυνατότητες για κινήσεις pan/tilt, ώστε έπειτα από μελέτη και δειγματοληψία των προς απόκρυψη

περιοχών να κάνουν κινήσεις ώστε να απεικονίζουν την πληροφορία χωρίς την περιοχή που είναι επιθυμητό να διασφαλιστεί η ιδιωτικότητα.

Όσον αφορά τη διασφάλιση της ιδιωτικότητας κατά τη διάρκεια επιτήρησης ασθενών, σημαντικό ρόλο κατέχει η τεχνική ανίχνευσης των πιθανών πτώσεων ασθενών που επιτηρούνται σε δωμάτια νοσοκομείων [16]. Για λόγους διακριτικότητας τοποθετούνται στους ασθενείς ασύρματοι αισθητήρες που παρακολουθούν τις κινήσεις τους οι οποίες καταγράφονται σε βάσεις δεδομένων. Έχοντας δημιουργήσει τρισδιάστατα ομοιώματα του σώματος του ασθενούς και γνωρίζοντας τη μορφή που θα πρέπει να έχουν αυτά σε φυσιολογική ή μη περίπτωση, ο υπολογιστής που λαμβάνει τα δεδομένα συγκρίνει με αλγόριθμους τα αποτελέσματα και εξάγει αποφάσεις συναγερμού. Επιπρόσθετα, έχει μελετηθεί εκτενώς και η ασφάλεια κατά τη μετάδοση δεδομένων ιατρικής πληροφορίας από αισθητήρες σώματος [17]. Η αποστολή και η λήψη των δεδομένων ελέγχεται από τους πιο εξελιγμένους αλγόριθμους κρυπτογράφησης και γίνεται διαχωρισμός ως προς το βαθμό εμπιστοσύνης κάθε μηχανισμού καθώς και διασφάλιση της ανωνυμίας του ασθενούς κατά τη διάρκεια της μετάδοσης, ώστε η πληροφορία να καταλήγει μόνο σε εξουσιοδοτημένο νοσηλευτικό προσωπικό. Η παραπάνω έρευνα έχει συνεχιστεί και σε περίπτωση ελέγχου της πληροφορίας των ασύρματων αισθητήρων και από κινητά τηλέφωνα τελευταίας γενιάς [18] ώστε να γίνεται η διαχείριση μέσω ενός διαχειριστή σύνδεσης στην πύλη gateway του δικτύου και να κρυπτογραφείται ανάλογα πριν φτάσει στους αποδέκτες μέσω του εξυπηρετητή.

ΚΕΦΑΛΑΙΟ 2-MOTION

2.1 Περιγραφή

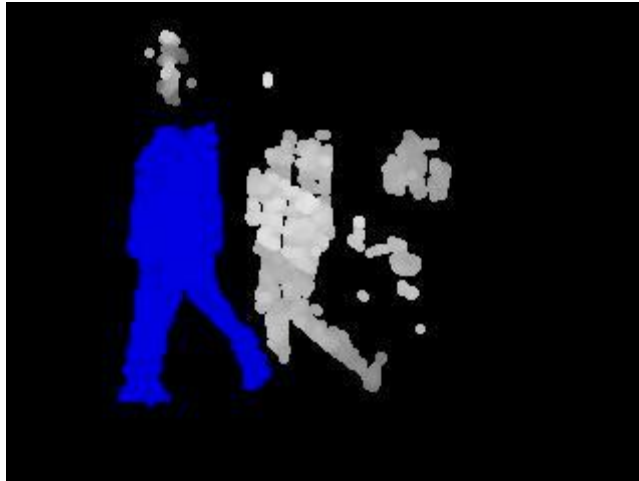
Το λογισμικό Motion χρησιμοποιεί μια συσκευή video4linux (V4L) για να ανιχνεύσει την κίνηση. Αν ανιχνευτεί κίνηση μπορούν να εξαχθούν είτε κανονικές εικόνες, είτε εικόνες Motion. Είναι δυνατόν να εξαγεί επίσης αυτοματοποιημένα στιγμιότυπα (snapshots).

Το Motion καλείται από την γραμμή εντολών και δεν έχει απολύτως καμία γραφική διεπαφή (GUI). Τα πάντα ελέγχονται από την γραμμή εντολών και από τα αρχεία configuration. Συνίσταται γενικά η χρησιμοποίηση μόνο των αρχείων configuration καθώς καθίσταται πιο εύκολη η μακροχρόνια εκτέλεση του προγράμματος αφού είναι λιγότερο πολύπλοκο να υπάρχουν όλες οι παραμετροποιήσεις σε ένα αρχείο. Μερικοί, χρήσιμοι για τη συνέχεια ορισμοί, σχετικά με τον τύπο των εικόνων που αναφέρθηκε στην αρχή της προηγούμενης παραγράφου, αναλύονται στις παρακάτω γραμμές:

- Η κανονική εικόνα είναι ολόκληρη η εικόνα που λαμβάνεται από την κάμερα.
- Η εικόνα Motion αποτελεί τα pixels τα οποία έχουν πραγματικά μεταβληθεί κατά τη διάρκεια των τελευταίων frames. Αυτές οι εικόνες δεν είναι αρκετά χρήσιμες προς παρουσίαση στους χρήστες αλλά βοηθούν πάρα πολύ κατά τη διαδικασία ελέγχου, ρύθμισης και δημιουργίας αρχείων mask από όπου ο χρήστης μπορεί να βρει το ακριβές σημείο στο οποίο το Motion ανιχνεύει την κίνηση.



Εικόνα (2.1-1): Καρέ κανονικής εικόνας



Εικόνα (2.1-2): Καρέ εικόνας Motion

Η μεγαλύτερη περιοχή έχει χρώμα μπλε και μόνο αυτή είναι που λαμβάνεται ως κίνηση. Η εικόνα Motion είναι χρήσιμη διότι αποκαλύπτει με ποιο τρόπο το Motion διατηρεί ένα καρέ αναφοράς το οποίο δεν είναι μόνο το καρέ της τελευταίας εικόνας αλλά ένας μαθηματικός υπολογισμός των προηγούμενων εικόνων. Φυσικά, η ιδιότητα αυτή συμβάλλει στην σωστή ανίχνευση από το σύστημα και στην αποφυγή ψευδών συναγερωμών.

- Το στιγμιότυπο (snapshot) είναι μια εικόνα η οποία λαμβάνεται ανά τακτά χρονικά διαστήματα, ανεξάρτητα από το γεγονός αν υπάρχει κίνηση ή όχι στο προσκήνιο.

2.2 Απόκτηση Motion

Το Motion είναι ένα λογισμικό ανοιχτού κώδικα υπό τους κανόνες συμμόρφωσης GNU GENERAL PUBLIC LICENCE (GPL) και προφανώς δεν κοστίζει η απόκτησή του, αφού ο οποιοσδήποτε χρήστης Linux μπορεί να κάνει λήψη και εγκατάσταση στον προσωπικό του υπολογιστή χωρίς να καταχωρηθεί κάπου ή να ζητήσει την άδεια από τον διαχειριστή. Το εκτελέσιμο για τις δημοφιλέστερες εκδόσεις Linux (Debian, Ubuntu, RPM) αρχείο εγκατάστασης του Motion βρίσκεται στην κεντρική ιστοσελίδα του Motion (Motion Homepage).

Το Motion αναπτύχθηκε αρχικά από τον Jeroen Vreeken ο οποίος ακόμη συμμετέχει ενεργά στην περαιτέρω ανάπτυξη του λογισμικού και μετέπειτα από τον Folkert van Heusden ως επικεφαλής προγραμματιστή. Τέλος, αρκετά ενεργό ρόλο έχει ο Kenneth Lavrsen ο οποίος είναι υπεύθυνος για τον Motion Guide και το Motion website και από την έκδοση 3.1.12 και έπειτα είναι ο υπεύθυνος διαχειριστής του έργου.

Η τρέχουσα έκδοση του Motion, η οποία χρησιμοποιήθηκε και στο εν λόγω project, είναι η 3.2.12-3.4 η οποία διαδέχθηκε την 3.1.20 και στην περίπτωση μας εγκαταστάθηκε στο λειτουργικό σύστημα Ubuntu έκδοση 13.10 (Saucy Salamander).

Διαφορές ανάμεσα στις εκδόσεις είναι αρκετά χαρακτηριστικά τα οποία συνεχώς αναπτύσσονται και διατίθενται στο ευρύ κοινό.

2.3 Χαρακτηριστικά του Motion

Το Motion έχει κάποια βασικά χαρακτηριστικά (default) καθώς και κάποια πιο εξελιγμένα τα οποία μπορούν να προγραμματιστούν. Συγκεκριμένα, έχει τη δυνατότητα:

- Λήψης στιγμιότυπων κίνησης.
- Παρακολούθησης πολλαπλών συσκευών video ταυτόχρονα.
- Παρακολούθησης πολλαπλών εισόδων σε μια κάρτα διεπαφής ταυτόχρονα.
- Χρήση live streaming webcam.
- Αυτομάτου ελέγχου θορύβου και καταφλίων.
- Επιλογής και ρύθμισης του κειμένου που απεικονίζεται στις ληφθείσες εικόνες.
- Επιλογής και ρύθμισης των ονομάτων και των θέσεων των αρχείων των λαμβανόμενων εικόνων/στιγμιότυπων.
- Δημιουργίας mpreg αποσπασμάτων video σε πραγματικό χρόνο.
- Λήψης αυτοματοποιημένων στιγμιότυπων σε τακτά ή μη χρονικά διαστήματα.
- Εξαγωγής εξωτερικών προγραμμάτων κατά την ανίχνευση κίνησης ή κατά τη δημιουργία και λήξη των συμβάντων που έχουν οριστεί από τον χρήστη.
- Εξαγωγής των συμβάντων σε μια βάση δεδομένων MySQL ή PostgreSQL.
- Επανατροφοδότησης video προς ένα βρόγχο video4linux για απεικόνιση πραγματικού χρόνου.
- Ελέγχου μέσω ενός απλού web interface.

Επιπρόσθετα, μέρος των χαρακτηριστικών του αποτελούν και οι τεχνολογίες που περιγράφονται στις επόμενες παραγράφους:

- **Τεχνική ανίχνευσης κίνησης**

Το Motion ανιχνεύει κίνηση συγκρίνοντας την ένταση των pixels σε μια τρέχουσα νέα εικόνα σε σχέση με μια εικόνα αναφοράς, με μια παρόμοια διαδικασία όπως αυτή που περιγράφηκε στις παραγράφους 1.2 και 1.3, χωρίς να χρησιμοποιεί πληροφορία για τα χρώματα. Εάν δεν υπάρχει κίνηση ή θόρυβος, η παράμετρος νέα εικόνα-εικόνα αναφοράς (new_image-ref_image) θα πρέπει να είναι 0. Εάν, όμως, υπάρχει μια αλλαγή στην εικόνα το αποτέλεσμα θα είναι διαφορετικό. Για την αποφυγή αναγνώρισης του θορύβου ως κίνηση, η αλλαγή θα πρέπει να ξεπεράσει το κατώφλι (threshold) που έχει οριστεί καθώς και να υπάρχει ένας επαρκής αριθμός αλλαγμένων pixels πριν να αποφασιστεί από το λογισμικό ότι υπάρχει κίνηση.

Το καρέ αναφοράς ανανεώνεται αυτόματα και ταυτόχρονα σε σχέση με την τρέχουσα εικόνα, και όταν λοιπόν ανανεωθεί, αν υποθεθεί ότι η τρέχουσα εικόνα έχει την τιμή pictureN, η σύστασή του θα αποτελείται από τις:

$$1/2*pictureN + 1/4*pictureN-1 + 1/8*pictureN-2 \text{ κ.ο.κ.}$$

- **Locating motion**

Από το σύνολο των pixels που έχουν αλλάξει μεταξύ του καρέ αναφοράς και του νέου καρέ η μέση τιμή στους άξονες x και y υπολογίζεται για να χρησιμοποιηθεί ως το κέντρο της κίνησης. Έπειτα, υπολογίζεται η μέση απόσταση προς αυτό το σημείο. Στη συνέχεια σχηματίζεται ένα box ίσο με 2 φορές των αποστάσεων από του άξονες x και y γύρω από το κέντρο της κίνησης.

Όταν είναι ενεργοποιημένη η επιλογή labelling, το location box υπολογίζεται βάση όλων των επισημασμένων περιοχών, το μέγεθος των όποιων είναι παραπάνω από το καθορισμένο κατώφλι.

- **Motion masks**

Εάν υπάρχει ένα σημείο της εικόνας στο οποίο δεν χρειάζεται να γίνεται ανίχνευση κίνησης μπορεί να χρησιμοποιηθεί ένα φίλτρο mask ώστε να αποφευχθεί αυτή και να αποτρέψει την πιθανότητα ψευδούς συναγερμού. Το χαρακτηριστικό mask χρησιμοποιεί ένα φίλτρο για να κάνει επικάλυψη των pixels που μεταβάλλονται συνεχώς. Εάν το mask είναι λευκό, το pixel χρησιμοποιείται για ανίχνευση. Εάν είναι μαύρο δεν ανιχνεύεται καθόλου και δεν λαμβάνεται υπόψη. Με τη βοήθεια των grayscales μπορεί να γίνει επιλογή της ευαισθησίας της ανίχνευσης σε συγκεκριμένες περιοχές. Σε περίπτωση που το pixel είναι ανενεργό για αρκετό χρονικό διάστημα, το φίλτρο επαναφέρει προοδευτικά την ανίχνευση κίνησης και πάλι.

Σαν εργαλείο για το φίλτρο mask χρησιμοποιούνται αρχεία-εικόνες τύπου pgm. Αυτές οι εικόνες πρέπει να έχουν ακριβώς τις ίδιες διαστάσεις με τα καρέ που λαμβάνονται από την κάμερα καθώς και να έχουν δυαδική μορφή (binary format). Μπορούν να δημιουργηθούν σε ένα οποιοδήποτε πρόγραμμα επεξεργασίας και προβολής εικόνων. Η μονή μεταβολή θα είναι το λευκό χρώμα στις περιοχές όπου η ανίχνευση είναι επιθυμητή ενώ το μαύρο χρώμα στις περιοχές που δεν είναι επιθυμητό να ληφθούν υπόψη. Η ευαισθησία μεταβάλλεται ρυθμίζοντας ανάλογα την κλίμακα του γκρι. Η ιδιότητα του αρχείου mask είναι η απενεργοποίηση της ανίχνευσης σε συγκεκριμένα σημεία. Να σημειωθεί ότι στη ληφθείσα εικόνα, φαίνονται κανονικά όλες οι πληροφορίες και δεν αποτελεί τρόπο ώστε να αποκρύψει πληροφορία από την εικόνα που λαμβάνεται.

Παρακάτω (Εικόνα 2.3-1), παρατίθεται ένα παράδειγμα μιας εικόνας από webcam καθώς και το αρχείο mask file στο οποίο είναι επιθυμητή η αποφυγή ψευδών συναγερμών-ανιχνεύσεων, δηλαδή στο κομμάτι του δρόμου που βρίσκεται στο βάθος όπου είναι προφανές ότι θα δημιουργούνται συνεχώς ανιχνεύσεις από την κίνηση αυτοκινήτων. Συνεπώς, αφαιρώντας αυτό το σημείο θα έχουμε ανιχνεύσεις μόνο στο σημείο που μας ενδιαφέρει.



Εικόνα (2.3-1): Κανονική εικόνα με την ύπαρξη του δρόμου στο βάθος



Εικόνα (2.3-2): Αρχείο mask το οποίο έχει μετατραπεί σε png format

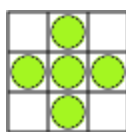
- **Despeckle**

Εναλλακτικά από τα αρχεία mask μπορεί να χρησιμοποιηθεί το φίλτρο Despeckle για την αποφυγή ψευδών συναγεμίων. Σε ένα οποιοδήποτε επιθυμητό μοτίβο παρακολούθησης και ειδικότερα σε εξωτερικούς χώρους ενδέχεται να υπάρχουν πολλά αντικείμενα τα οποία κινούνται συνεχώς όπως δέντρα, γρασίδι, θάλασσα και το Motion να τα ανιχνεύει. Για να αποφευχθεί η δημιουργία συνεχόμενων ανιχνεύσεων θα πρέπει να αυξηθεί ο βαθμός του threshold. Παράλληλα με την αύξηση του threshold, όμως, είναι πολύ πιθανό σαν αποτέλεσμα να χάνονται ανιχνεύσεις από σημεία όπου είναι επιθυμητό να υπάρχει ανίχνευση. Έτσι, λοιπόν, με τη χρήση του φίλτρου Despeckle είναι δυνατόν να απομακρυνθούν οι ανιχνεύσεις των μικρών αντικειμένων και να ανιχνευθεί μόνο η πραγματική κίνηση.

Φυσικά, για την προκειμένη περίπτωση θα μπορούσε να χρησιμοποιηθεί ένα αρχείο mask ώστε να μπλοκάρει τα κομμάτια της εικόνας όπου δεν είναι επιθυμητό

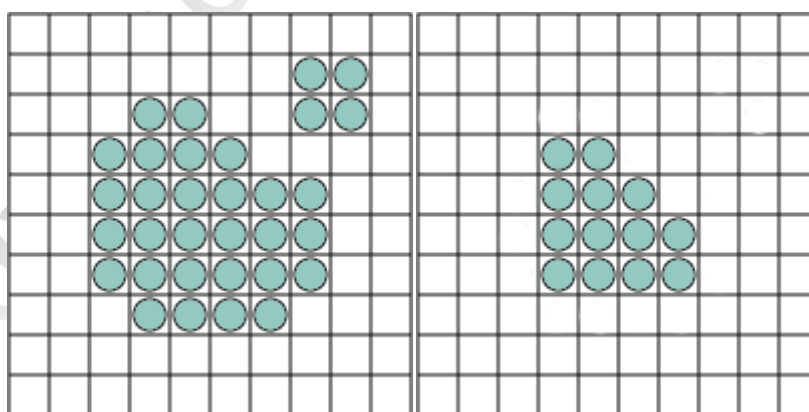
να υπάρχει ανίχνευση, αλλά όπως προτυπώθηκε το αρχείο mask θα πρέπει να είναι πλήρως ταυτισμένο με την λαμβάνουσα εικόνα. Αν για κάποιο λόγο εσκεμμένο ή μη, η κάμερα μετακινηθεί αυτομάτως το αρχείο αυτό αναιρείται. Αντιθέτως, με την επιλογή του despeckle patch, ο αλγόριθμος μπορεί να χρησιμοποιηθεί ανεξάρτητα με την αλλαγή θέσης της κάμερας και να αγνοήσει τις μικρές ανιχνεύσεις με τον τρόπο που προαναφέρθηκε.

Στις επόμενες παραγράφους, αναλύεται ο τρόπος λειτουργίας του φίλτρου despeckle. Αρχικά, θα πρέπει να οριστεί ένα δομικό στοιχείο (structural element), η μορφή του οποίου μπορεί να είναι οποιαδήποτε αλλά κατά την επιλογή του θα πρέπει να ληφθεί υπόψη ότι θα πρέπει να έχει σχήμα που να αντιστοιχεί στα χαρακτηριστικά του αντικειμένου που μας ενδιαφέρει. Αν για παράδειγμα, απαιτείται ανίχνευση λεπτών και μακριών αντικειμένων, αντίστοιχο σχήμα θα πρέπει να έχει και το structural element.



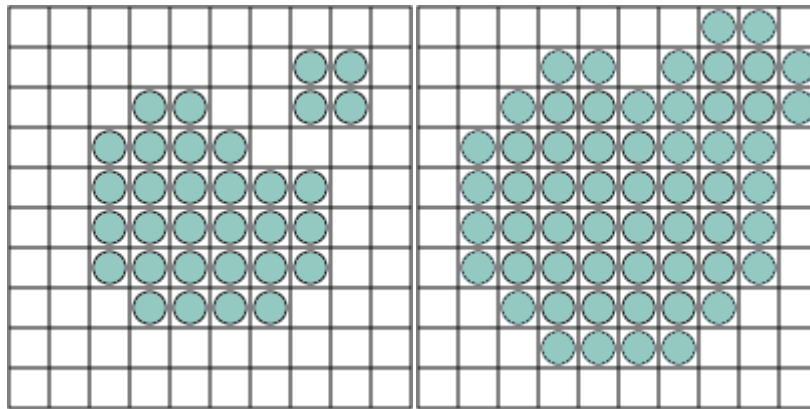
Εικόνα (2.3-3): Δημιουργία δομικού στοιχείου (structural element) σε μορφή σταυρού

Χρησιμοποιώντας το πράσινο structural element της Εικόνας 2.3-3, είναι δυνατή η σάρωση του σχήματος του σταυρού ανά την εικόνα. Εάν ανιχνευθούν 5 pixels στα οποία υπάρχει κίνηση στο ίδιο σχήμα τότε σημειώνεται σαν απλή ανίχνευση. Στην παρακάτω εικόνα (Εικόνα 2.3-4) είναι ορατό το αποτέλεσμα της εικασίας αυτής. Όσο κάθε σχήμα σταυρού που έχει ανιχνεύσει 5 pixels αντικαθιστάται από ένα pixel, έχει σαν αποτέλεσμα την μετατροπή των ανιχνεύσεων σε μικρότερες. Κανένα από τα 4 pixels στο 2x2 box στην πάνω δεξιά γωνία της εικόνας δεν ταιριάζει με το structural element το οποίο έχει σχήμα σταυρού και έτσι διαγράφεται. Η διαδικασία αυτή ονομάζεται Erosion (διάβρωση).



Εικόνα (2.3-4): Despeckle σε λειτουργία erosion χρησιμοποιώντας το structure element της Εικόνας 2.3-1

Η ακριβώς αντίστροφη διαδικασία επιτυγχάνεται με τη χρήση του Dilate (διαστολή). Με την εν λόγω διαδικασία γίνεται σάρωση με το structural element σε όλη την εικόνα και κάθε pixel ανίχνευσης που αντιστοιχεί στο σχήμα του σταυρού, μαρκάρεται. Το αποτέλεσμα είναι οι ανιχνεύσεις να γίνονται μεγαλύτερες.



Εικόνα (2.3-5): *Despeckle σε λειτουργία dilate χρησιμοποιώντας το structure element της Εικόνας 2.3-1*

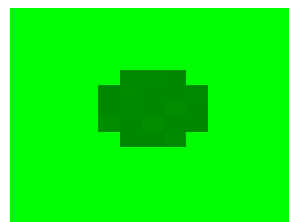
Το μέγεθος και το σχήμα του θορύβου ο οποίος δεν είναι επιθυμητός προς ανίχνευση ποικίλει ανάλογα με την κάμερα που χρησιμοποιείται ή το μοτίβο που παρακολουθείται, παρόλα αυτά είναι δυνατό να αναφερθεί ένας γενικός κανόνας για την μέθοδο του despeckle.

Παρακάτω, απεικονίζονται δυο διαφορετικά structural elements:



Εικόνα (2.3-6): *2 Διαφορετικά structural elements*

Να σημειωθεί ότι τα στοιχεία αυτά μπορούν να χρησιμοποιηθούν όσες φορές είναι αναγκαίο, μέχρι να εμφανιστεί το επιθυμητό αποτέλεσμα. Έτσι, λοιπόν, το κόκκινο τετράγωνο structural element συνοδευόμενο από τον πράσινο σταυρό ορίζεται από την επιλογή *despeckle Ee* και έχει σαν αποτέλεσμα το παρακάτω σχήμα ανίχνευσης:



Εικόνα (2.3-7): *Αποτέλεσμα ανίχνευσης έπειτα από την εφαρμογή των structural elements της Εικόνας 2.3-6*

Εάν γίνεται επαναλαμβανόμενα η εφαρμογή του σταυρού θα έχει σαν αποτέλεσμα την εξαγωγή ανιχνεύσεων σε σχήμα διαμαντιού, ενώ αν γίνεται εφαρμογή του box, τετραγώνου. Επίσης, μπορούν να χρησιμοποιηθούν διάφορες παραλλαγές με στόχο την δημιουργία κυκλικών σχημάτων. Κάθε φορά που χρησιμοποιείται ένα structural element, προστίθεται ή αφαιρείται αντίστοιχα ένα στρώμα pixels από την εξωτερική πλευρά όπου υπάρχει ανίχνευση. Συγκεκριμένα, κάθε φορά που επιλέγεται η erode (e/E) απομακρύνεται ένα στρώμα pixels γύρω από το κάθε σημείο ανίχνευσης, ενώ αντίθετα με την dilate (d/D) προστίθεται.

2.4 Εγκατάσταση και εκτέλεση του Motion

Απαραίτητη προϋπόθεση είναι να υπάρχει ήδη μια τουλάχιστον κάμερα video4linux (V4L) εγκατεστημένη στον υπολογιστή με ενημερωμένους drivers. Είναι γεγονός ότι μια κάμερα USB καταλαμβάνει μεγάλο bandwidth. Αν συνδεθεί σε μια υποδοχή USB 1.1 ή σε hub καταναλώνει όλο το bandwidth ακόμα και αν χρησιμοποιηθεί μικρό μέγεθος και χαμηλός ρυθμός frame οπότε είναι αδύνατο να υπάρξει περισσότερη της μιας κάμερας σε έναν ελεγκτή USB 1.1. Σε περίπτωση που το σύστημα διαθέτει ελεγκτή USB 2.0 είναι δυνατόν να συνυπάρξουν δύο ή ακόμα και τρεις κάμερες ρυθμισμένες, όμως, με αρκετά χαμηλό ρυθμό frame. Σε κάθε άλλη περίπτωση, θα πρέπει να προστεθούν επιπλέον USB PCI κάρτες στον υπολογιστή κάθε μία από τις οποίες θα πρέπει να έχει κοινό ελεγκτή για όλες τις εισόδους.

Έχοντας κάνει λήψη του Motion από το Motion Website για να το εγκαταστήσουμε εκτελούμε στο terminal την εξής εντολή:

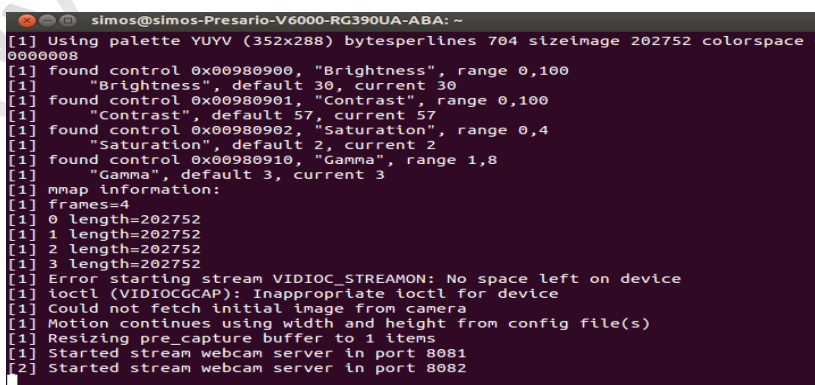
```
sudo apt-get install motion
```

Για να απενεργοποιηθεί η αυτόματη εκκίνηση του Motion κατά την εκκίνηση του λειτουργικού συστήματος, πληκτρολογούμε την παρακάτω εντολή:

```
update-rc.d -f motion remove
```

Με αυτόν τον τρόπο, οποιαδήποτε στιγμή ο χρήστης θελήσει να ενεργοποιήσει το Motion αρκεί μια απλή εντολή στο terminal με δικαιώματα superuser, η οποία είναι απλά το όνομα του προγράμματος, δηλαδή:

```
sudo motion
```



```
simos@simos-Presario-V6000-RG390UA-ABA: ~  
[1] Using palette YUYV (352x288) bytesperlines 704 sizeimage 202752 colorspace 0  
0000008  
[1] found control 0x00980900, "Brightness", range 0,100  
[1] "Brightness", default 30, current 30  
[1] found control 0x00980901, "Contrast", range 0,100  
[1] "Contrast", default 57, current 57  
[1] found control 0x00980902, "Saturation", range 0,4  
[1] "Saturation", default 2, current 2  
[1] found control 0x00980910, "Gamma", range 1,8  
[1] "Gamma", default 3, current 3  
[1] mmap information:  
[1] frames=4  
[1] 0 length=202752  
[1] 1 length=202752  
[1] 2 length=202752  
[1] 3 length=202752  
[1] Error starting stream VIDIOC_STREAMON: No space left on device  
[1] ioctl (VIDIOC_GCAP): Inappropriate ioctl for device  
[1] Could not fetch initial image from camera  
[1] Motion continues using width and height from config file(s)  
[1] Resizing pre_capture buffer to 1 items  
[1] Started stream webcam server in port 8081  
[2] Started stream webcam server in port 8082
```

Εικόνα (2.4-1): Έναρξη λογισμικού έπειτα από την κλήση της εντολής έναρξης

Έτσι, σε περίπτωση που δεν υπάρξει καμία άλλη ρύθμιση το σύστημα ανιχνεύει την κίνηση και η μόνη πληροφορία που είναι ορατή στον χρήστη είναι η εξαγωγή και ο προορισμός των αρχείων εικόνων.

```
[2] File of type 4 saved to: /var/www/imageviewer/images/01-20140407143934-04m.jp
pg
[2] File of type 4 saved to: /var/www/imageviewer/images/01-20140407143934-05m.jp
pg
[2] File of type 4 saved to: /var/www/imageviewer/images/01-20140407143934-06m.jp
pg
```

Εικόνα (2.4-2): Εξαγωγή αρχείων εικόνων έπειτα από ανίχνευση κίνησης

Θα ήταν χρήσιμο (όχι βέβαια απαραίτητο για την απλή εκτέλεση του προγράμματος) κατά την διάρκεια της εγκατάστασης να ληφθούν και να εγκατασταθούν οι πρόσθετες βιβλιοθήκες του Motion έτσι ώστε να γίνει εκμετάλλευση των πρόσθετων χαρακτηριστικών. Αν γίνει εγκατάσταση χρησιμοποιώντας προμεταγλωττισμένα binaries (Redhat type RPMs, Debian debs κτλ) υπάρχουν μόνο οι απαραίτητες βιβλιοθήκες και τα βασικά χαρακτηριστικά για την εκτέλεση του προγράμματος. Αν χρειαστεί να γίνει μεταγλώττιση από πηγή που χρησιμοποιεί τις προαναφερθείσες βιβλιοθήκες είναι αναγκαίο να εγκατασταθούν τα development packages. Η ονομασία αυτών των βιβλιοθηκών συνήθως είναι ίδια με αυτή του ονόματος του package ακολουθούμενη από την κατάληξη -devel ή -dev. Αυτά τα development packages εμπεριέχουν τα αρχεία κεφαλίδας (xxx.h) τα οποία χρειάζεται το Motion για να κάνει build τις βιβλιοθήκες.

Στον παρακάτω πίνακα (Πίνακας 2.4-1) απεικονίζονται οι βιβλιοθήκες που χρησιμοποιούνται από το Motion καθώς και τα packages που τις παρέχουν.

Βιβλιοθήκη	RPM Packages	Debian Packages
libm, libresolv, libdl, libpthread, libc, ld-linux, libcrypt, and libnsl	glibc and glibc-devel	libc6 , libc6-dev, libgl1.2
libjpeg	libjpeg and libjpeg-devel	libjpeg62 and libjpeg62- dev (optional libjpeg- mmx-dev)
libz	zlib and zlib-devel	zlib1g and zlib1g-dev

Πίνακας (2.4-1): Πρόσθετες βιβλιοθήκες και packages Motion

Για ενεργοποίηση και χρήση της **MySQL** είναι απαραίτητη η παρακάτω βιβλιοθήκη:

Βιβλιοθήκη	RPM Packages	Debian Packages
libmysqlclient	mysql and mysql-devel	libmysqlclient15-off and libmysqlclient15-dev

Πίνακας (2.4-2): Πρόσθετες βιβλιοθήκες και packages Motion για χρήση MySQL

Για ενεργοποίηση και χρήση της **PostgreSQL** είναι απαραίτητη η παρακάτω βιβλιοθήκη:

Βιβλιοθήκη	RPM Packages	Debian Packages
libpq	postgresql-libs and postgresql-devel	libpq-dev and libpq4

Πίνακας (2.4-3): Πρόσθετες βιβλιοθήκες και packages Motion για χρήση PostgreSQL

2.5 Motion configuration

Το configure είναι ένα script το οποίο εκτελείται για να γίνει έναρξη του περιβάλλοντος build για τον μεταγλωττιστή της C. Δημιουργεί και εξάγει το αρχείο "makefile" το οποίο το χρησιμοποιεί η συνάρτηση "make" για να μεταγλωττίσει και να εγκαταστήσει το πρόγραμμα.

Για να γίνει εκτέλεση του configure πρέπει να βρισκόμαστε στην ίδια διεύθυνση αρχείου που βρίσκεται και το Motion.

Δηλαδή, πληκτρολογώντας:

```
./configure
```

Επίσης, αν θέλουμε να δούμε το μενού βοήθειας, πληκτρολογούμε:

```
./configure --help
```

Στον παρακάτω πίνακα (2.5-1) απεικονίζονται όλες οι δυνατές επιλογές. Η ακριβή τους περιγραφή και λειτουργία είναι αναρτημένη στην κεντρική ιστοσελίδα του Motion (Motion Homepage). Στην παρούσα εργασία θα αναλυθούν όσες θα

χρησιμοποιηθούν ή όσες είναι απαραίτητες για την δημιουργία και επέκταση του project.

-h, --help	--libexecdir=DIR	--without-v4l
--help=short	--datadir=DIR	--with-jpeg-mmx=DIR
--help=recursive	--sysconfdir=DIR	--with-ffmpeg=DIR
-V, --version	--sharedstatedir=DIR	--without-ffmpeg
-q, --quiet, --silent	--localstatedir=DIR	--with-mysql-lib=DIR
--cache-file=FILE	--libdir=DIR	--with-mysql-include=DIR
-C, --config-cach	--includedir=DIR	--without-mysql
-n, --no-create	--oldincludedir=DIR	--without-pgsql
--srcdir=DIR	--infodir=DIR	--with-pgsql-include=DIR
--prefix=PREFIX	--mandir=DIR	--with-pgsql-lib=DIR
--exec-prefix=EPREFIX	--with-linuxthreads	--without-optimizcpu
--bindir=DIR	--with-pwcbbsd	--with-developer-flags
--sbindir=DIR	--without-bktr	

Πίνακας (2.5-1): Εντολές configuration script

2.5.1 Configuration αρχεία Motion

Το όνομα των Configuration αρχείων πρέπει να είναι "motion.conf". Όταν καλείται το Motion, ψάχνει για τα αρχεία configuration με ονομασία "motion.conf" στην τοποθεσία ".motion" που βρίσκονται στην home διεύθυνση του χρήστη. Για παράδειγμα, αν η μεταβλητή περιβάλλοντος είναι \$HOME και η ονομασία του χρήστη simos τα configuration αρχεία θα βρίσκονται στη θέση:

/home/simos/.motion/motion.conf

Ο ορισμός της διεύθυνσης της θέσης του αρχείου γίνεται μέσω της εντολής sysconfdir=DIR κατά τη διάρκεια της εκτέλεσης της configure όταν εγκαθίσταται το Motion. Εάν αυτή η επιλογή δεν έχει εκτελεστεί η προεπιλεγμένη θέση του αρχείου

θα είναι /usr/local/etc/. Συνίσταται να υπάρχει μόνο ένα configuration αρχείο "motion.conf" στην προεπιλεγμένη θέση /usr/local/etc/.

Το Motion έχει ένα configuration αρχείο στο κατανεμημένο package το οποίο ονομάζεται "motion-dist.conf" και τα αρχεία του αντιγράφονται στη θέση /usr/local/etc κάθε φορά που εκτελείται η εντολή "make install". Το αρχείο αυτό θα πρέπει να μετονομαστεί από "motion-dist.conf" σε "motion.conf" οπότε σε πιθανή μετεγκατάσταση ή αναβάθμιση του Motion να υπάρχει το γνήσιο αρχείο "motion-dist.conf" και το "motion.conf" να μην μεταβληθεί ή υπερκαλυφθεί.

Το Motion αναγνωρίζει τις παραμέτρους για το configuration λαμβάνοντας υπόψη τους εξής κανόνες:

- Διαβάζει το αρχείο configuration motion.conf από την αρχή προς το τέλος γραμμή προς γραμμή.
- Αν η ίδια παράμετρος υπάρχει σε περισσότερα από ένα σημεία, λαμβάνεται υπόψη αυτή που διαβάζεται από το τελευταίο σημείο.
- Εάν δεν έχουν εγκατασταθεί πρόσθετα χαρακτηριστικά όπως ffmpeg, mysql κτλ θα αγνοηθούν οι επιλογές που αντιστοιχούν σε αυτά χωρίς να χρειάζεται να σβηστούν ή να γίνουν σχόλιο στον κώδικα.

2.5.2 Επιλογές configuration στη γραμμή εντολών

-a seconds	-h	-p
-C	-i input	-Q
-c changes	-L noise	-q quality
-D	-l	-S number
-d device	-M address	-s widthxheight
-E command	-m	-t target-dir
-F file	-N	-U url or IP_addr
-f number	-n norm	-u user:pass
-G seconds	-O command	-V device
-g seconds	-P device	-w

Πίνακας (2.5.2-1): Επιλογές configuration στη γραμμή εντολών

2.5.3 Επιλογές αρχείου configuration

Στον επόμενο πίνακα (Πίνακας 2.5.3-1) απεικονίζονται όλες οι δυνατές επιλογές για ρύθμιση του αρχείου configuration σύμφωνα με τις προτιμήσεις του εκάστοτε χρήστη. Αξίζει να αναφερθεί ότι όλες οι επιλογές που ενδεχομένως να γίνουν στο αρχείο μπορούν να μη ληφθούν υπόψη σε περίπτωση που χρησιμοποιηθούν οι αντίστοιχες της γραμμής εντολών.

Όλες οι τιμές των μεταβλητών είναι ακέραιοι αριθμοί και δεν επιτρέπονται δεκαδικοί. Οι λογικοί τελεστές των επιλογών μπορεί να είναι ορισμένοι είτε σε λειτουργία "on" είτε σε "off" , "1" ή "0" και "yes" ή "no".

always_changes on/off	lightswitch percent	pgsql_db	text_left TEXT
auto_brightness on/off	locate on/off	pgsql_host	text_right TEXT
brightness	low_cpu framerate	pgsql_user	thread filename
Contrast	mail address	pgsql_password	threshold value
control_localhost on/off	mask_file file	pgsql_port	threshold_tune on/off
control_port port_no	max_mpeg_time seconds	post_capture number	timelapse_filename value
Daemon on/off	minimum_gap seconds	ppm on/off	track_type number
despeckle EedDI (and other combinations of E, e, d and D and optionally ending with I).	minimum_motion_frames number	pre_capture number	track_iomoyo_id
debug_parameter integer	motion_video_pipe devicename/-	quality value	track_maxx
execute command	mysql_db	quiet on/off	track_motorx
ffmpeg_bps bps	mysql_host	rotate degrees	track_port
ffmpeg_cap_new on/off	mysql_user	roundrobin_frames number	track_speed
ffmpeg_cap_motion on/off	mysql_password	roundrobin_skip number	track_stepsize
ffmpeg_filename value	netcam_url URL	saturation	tunerdevice device_name

ffmpeg_timelapse_mode	netcam_userpass user:pass	smart_mask_speed	videodevice device_name
ffmpeg_timelapse seconds	night_compensate on/off	sms number	video_pipe device_name/-
ffmpeg_variable_bitrate value	noise_level level	snapshot_filename value	webcam_limit on/off
ffmpeg_video_codec codec_name	noise_tune on/off	snapshot_interval seconds	webcam_localhost on/off
framerate number	norm 0/1/2/3	sql_log_image on/off	webcam_maxrate rate
frequency value	onffmpegclose command	sql_log_snapshot on/off	webcam_motion on/off
gap seconds	onmpeg command	sql_log_mpeg on/off	webcam_port port_no
height pixels	onsave command	sql_log_timelapse on/off	webcam_quality level
hue	output_all on/off	switchfilter on/off	width pixels
input number	output_motion on/off	target_dir directory_path	
jpeg_filename value	output_normal on/off/first	text_changes on/off	

Πίνακας (2.5.3-1): Επιλογές για τη ρύθμιση του αρχείου configuration

2.6 Μεταγλώττιση του Motion

Όταν εκτελεστεί η make όλα τα αρχεία της C μεταγλωττίζονται αυτόματα και ενοποιούνται. Η make χρησιμοποιεί το αρχείο "makefile" το οποίο δημιουργείται και εξάγεται από την εκτέλεση του configure script όπως προαναφέρθηκε στην αρχή της παραγράφου 2.5. Κάθε φορά που εκτελείται το Configure Script δημιουργείται και ένα νέο αρχείο "makefile" και προφανώς οι όποιες μετατροπές έχουν γίνει στο προηγούμενο "makefile" χάνονται.

Εάν έχει γίνει εκτέλεση της εντολής make πρωτίστως και είναι επιθυμητή η επανεκτέλεσή της, θα πρέπει να γίνει πρώτα εκτέλεση της make clean έτσι ώστε να αναιρεθούν όλα τα αρχεία που δημιουργήθηκαν την πρώτη φορά της εκτέλεσης της make. Είναι ευνόητο ότι αν δεν γίνει αυτό δεν θα υπάρχει πρόσβαση στα πρόσθετα χαρακτηριστικά τα οποία θα έχουν εγκατασταθεί μεταγενέστερα. Για παράδειγμα, εάν εκτελεστεί η make χωρίς το πρόσθετο χαρακτηριστικό που υποστηρίζει ffmpeg

και αυτό εγκατασταθεί αργότερα θα πρέπει να γίνει πρώτα εκτέλεση της `make clean` και έπειτα της `make`, ειδικά το `ffmpeg` δεν θα ενεργοποιηθεί.

Πιο πρακτικά, την πρώτη φορά που γίνεται μεταγλώττιση του Motion θα πρέπει να εκτελεστεί:

```
./configure, make, make install
```

Και αν για οποιονδήποτε λόγο αλλάξει κάτι στο Configuration θα πρέπει να εκτελεστεί:

```
./configure, make clean, make, make install
```

Στη συνέχεια αναλύονται όλες οι δυνατές επιλογές της εντολής `make`.

• **Make Install**

Με αυτή την εντολή γίνεται απλά αντιγραφή των στοιχείων που δημιουργήθηκαν και εξάχθηκαν κατά την μεταγλώττιση και ενοποίηση που έγινε με την `make` δημιουργώντας τις θέσεις των αρχείων αν αυτές δεν υπάρχουν ήδη.

Οι προεπιλεγμένες θέσεις των αρχείων είναι:

```
/usr/local/bin, usr/local/man/man1, /usr/local/etc, /usr/local/share/doc/motion-3.2.X
```

και

```
/usr/local/share/doc/examples/motion-3.2.X
```

Έτσι, αντιγράφει τα παρακάτω αρχεία από την βασική θέση αρχείων του Motion με την προϋπόθεση ότι χρησιμοποιήθηκε η προεπιλεγμένη `/usr/local` όταν εκτελέστηκε το Configuration Script. Αν κατά τη διάρκεια εκτέλεσης του Configuration Script η προεπιλεγμένη `/usr/local` αλλάχθηκε, είναι προφανές ότι τα αρχεία θα μετακινηθούν στην αλλαγμένη θέση αρχείου. Αναλυτικότερα, δημιουργούνται τα παρακάτω αρχεία στις εξής διευθύνσεις:

- Το αρχείο Configuration "motion-dist.conf" στη θέση `/usr/local/etc`
- Το εκτελέσιμο binary αρχείο "motion" στη θέση `/usr/local/bin`
- Η σελίδα Manual "motion.1" στη θέση `/usr/local/man/man1`
- Τα αρχεία Documentation "Changelog, Copying, Credits, Install και Readme" στη θέση `/usr/local/share/doc/motion-3.2.X`
- Τα παραδείγματα αρχείων configuration "*.conf" στη θέση `/usr/local/share/doc /examples/motion-3.2.X`

• **Make Uninstall**

Με τη χρήση της γίνεται διαγραφή των αρχείων που βρίσκονται στην βασική θέση `/usr/local` που δημιουργήθηκαν κατά την Make Install και οποιονδήποτε σχετίζονται με αυτά.

- **Make Clean**

Όπως προτυπώθηκε, διαγράφει όλα τα αρχεία που δημιουργήθηκαν από την εκτέλεση της make και είναι απαραίτητο να εκτελείται κάθε φορά που αλλάζει οτιδήποτε στο configuration έτσι ώστε αυτό να αποκτήσει ισχύ. Τέλος, διαγράφει όλα τα προσωρινά αρχεία καθώς και αρχεία μορφής jpeg που έχουν δημιουργηθεί στη βασική θέση αρχείων του Motion.

- **Make DistClean**

Κατά την εκτέλεσή της διαγράφει τα αρχεία: config.status, config.log, config.cache, Makefile, και motion.spec.

- **Make Dist**

Με αυτήν την εντολή γίνεται ταυτόχρονα εκτέλεση των make clean και make distclean.

2.7 Πηγαία αρχεία Motion

Συνολικά υπάρχουν 45 αρχεία (αρχεία C και κεφαλίδες):

alg.c, alg.h, conf.c, conf.h, draw.c, event.c, event.h, ffmpeg.c, ffmpeg.h, jpegutils.c, jpegutils.h, logger.c, logger.h, md5.c, md5.h, mmx.h, motion.c, motion.hnetcam.c, netcam.h, netcam_ftp.c, netcam_ftp.h, netcam_jpeg.c, netcam_wget.c, netcam_wget.h, picture.c, picture.h, pwc-ioctl.h, rotate.c, rotate.h, sdl.c, sdl.h, stream.c, stream.h, track.c, track.h, video.c, video.h, video2.c, video_common.c, video_freebsd.c, video_freebsd.h, vloopback_motion.c, vloopback_motion.h, webhttpd.c, webhttpd.h.

Τα σημαντικότερα από αυτά είναι το motion.c ,που είναι και η καρδιά του Motion, το alg.c που εμπεριέχει όλους τους αλγορίθμους του λογισμικού και το conf.c που είναι αρμόδιο για την κατασκευή του αρχείου configuration.

Στις επόμενες παραγράφους αναφέρονται οι αλγόριθμοι που ορίζουν το λογισμικό στο αρχείο alg.c και ο οποίοι έχουν τις παρακάτω ιδιότητες και χαρακτηριστικά:

- Αλγόριθμος alg_diff_standard δημιουργεί τη στάνταρ συνάρτηση του αλγορίθμου διαφοράς διαδοχικών καρέ.
- Αλγόριθμος alg_diff_fast δημιουργεί μια πολύ γρήγορη συνάρτηση του αλγορίθμου διαφοράς διαδοχικών καρέ αλλά δεν έχει τη δυνατότητα mask overlaying.
- Αλγόριθμος alg_diff χρησιμοποιεί τον αλγόριθμο diff_fast ώστε να αποφασιστεί γρήγορα εάν εντοπίζει κάτι και αν γίνει αυτό καλεί τον alg_diff_standard.

- Αλγόριθμος `alg_update_reference_frame` καλείται από το Βρόγχο Motion ώστε να υπολογίσει το καρέ αναφοράς. Τα κινούμενα αντικείμενα εξαιρούνται από το συγκεκριμένο καρέ για κάποιο χρονικό διάστημα ώστε να βελτιωθεί η ποιότητα της ανίχνευσης. Συντελείται από τις παραμέτρους `cnt -current thread's context struct` και `action - UPDATE_REF_FRAME or RESET_REF_FRAME`.
- Αλγόριθμος `alg_switchfilter` αφανίζει το θόρυβο που δημιουργείται κατά τη λειτουργία `roundrobin` των καμερών.
- Αλγόριθμος `alg_tune_smartmask` δημιουργεί την πραγματική `smartmask` και υπολογίζει την ευαισθησία που βασίζεται η κίνηση.
- Αλγόριθμος `alg_noise_tune` ελέγχει την ευαισθησία ως προς τη στάθμη του θορύβου στον προς μελέτη χώρο.
- Αλγόριθμος `alg_threshold_tune` ελέγχει την ευαισθησία ως προς την κατωφλίωση των αλλαγμένων καρέ.
- Αλγόριθμος `alg_despeckle` χρησιμοποιεί τη διαδικασία του `despeckle` ώστε να απομακρύνει τις ανιχνεύσεις που περιλαμβάνουν θόρυβο.
- Αλγόριθμος `dilate5` χρησιμοποιεί την ιδιότητα `dilate` του `despeckle` σε `a + shape`.
- Αλγόριθμος `erode9` χρησιμοποιεί την ιδιότητα `erode` του `despeckle` σε `3x3 box`.
- Αλγόριθμος `erode5` χρησιμοποιεί την ιδιότητα `erode` του `despeckle` σε `a + shape`.
- Αλγόριθμος `alg_labeling` απεικονίζει πληροφορίες και τις συνθήκες κάτω από τις οποίες εντοπίστηκε κίνηση σε μορφή `text`.
- Αλγόριθμος `alg_locate_center_size` εντοπίζει το κέντρο και το μέγεθος της κίνησης.
- Αλγόριθμος `alg_draw_location` σχηματίζει ένα ορθογώνιο κουτί γύρω από το σημείο όπου έχει εντοπιστεί κίνηση.
- Αλγόριθμος `alg_draw_red_location` σχηματίζει ένα κόκκινο ορθογώνιο κουτί γύρω από το σημείο όπου έχει εντοπιστεί κίνηση.

Βρόχος Motion (Motion Loop)

Επιπρόσθετα, ακολουθεί αναφορά στον τρόπο με τον οποίο γίνεται η ανίχνευση της κίνησης από το αρχείο `motion.c` και συγκεκριμένα στο κομμάτι όπου βρίσκεται ο βρόγχος Motion (Motion Loop). Όπως προτυπώθηκε, είναι η καρδιά του

Motion και αποτελείται από τα παρακάτω στάδια τα οποία αναλύονται στις επόμενες παραγράφους:

- Προετοιμασία για νέο καρέ (frame)
- Επαναπροσπάθεια αρχικοποίησης του τομέα Netcam
- Τμήμα σύλληψης εικόνας
- Τμήμα ανίχνευσης κίνησης
- Τμήμα συντονισμού
- Τμήμα επικάλυψης κειμένου και γραφικών
- Τμήμα ελέγχου ενεργειών και διαχείρισης γεγονότων
- Τμήμα frame αναφοράς
- Τμήμα εξόδου της τερματικής κονσόλας σε λειτουργία Setup Mode
- Τμήμα χαρακτηριστικών των στιγμιότυπων Snapshot
- Τμήμα χαρακτηριστικών λήξης Timelapse
- Τμήμα επανάληψης Video Loopback
- Τμήμα ενημέρωσης παραμέτρων ανά δευτερόλεπτο
- Τμήμα ορισμού καρέ για το χρονοδιάγραμμα ύπνου sleeping mode

Αναλυτικότερα, τα επιμέρους στάδια αποτελούνται από τις παρακάτω διαδικασίες:

Προετοιμασία για νέο frame

- Αποθήκευση χρόνου για τον υπολογισμό του ρυθμού των καρέ (framerate).
- Αν παραστεί αναγκαίο, αλλαγή μεγέθους του ring buffer των καρέ.
- Άντληση χρόνου για το τρέχον καρέ (time_t cnt->currenttimep και struct tm cnt->currenttime).
- Επαναφορά των *shots* σε -1 εάν υπάρχει νέο δευτερόλεπτο και μείωση του μετρητή **minimum_frame_time_downcounter**.
- Αύξηση των *shots* (αριθμός καρέ μέσα στο τρέχον δευτερόλεπτο).
- Εάν ο μετρητής **minimum_frame_time_downcounter** είναι μηδέν, ορισμός του σε **minimum_frame_time** και συνέχεια από το σημείο "ΕΠΑΝΕΙΣΟΛΟΣ ΣΤΟΝ ΒΡΟΓΧΟ".
- Αποθήκευση της χρονοσφραγίδας στον ring buffer .
- Αποθήκευση των *shots* στον ring buffer.
- Ρύθμιση των δεικτών του ring buffer (αν βρίσκεται στο τέλος του buffer οδηγείται στην αρχή).

Επαναπροσπάθεια αρχικοποίησης του τομέα Netcam

Η μεταβλητή cnt->video_dev ορίζεται κατά τη διάρκεια αρχικοποίησης της κάμερας όταν γίνεται η σύνδεση για πρώτη φορά. Αν πρόκειται για netcam κάμερα, χρησιμοποιείται η cnt->video_dev σαν flag ώστε να υποδεικνύει ότι η κάμερα δεν είναι ακόμα διαθέσιμη (τιμή -1) και θα πρέπει να γίνει επαναπροσπάθεια αρχικοποίησης της έως ότου γίνει διαθέσιμη.

- Εάν η video_dev είναι -1 (δεν έχει γίνει αρχικοποίηση) είναι netcam και το πρώτο καρέ λαμβάνεται κάθε δέκατο δευτερόλεπτο:
 - Κλήση της netcam_cleanup. Η παράμετρος 1 των δευτερολέπτων ενημερώνει την netcam_cleanup ότι ο τρόπος λειτουργίας δεν είναι επανεκκίνηση ή έξοδος. Η netcam_cleanup απελευθερώνει όλα τα

structs και τους buffers που δημιουργεί η netcam_start επιτρέποντας έτσι να γίνει κλήση της vid_start ξανά.

- Κλήση της vid_start. Καταχωρεί την επιστρεφόμενη τιμή στην cnt->video_dev. Εάν η τιμή είναι -1 τότε η netcam δεν είναι ακόμη διαθέσιμη. Όταν, τελικά, η netcam αποκριθεί η cnt->video_dev γίνεται 0.
- Επανεκκίνηση του Motion όταν η netcam επιστρέψει μια εικόνα με διαφορετικές διαστάσεις από αυτές που έχουν δοθεί στο αρχείο motion.conf, ώστε να γίνει επαναρχικοποίηση όλων των buffers.

Τμήμα σύλληψης εικόνας

- Λήψη του επόμενου καρέ από την κάμερα (vid_next)
 - Αν είναι netcam κάμερα, κλήση του netcam_next.
 - Εάν επιτευχθεί η λήψη ενός καρέ από τη netcam επιστροφή του καρέ και ορισμός της μεταβλητής vid_return_code=0 (μήνυμα success).
 - Εάν αποτύχει επιστροφή ενός κωδικού για αποτυχία. -2 για fatal, >2 για non-fatal.
 - Αν το error οφείλεται στο γεγονός ότι η netcam έχει αλλάξει διαστάσεις, επιστροφή της τιμής NETCAM_RESTART_ERROR.
 - Αν είναι V4l κάμερα:
 - Κλήση της v4l_set_input για αρχικοποίηση της συσκευής.
 - Λήψη εικόνας μέσω της v4l_next.
 - Αν η λήψη της εικόνας αποτύχει επιστροφή της vid_return_code=-1 (fatal).
 - Αν η λήψη της εικόνας επιτύχει επιστροφή της vid_return_code=0 (success).
- Αν η μεταβλητή vid_return_code είναι 0 (success) :
 - Επαναφορά-μηδενισμός της missing_frame_counter
 - Αποθήκευση της νέας εικόνας στην cnt->imgs.image_virgin η οποία θα παραμείνει απaráλλακτη.
 - Αν είναι netcam κάμερα, επαναρχικοποίηση του χρόνου στον τρέχοντα χρόνο έτσι ώστε να γίνει η κάμερα αυτή η οποία ορίζει το ελάχιστο βήμα και όχι το Motion έτσι ώστε να αποφευχθούν διπλά καρέ.
- Αν η μεταβλητή vid_return_code δεν είναι 0 :
 - Εάν είναι vid_return_code < 0 (fatal) έξοδος από το thread
 - Εάν είναι vid_return_code > 0 (non-fatal)
 - Αν η vid_return_code = NETCAM_RESTART_ERROR (αλλαγή των διαστάσεων της netcam) επανεκκίνηση του Motion έτσι ώστε να γίνει αλλαγή διαστάσεων όλων των buffers με ελεγχόμενο τρόπο.
 - Για το πρώτο χαμένο καρέ (cnt->missing_frame_counter=0) αποθήκευση της χρονοσφραγίδας στη cnt->connectionlosttime.
 - Αν υπάρχει αναμονή για αρχικοποίηση της κάμερας, ορισμός της cnt->missing_frame_counter = 1 ώστε να αποφευχθεί την επόμενη φορά η ανανέωση της cnt->connectionlosttime.

- Αν η netcam έχει ήδη αρχικοποιηθεί και η σύνδεση χάθηκε, αντιγραφή της προηγούμενης εικόνας στην τρέχουσα εικόνα τα πρώτα 30 δευτερόλεπτα.

Τμήμα ανίχνευσης κίνησης

- Ανίχνευση κίνησης
 - Χρήση του standard αλγορίθμου διαφοράς εάν ανιχνεύεται κίνηση ή βρισκόμαστε σε setup mode, διαφορετικά χρήση του fast αλγορίθμου διαφοράς. Εάν ο fast αλγορίθμος εντοπίζει πολλά αλλαγμένα pixels, χρήση του normal αλγορίθμου.
 - Ρύθμιση της παραμέτρου Nightcompensate αν είναι ενεργοποιημένη για αντιστάθμιση του θορύβου.
 - Αρχικοποίηση των pixels του frame κίνησης (cnt->imgs.out) στο μαύρο χρώμα (YUV420P, Y=0, UV=0x80).
 - Για κάθε pixel εφαρμογή mask και smartmask .
 - Τα Pixels που περιλαμβάνουν ανίχνευση κίνησης αντιγράφονται στην εικόνα Motion (cnt->imgs.out) (Y channel only = B/W).
 - Αύξηση των diffs αν τα pixels μετά την διαδικασία masking έχουν αλλάξει περισσότερο από τις ρυθμίσεις για το θόρυβο.
 - Επιστροφή των diffes.
 - Lightswitch (ρύθμιση των diffs στο 0 αν υπάρχουν πολλά αλλαγμένα pixels).
 - Switchfilter (προσπάθεια να αγνοηθούν οι εικόνες που περιέχουν θόρυβο όταν γίνεται αλλαγή της κάμερας σε round robin).
 - Despeckle
 - Ρύθμιση των diffs σύμφωνα με τις παραμέτρους Erode και Dilate.
 - Η εικόνα Motion (cnt->imgs.out) ρυθμίζεται σύμφωνα με τα αφαιρούμενα και τα γειτονικά pixels.
 - Ρύθμιση των diffs που περιλαμβάνουν Labeling.
 - Τα αποτελέσματα της ανίχνευσης κίνησης σε έναν αριθμό diffs.
- Ρύθμιση του smartmask.
- Αν είναι ενεργοποιημένη η επιλογή tracking και υπάρχει κίνηση της κάμερας (move counter > 0).
 - Μείωση του μετρητή της κίνησης (move counter).
 - Ορισμός diff=0 για να αποφευχθεί ανίχνευση.

Τμήμα συντονισμού

- Αν είναι ενεργοποιημένη η παράμετρος noise tune, ρύθμιση του θορύβου κάθε δευτερόλεπτο που δεν ανιχνεύεται κίνηση.
- Αν δεν είναι ενεργοποιημένη η παράμετρος noise tune, ορισμός του επιπέδου του θορύβου σύμφωνα με αυτό που έχει οριστεί από το αρχείο configuration.
- Αν είναι ενεργοποιημένη η παράμετρος threshold tune, ρύθμιση του threshold διαφορετικά ορισμός του με βάση την τιμή που έχει λάβει από το αρχείο configuration.

Τμήμα επικάλυψης κειμένου και γραφικών

- Επικάλυψη του καθορισμένου mask με κόκκινο στο frame κίνησης (cnt->imgs.out).
- Επικάλυψη του smartmask με κόκκινο στο frame κίνησης.
- Επικάλυψη του μεγαλύτερου label με μπλε στο frame κίνησης.
- Επικάλυψη του κέντρου και του μεγέθους της κίνησης αν υπάρχει ανίχνευση.
 - Υπολογισμός του γεωμετρικού μέσου των labels μεγαλύτερο από το threshold ή από όλα τα pixels που έχουν ανιχνευθεί εάν η παράμετρος labeling δεν είναι ενεργοποιημένη.
 - Υπολογισμός του γεωμετρικού μέσου όλων των pixels από τα αριστερά και τα δεξιά του κέντρου x καθώς και πάνω και κάτω από το κέντρο y.
 - Υπολογισμός του location box σαν 2 φορές των υπολογισμένων μέσων εκτός των άνω y συντεταγμένων οι οποίες πολλαπλασιάζονται με το 3 επειδή το box περιλαμβάνει και το ανθρώπινο κεφάλι.
 - Ρύθμιση του κέντρου y στο box που περιλαμβάνει το ανθρώπινο κεφάλι.
- Αρχικοποίηση των χαρακτήρων διπλού μεγέθους αν είναι αναγκαίο.
- Αν η παράμετρος text_double είναι απενεργοποιημένη, τότε γίνεται επαναφορά της scaling text_size_factor αν είναι αναγκαίο.
- Προσθήκη του κειμένου εικόνας και diffs στην άνω δεξιά γωνία.
- Αν βρίσκεται σε setup mode προσθήκη της πληροφορίας αυτής στην εικόνα.
- Προσθήκη text_left στα αριστερά της εικόνας.
- Προσθήκη text_right στα δεξιά της εικόνας.

Τμήμα ελέγχου ενεργειών και διαχείρισης γεγονότων

- Αν η παράμετρος output_all είναι ενεργοποιημένη ενέργεια σαν να υπάρχει ανίχνευση κίνησης και έναρξη του motion_detected (η διαδικασία για το βήμα motion_detected περιγράφεται παρακάτω).
- Διαφορετικά αν $\text{diffs} > \text{threshold}$
 - Αν είναι το πρώτο γεγονός ανίχνευσης ($\text{detecting_motion} = 0$) έναρξη μέτρησης από το 1.
 - Αύξηση της παραμέτρου detecting_motion (detecting_motion είναι ο αριθμός των καρέ στα οποία έχει εντοπιστεί κίνηση + 1).
 - Αν ο αριθμός των καρέ που περιλαμβάνουν κίνηση είναι τουλάχιστον ίσος με την παράμετρο minimum_motion_frames ($\text{detecting_motion} > \text{cnt->conf.minimum_motion_frames}$) κλήση της `motion_detected()`.
 - **motion_detected()**
 - Αν υπάρχει νέο γεγονός ($\text{cnt->event_nr} \neq \text{cnt->prev_event}$).
 - Επαναφορά του αριθμού της παραμέτρου prev_event σύμφωνα με το τρέχον γεγονός και αποθήκευση του χρόνου του συμβάντος.
 - Δημιουργία της event_text_string η οποία χρησιμοποιείται για τον προσδιορισμό της μετατροπής %C.

- EVENT_FIRST_MOTION (εξαγωγή της on_event_start εντολής και εκτέλεση της on_event_ffmpeg_newfile).
 - Διαδικασία σύλληψης καρτέ pre_capture (EVENT_IMAGE_DETECTED (αποθήκευση jpgs, πρόσθεση στα mpeg) για κάθε εικόνα.
 - Αν είναι ενεργοποιημένη η επιλογή output_normal=first ορισμός της cnt->preview_shot flag = 1 έτσι ώστε η event_image_detect να αποθηκεύσει την εικόνα.
- Αν δεν έχει γίνει κλήση της motion_detected κατά τη διάρκεια της post_capture ή της output_all (diffs > 0)
 - Σχεδιασμός του locate frame (frame γύρω από την κίνηση) αν η παράμετρος locate είναι ενεργοποιημένη.
 - EVENT_MOTION
 - δημιουργία ήχου εάν είναι ενεργοποιημένος.
 - εντολή on_motion_detected.
- Έλεγχος για τα πιο σημαντικά preview-shot όταν η επιλογή output_normal είναι ορισμένη στο best.
- EVENT_IMAGE_DETECTED (αποθήκευση jpgs, πρόσθεση στα mpeg) για την τρέχουσα εικόνα (μόνο εάν δεν υπάρχει υπέρβαση στον ρυθμό των καρτέ και δεν υπάρχει αναμονή για να περάσει η παράμετρος minimum_gap).
- Τροφοδότηση της webcam εάν η επιλογή webcam_motion είναι ενεργοποιημένη και δεν είναι το πρώτο shot σε αυτό το δευτερόλεπτο.
- Καθαρισμός της παραμέτρου cnt->preview_shot (=0) από τη στιγμή που έχει σωθεί η πρώτη εικόνα του γεγονότος.
 - Διαδικασία tracking εάν είναι ενεργοποιημένη.
- Επαναφορά του μετρητή post_capture (postcap).
- Διαφορετικά, αν δεν έχει υπάρξει ανίχνευση κίνησης αλλά ο μετρητής post_capture /post_cap δεν είναι 0 συνεχίζεται η κλήση της motion_detected().
- Ειδικά, επαναφορά του μετρητή detecting_motion στην τιμή 0 ώστε να είναι έτοιμος για το επόμενο γεγονός.
- Εάν το τρέχον γεγονός διαρκεί για μεγάλο χρονικό διάστημα (το οποίο ορίζεται από την max_mpeg_time) τερματισμός και έναρξη ενός νέου θέτοντας την cnt->makemovie=1.
- Τερματισμός του γεγονότος εάν είναι σε κατάσταση ηρεμίας μεγαλύτερο χρόνο από ότι το κενό (gap) ή αν το γεγονός διαρκεί περισσότερο από τα παραπάνω.
 - Ενεργοποιημένη η Preview_best of.
 - EVENT_ENDMOTION (event_ffmpeg_closefile).
 - Εάν είναι σε λειτουργία tracking μετακίνηση της κάμερας στο κέντρο.
 - Αύξηση του αριθμού γεγονότος.

- Απενεργοποίηση της παραμέτρου `text_event_string` ώστε να αποφευχθεί η λήψη χρονοσφραγίδας από το προηγούμενος γεγονός στις τρέχουσες εικόνες που βρίσκονται στο `buffer`.

Τμήμα καρέ αναφοράς

- Αν υπάρχει κίνηση της κάμερας ή ξαφνική μεταβολή της έντασης του φωτός, πρόληψη των λανθασμένων ανιχνεύσεων με τη δημιουργία ενός καινούριου καρέ αναφοράς.
- Διαφορετικά, ανανέωση του καρέ αναφοράς.

Τμήμα εξόδου της τερματικής κονσόλας σε λειτουργία Setup Mode

- Αν βρίσκεται σε τρόπο λειτουργίας `setup`, εξαγωγή αριθμών `setup numbers` στην τερματική κονσόλα.

"ΕΠΑΝΕΙΣΟΔΟΣ ΣΤΟΝ ΒΡΟΓΧΟ"

Τμήμα χαρακτηριστικών των στιγμιότυπων Snapshot

- Λήψη ενός στιγμιότυπου (`EVENT_IMAGE_SNAPSHOT`) εάν είναι προγραμματισμένο να γίνει αυτό ή αν ζητήθηκε να γίνει από τον χρήστη.

Τμήμα χαρακτηριστικών λήξης Timelapse

- Τερματισμός των `mregs` `EVENT_TIMELAPSEEND` εάν το `rollover` είναι ενεργοποιημένο και έχει φτάσει η ώρα που έχει οριστεί να συμβεί αυτό.
- Το `Timelapse mreg` επισυνάπτεται από την τρέχουσα εικόνα εάν το `timelapse` είναι ενεργοποιημένο και έχει φτάσει η ώρα που έχει οριστεί να συμβεί αυτό. `EVENT_TIMELAPSE` (δημιουργία νέου αρχείου `mreg` εάν είναι η πρώτη φορά που συμβαίνει ειδιάλλως προσαρτά το τρέχον).

Τμήμα επανάληψης Video Loopback

- Τροφοδότηση των `ripes` των βίντεο συσκευών με κανονικές εικόνες και εικόνες `motion`.
- Τροφοδότηση της `webcam`.
 - Αν βρίσκεται σε `setup_mode` τροφοδότηση της κάμερας με την ειδική εικόνα για το `setup`.
 - Αν η λειτουργία `webcam_motion` είναι ενεργοποιημένη τροφοδότηση της κάμερας με την πρώτη εικόνα του κάθε δευτερολέπτου.
 - Αν η λειτουργία `webcam_motion` είναι απενεργοποιημένη τροφοδότηση της κάμερας με όλες τις ληφθείσες εικόνες.

Τμήμα ενημέρωσης παραμέτρων ανά δευτερόλεπτο

- Ανανέωση των μεταβλητών `output_normal`, `locate`, `smartmask` and `sql_mask` κάθε δευτερόλεπτο.

Τμήμα ορισμού καρτέ για το χρονοδιάγραμμα ύπνου sleeping mode

- Υπολογισμός του αναμενόμενου frame rate που βασίζεται στις ρυθμίσεις του framerate από το αρχείο configuration (cnt->conf.frame_limit).
- Υπολογισμός του χρόνου που έχει περάσει.
- Είσοδος σε Sleep mode ώστε να φτάσει στο απαιτούμενο framerate.
- Αν εμφανίζεται η παράμετρος cpu_low μετάβαση σε sleep mode για περισσότερο χρόνο έως ότου να φτάσει στην δοθείσα τιμή για το cpu_low.
- Επανάληψη του βρόγχου από την αρχή.

Πανεπιστήμιο Πειραιώς

ΚΕΦΑΛΑΙΟ 3-ΒΑΣΙΚΕΣ ΡΥΘΜΙΣΕΙΣ ΕΝΤΟΠΙΣΜΟΥ ΚΙΝΗΣΗΣ

Έχοντας γίνει η εισαγωγή για το λογισμικό Motion, θα προχωρήσουμε στις ρυθμίσεις και παραμετροποιήσεις που έλαβαν χώρα κατά τη διάρκεια εκπόνησης του εν λόγω project. Να σημειωθεί ότι όλες οι δοκιμές έγιναν με την έκδοση 3.2.12-3.4 εγκατεστημένη στο λειτουργικό σύστημα Ubuntu έκδοση 13.10 (Saucy Salamander).

Όπως προτυπώθηκε, όλες οι βασικές ρυθμίσεις γίνονται από το αρχείο configuration η διεύθυνση του οποίου βρίσκεται by default στο /etc/motion.conf. Σε κάθε μεταβολή-ρύθμιση του αρχείου configuration θα πρέπει να γίνεται επανεκκίνηση του Motion server:

```
sudo /etc/init.d/motion restart
```

Για να γίνουν ορατές όλες οι επιλογές του αρχείου configuration και να προχωρήσουμε στις ρυθμίσεις αρκεί να χρησιμοποιηθεί ένας editor ώστε να ανοίξουμε το αρχείο, όπως για παράδειγμα ο vim και ο gedit. Εδώ, χρησιμοποιούμε τον gedit:

```
sudo gedit /etc/motion/motion.conf
```

Με αυτόν τον τρόπο γίνονται ορατές όλες οι επιλογές όπως αναφέρονται και στον Πίνακα (2.5.3-1) του 2ου κεφαλαίου.

3.1 Παραμετροποίηση σχετικά με την εικόνα που λαμβάνεται

Μεταβαίνοντας στο αρχείο configuration, μπορούν να παραμετροποιηθούν κάποιες επιλογές ώστε να γίνεται αποθήκευση των εικόνων σε μια διεύθυνση αρχείων πιο άμεσα προσβάσιμη από τον διαχειριστή/χρήστη (η προεπιλεγμένη είναι η /tmp) και πώς να ρυθμιστούν κάποιες βασικές παράμετροι εγγραφής γενικότερα. Αρχικά, ενεργοποιούμε το Motion σε daemon mode, έτσι ώστε να μπορεί να ενεργοποιείται ανεξάρτητα.

```
# Start in daemon (background) mode and release terminal (default:
off)
daemon on
```

Αλλάζουμε την επιλογή που υποδεικνύει που γίνεται η εγγραφή του pid του daemon τοποθετώντας το στη διεύθυνση /home/simos/ έτσι ώστε να έχουμε την απαραίτητη άδεια για πρόσβαση.

```
# File to store the process ID, also called pid file. (default: not
defined)
process_id_file /HOME/simos/motion.pid
```

Το Setup-Mode παραμένει απενεργοποιημένο, καθώς αν ενεργοποιηθεί θα απενεργοποιήσει το daemon mode.

```
# Basic Setup Mode
# Start in Setup-Mode, daemon disabled. (default: off)
setup_mode off
```

Επιλέγεται η συσκευή βίντεο που θα χρησιμοποιηθεί από το Linux στο αρχείο συστήματος `/dev/video0`.

```
# Videodevice to be used for capturing (default /dev/video0)
# for FreeBSD default is /dev/bktr0
videodevice /dev/video0
```

Η είσοδος βίντεο αν χρησιμοποιηθεί USB κάμερα θα πρέπει να λάβει την τιμή 8 ή τις 0 ή 1 αν χρησιμοποιηθεί κάρτα βίντεο ή τηλεόρασης αντίστοιχα. Εδώ, χρησιμοποιούμε USB κάμερες.

```
# The video input to be used (default: 8)
# Should normally be set to 0 or 1 for video/TV cards, and 8 for USB
cameras
input 8
```

Μεταβάλλουμε τις επιλογές για την ανάλυση της εικόνας, χρησιμοποιώντας την 640×480.

```
# Image width (pixels). Valid range: Camera dependent, default: 352
width 640
# Image height (pixels). Valid range: Camera dependent, default: 288
height 480
```

Έπειτα, μπορούμε να ορίσουμε τον αριθμό των εικόνων που θα λαμβάνονται από το πρόγραμμα ανά δευτερόλεπτο, διαλέγοντας μεταξύ 2 και 100. Μια επιλογή της τάξης των 8 εικόνων ανά δευτερόλεπτο θα ήταν ικανοποιητική.

```
# Maximum number of frames to be captured per second.
# Valid range: 2-100. Default: 100 (almost no limit).
framerate 8
```

Επιπρόσθετα, θα ήταν χρήσιμο να ρυθμιστεί η φωτεινότητα, η αντίθεση και ο κορεσμός των ληφθέντων εικόνων αν η κάμερα που χρησιμοποιείται δεν έχει την επιλογή του auto-detection. Κατά την εκπόνηση της συγκεκριμένης εργασίας οι δοκιμές έγιναν στο χώρο που απεικονίζεται στην παρακάτω φωτογραφία, οπότε λήφθηκαν υπόψη συγκεκριμένες παράμετροι για τις εν λόγω ρυθμίσεις.



Εικόνα (3.1-1): Χώρος δοκιμών, παρακολούθησης και ανίχνευσης κίνησης

```
# Let motion regulate the brightness of a video device (default:
off).

# The auto_brightness feature uses the brightness option as its
target value.

# If brightness is zero auto_brightness will adjust to average
brightness value 128.

# Only recommended for cameras without auto brightness
auto_brightness off

# Set the initial brightness of a video device.

# If auto_brightness is enabled, this value defines the average
brightness level

# which Motion will try and adjust to.

# Valid range 0-255, default 0 = disabled
brightness 50

# Set the contrast of a video device.

# Valid range 0-255, default 0 = disabled
contrast 175

# Set the saturation of a video device.

# Valid range 0-255, default 0 = disabled
saturation 25
```

Σε περίπτωση που γίνει και μελέτη από λήψη βίντεο θα ήταν χρήσιμο να αλλαχτεί ο κωδικοποιητής της βίντεο λήψης σε mpeg4 που είναι προτιμότερος από τον προεπιλεγμένο flash.

```
# Codec to used by ffmpeg for the video compression.
```



```

# Timelapse mpegs are always made in mpeg1 format independent from
this option.

# Supported formats are: mpeg1 (ffmpeg-0.4.8 only), mpeg4 (default),
and msmpeg4.
# mpeg1 - gives you files with extension .mpg

# mpeg4 or msmpeg4 - gives you files with extension .avi

# msmpeg4 is recommended for use with Windows Media Player because
# it requires no installation of codec on the Windows client.

# swf - gives you a flash film with extension .swf
# flv - gives you a flash video with extension .flv
# ffv1 - FF video codec 1 for Lossless Encoding ( experimental )
# mov - QuickTime ( testing )

ffmpeg_video_codec mpeg4

```

Όσον αφορά τη διεύθυνση αρχείου που γίνεται εγγραφή είναι απαραίτητο να αλλάξει διότι ο υπολογιστής στον οποίο γίνονται οι δοκιμές χρησιμοποιεί λειτουργικό σύστημα Ubuntu το οποίο σε κάθε επανεκκίνηση του υπολογιστή σβήνει τα αρχεία. Έτσι, λοιπόν, από την προεπιλεγμένη θέση η οποία είναι η /tmp προσπαθούμε να βρούμε μια διεύθυνση στην οποία έχουμε δικαιώματα πρόσβασης σαν χρήστες καθώς και αρκετό χώρο στον δίσκο για εγγραφές. Κατά τη διάρκεια εκπόνησης του τρέχοντος project επειδή όπως θα αναλυθεί παρακάτω γίνεται χρησιμοποίηση web server, χρησιμοποιήθηκε η /var/www/imageviewer/motion.

```

# Target base directory for pictures and films

# Recommended to use absolute path. (Default: current working
directory)
target_dir /var/www/imageviewer/motion

```

Είναι πολύ σημαντικό στην εικόνα που θα ληφθεί να είναι ορατό το σημείο στο οποίο ανιχνεύτηκε η κίνηση, οπότε ενεργοποιώντας την επιλογή Locate μπορεί να συμβεί αυτό δημιουργώντας ένα ορθογώνιο στην περιοχή που μας ενδιαφέρει.

```

# Locate and draw a box around the moving object.
# Valid values: on, off and preview (default: off)
# Set to 'preview' will only draw a box in preview_shot pictures.
locate on

```

Στην εικόνα που εξάγεται μπορούν να προστεθούν διάφορες πληροφορίες, όπως η πλήρης ημερομηνία λήψης της εικόνας με ακρίβεια δευτερολέπτου, χαρακτηριστικά όπως οι συντεταγμένες όπου εντοπίστηκε κίνηση, από ποιά κάμερα εντοπίστηκε, ο αριθμός των αλλαγμένων pixels και ο αριθμός καταγεγραμμένου γεγονότος.

```

# Text Display
# %Y = year, %m = month, %d = date,
# %H = hour, %M = minute, %S = second, %T = HH:MM:SS,

```

```

# %v = event, %q = frame number, %t = thread (camera) number,
# %D = changed pixels, %N = noise level, \n = new line,
# %i and %J = width and height of motion area,
# %K and %L = X and Y coordinates of motion center
# %C = value defined by text_event - do not use with text_event!
# You can put quotation marks around the text to allow
# leading spaces

```

Έτσι, λοιπόν, μπορούμε να επιλέξουμε με βάση τις παραπάνω μεταβλητές σε ποίο σημείο των άκρων της οθόνης θα είναι ορατά τα χαρακτηριστικά, καθώς και ποιά θα είναι αυτά τα οποία μας ενδιαφέρουν να απεικονίζονται.

```

# Draws the timestamp using same options as C function strftime(3)
# Default: %Y-%m-%d\n%T = date in ISO format and time in 24 hour
clock
# Text is placed in lower right corner
text_right %Y-%m-%d\n%T-%q

# Draw the number of changed pixed on the images (default: off)
# Will normally be set to off except when you setup and adjust the
motion settings
# Text is placed in upper right corner
text_changes off
# This option defines the value of the special event conversion
specifier %C
# You can use any conversion specifier in this option except %C. Date
and time
# values are from the timestamp of the first image in the current
event.
# Default: %Y%m%d%H%M%S
# The idea is that %C can be used filenames and text_left/right for
creating
# a unique identifier for each event.
text_event %Y%m%d%H%M%S

```



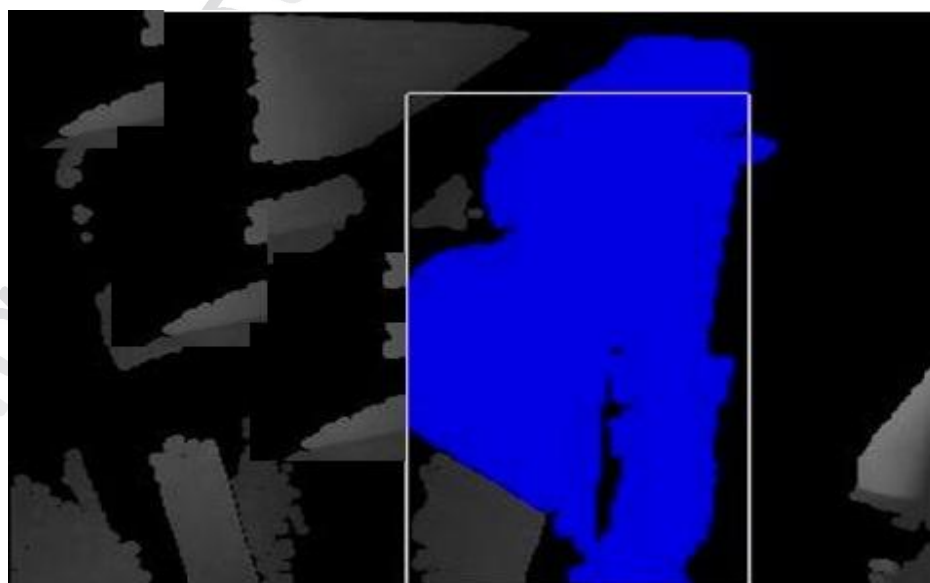
Εικόνα (3.1-2): Εικόνα που δημιουργήθηκε έπειτα από ανίχνευση κίνησης έχοντας ενεργοποιημένες τις επιλογές *locate* και *text*

Επίσης, υπάρχει η επιλογή εξαγωγής και αποθήκευσης των εικόνων όπου ανιχνεύεται περισσότερο κίνηση, όταν ανιχνεύεται για πρώτη φορά κίνηση μόνο ή ακόμα και όταν ανιχνεύεται κίνηση μόνο στο κέντρο της εικόνας. Φυσικά, υπάρχουν και οι επιλογές απλής ενεργοποίησης και απενεργοποίησης. Υπάρχουν, ακόμη, ρυθμίσεις με ποσοστιαία επιλογή για την ποιότητα της εικόνας όταν αυτή έχει οριστεί να είναι jpeg και επιλογή εξαγωγής εικόνας σε μορφή ppm.

```
# Output 'normal' pictures when motion is detected (default: on)
# Valid values: on, off, first, best, center
# When set to 'first', only the first picture of an event is saved.
# Picture with most motion of an event is saved when set to 'best'.
# Picture with motion nearest center of picture is saved when set to
'center'.
# Can be used as preview shot for the corresponding movie.
output_normal on
# The quality (in percent) to be used by the jpeg compression
(default: 75)
quality 75
# Output ppm images instead of jpeg (default: off)
ppm off
```

Η σημαντικότερη ρύθμιση που προσφέρεται είναι η επιλογή για εξαγωγή της ληφθείσας σε μορφή ghost εικόνας (Εικόνα 3.1-3), όπου το μόνο που είναι ορατό πλέον είναι τα pixels που έχουν αλλάξει και κατ' επέκταση ανήκουν στο κινούμενο αντικείμενο, διασφαλίζοντας έτσι την ιδιωτικότητα, αφού ουσιαστικά υπάρχει μόνο η πληροφορία της κίνησης.

```
# Output pictures with only the pixels moving object (ghost images)
(default: off)
output_motion on
```



Εικόνα (3.1-3): Εικόνα Ghost. Είναι ορατά μόνο τα pixels που έχουν μεταβληθεί εξασφαλίζοντας την ιδιωτικότητα του παρακολουθούμενου

3.1-1 Προσθέτοντας περισσότερες από μία κάμερες

Ένα από τα βασικά χαρακτηριστικά του Motion είναι ότι μπορεί να εργαστεί ταυτόχρονα με περισσότερες από μία κάμερες. Σε αυτήν την περίπτωση, λοιπόν, δεν θα πρέπει να γίνει κλήση του Motion τόσες φορές όσες οι επιπλέον κάμερες που διαθέτει ο χρήστης απλά θα πρέπει να δημιουργηθούν και να τροποποιηθούν τα ανάλογα αρχεία "thread config files". Δηλαδή, αν έχουμε n κάμερες θα χρειαστούν το βασικό αρχείο "motion.conf" και n αρχεία "thread config files", σε αντίθεση με την ύπαρξη μόνο μιας κάμερας που τότε θα χρειαζόταν μόνο το αρχείο "motion.conf". Έτσι, θα έχουμε συνολικά n+1 configuration αρχεία. Μία επιλογή που είναι κοινή για όλες τις κάμερες μπορεί να οριστεί στο αρχείο "motion.conf" ενώ οι διαφορετικές-μοναδικές επιλογές της κάθε κάμερας ορίζονται στο thread config file που της αναλογεί. Το thread config file της πρώτης κάμερας θα είναι το πρώτο thread config file που καλείται από το αρχείο "motion.conf", της δεύτερης κάμερας το δεύτερο thread config file που καλείται από το αρχείο "motion.conf" κ.ο.κ. Κάθε επιλογή η οποία ορίζεται στο αρχείο motion.conf χρησιμοποιείται για όλες τις κάμερες εκτός από τις κάμερες στις οποίες ορίζεται ξεχωριστά μια διαφορετική ρύθμιση για την ίδια όμως επιλογή στο αντίστοιχο thread config file. Αν ορίζεται η επιλογή "thread" στο αρχείο motion.conf, διαβάζονται τα αρχεία thread configuration. Στη συνέχεια το Motion διαβάζει και το υπόλοιπο αρχείο motion.conf.

Όπως προτυπώθηκε και στην παράγραφο 2.5.1 το Motion αναγνωρίζει τις παραμέτρους για το configuration λαμβάνοντας υπόψη κάποιους κανόνες, ένας εκ των οποίων είναι ότι αν η ίδια παράμετρος υπάρχει σε περισσότερα από ένα σημεία, λαμβάνεται υπόψη αυτή που διαβάζεται από το τελευταίο σημείο. Έτσι, λοιπόν, αν οποιαδήποτε επιλογή είναι κοινή στο αρχείο motion.conf και στα thread config files που έχουν ήδη διαβαστεί, ακυρώνεται και λαμβάνεται υπόψη μόνο αυτή του motion.conf. Για το λόγο αυτό θα πρέπει τα thread config files να τοποθετούνται στο τέλος του motion.conf ώστε να είναι αυτά που θα λαμβάνονται τελικώς υπόψη.

Επομένως, για την προσθήκη περισσότερων από μιας καμερών θα πρέπει να προστεθούν τα αρχεία thread config. Στη διεύθυνση/etc/motion.conf είναι πολύ πιθανό να παρατηρηθούν πολλαπλά αρχεία threadN.conf αν υπάρχουν περισσότερες από μία κάμερες και θα πρέπει να παραμετροποιηθούν και αυτά καταλλήλως. Στη δική μας περίπτωση χρησιμοποιούνται 3 κάμερες, οπότε απαιτούνται 3 αρχεία thread config και φυσικά το αρχείο configuration.

```
# Thread config files - One for each camera.
# Except if only one camera - You only need this config file.
# If you have more than one camera you MUST define one thread
# config file for each camera in addition to this config file.
#####

# Remember: If you have more than one camera you must have one
# thread file for each camera. E.g. 2 cameras requires 3 files:
# This motion.conf file AND thread1.conf and thread2.conf.
# Only put the options that are unique to each camera in the
# thread config files.
```

```
thread /usr/local/etc/thread1.conf
thread /usr/local/etc/thread2.conf
thread /usr/local/etc/thread3.conf
```

Το περιεχόμενο ενός αρχείου thread config φαίνεται παρακάτω. Σε αυτά τα αρχεία εμπεριέχονται μόνο οι ρυθμίσεις που είναι ξεχωριστές για κάθε κάμερα. Για όσα χαρακτηριστικά είναι κοινά, αρκεί η δήλωσή τους μόνο στο γενικό αρχείο configuration. Για παράδειγμα, λοιπόν, η μορφή του αρχείου thread1.conf για την πρώτη κάμερα είναι:

```
# /usr/local/etc/thread1.conf
#
# This config file was generated by motion 3.2.12

#####
# Capture device options
#####
# Videodevice to be used for capturing (default /dev/video0)
# for FreeBSD default is /dev/bktr0
videodevice /dev/video0

# The video input to be used (default: 8)
# Should normally be set to 1 for video/TV cards, and 8 for USB
cameras
input 8

# Draw a user defined text on the images using same options as C
function strftime(3)
# Default: Not defined = no text
# Text is placed in lower left corner
text_left CAMERA 1

# Target Directories and filenames For Images And Films
# For the options snapshot_, jpeg_, mpeg_ and timelapse_filename
# you can use conversion specifiers
# %Y = year, %m = month, %d = date,
# %H = hour, %M = minute, %S = second,
# %v = event, %q = frame number, %t = thread (camera) number,
# %D = changed pixels, %N = noise level,
# %i and %J = width and height of motion area,
# %K and %L = X and Y coordinates of motion center
# %C = value defined by text_event
# Quotation marks round string are allowed.

# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working
directory)
target_dir /var/www/imageviewer/motion/cam1

# Live Webcam Server

# The mini-http server listens to this port for requests (default: 0
= disabled)
webcam_port 8081

# Command to be executed when a picture (.ppm|.jpg) is saved
(default: none)
# The filename of the picture is appended as an argument for the
command.
on_picture_save /usr/local/motion-extras/camparse1.pl
```

```
# Command to be executed when a movie file (.mpg|.avi) is closed.
(default: none)
# Filename of movie is appended as an argument for the command.
on_movie_end /usr/local/motion-extras/mpegparse1.pl
```

Για τη δεύτερη κάμερα θα πρέπει να δημιουργηθεί το δεύτερο αρχείο thread config thread2.conf.

```
# /usr/local/etc/thread1.conf
#
# This config file was generated by motion 3.2.12

#####
# Capture device options
#####
# Videodevice to be used for capturing (default /dev/video0)
# for FreeBSD default is /dev/bktr0
videodevice /dev/video1
```

Αρχικά, θα πρέπει να προσδιοριστεί ένα διαφορετικό port για την κάμερα αυτή που θα χρησιμεύσει αργότερα για τον έλεγχο και τη διαχείριση της.

```
# The mini-http server listens to this port for requests (default: 0
= disabled)
webcam_port 8082
```

Επίσης, για να είναι ευδιάκριτο από ποιά κάμερα γίνεται η λήψη θα πρέπει να αναγράφεται ο αριθμός της στην εικόνα:

```
# Draw a user defined text on the images using same options as C
function strftime(3)
# Default: Not defined = no text
# Text is placed in lower left corner
text_left CAMERA 2
```

Οι εικόνες που λαμβάνονται από αυτή την κάμερα αποθηκεύονται σε ένα ξεχωριστό φάκελο για την ίδια:

```
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working
directory)
target_dir /var/www/imageviewer/motion/cam2
```

Αντίστοιχα, για την τρίτη κάμερα θα πρέπει να δημιουργηθεί το τρίτο αρχείο thread config thread3.conf.

```
# /usr/local/etc/thread1.conf
#
# This config file was generated by motion 3.2.12

#####
# Capture device options
#####
# Videodevice to be used for capturing (default /dev/video0)
# for FreeBSD default is /dev/bktr0
videodevice /dev/video2
```

Θα πρέπει να προσδιοριστεί ένα διαφορετικό port για την κάμερα αυτή που θα χρησιμεύσει αργότερα για τον έλεγχο και τη διαχείριση της.

```
# The mini-http server listens to this port for requests (default: 0
= disabled)
webcam_port 8083
```

Για να είναι ευδιάκριτο από ποιά κάμερα γίνεται η λήψη θα πρέπει να αναγράφεται ο αριθμός της στην εικόνα:

```
# Draw a user defined text on the images using same options as C
function strftime(3)
# Default: Not defined = no text
# Text is placed in lower left corner
text_left CAMERA 3
```

Οι εικόνες που λαμβάνονται από αυτή την κάμερα αποθηκεύονται σε ένα ξεχωριστό φάκελο για την ίδια:

```
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working
directory)
target_dir /var/www/imageviewer/motion/cam3
```

3.2 Παραμετροποιήσεις για την ανίχνευση κίνησης

Σε αυτή την παράγραφο συνεχίζονται οι ρυθμίσεις στο αρχείο configuration που αφορούν τα χαρακτηριστικά για την ανίχνευση της κίνησης. Το σημαντικότερο από αυτά είναι το όριο (threshold) πάνω από το οποίο ανιχνεύεται ένα γεγονός , σε σχέση με την κατάσταση ηρεμίας. Στο συγκεκριμένο project, έπειτα από δοκιμές επιλέχθηκε η τιμή 20000 και απενεργοποιήθηκε η διακύμανσή του.

```
# Threshold for number of changed pixels in an image that
# triggers motion detection (default: 1500)
threshold 20000
# Automatically tune the threshold down if possible (default: off)
threshold_tune off
```

Επίσης, επιλέχθηκε ο θόρυβος για τις συνθήκες του χώρου που διενεργείται το συγκεκριμένο project να λάβει την τιμή 20 και απενεργοποιήθηκε η αυτόματη ρύθμισή του, παραμένοντας στις αρχικά ορισθείσες τιμές.

```
# Noise threshold for the motion detection (default: 32)
noise_level 20
# Automatically tune the noise threshold (default: on)
noise_tune off
```

Η ρύθμιση του Despeckle για τον θόρυβο εξαρτάται από τις συνθήκες του χώρου στον οποίο εφαρμόζεται η βιντεοεπιτήρηση. Κατά τη διάρκεια των δοκιμών ήταν ικανοποιητική η προεπιλεγμένη και προτεινόμενη εφαρμογή όλων των χαρακτηριστικών του Despeckle ταυτόχρονα (Erode,Dilate,Label).

```
# Despeckle motion image using (e)rode or (d)ilate or (l)abel
(Default: not defined)
# Recommended value is EedDl. Any combination (and number of) of E,
e, d, and D is valid.
# (l)abeling must only be used once and the 'l' must be the last
letter.
# Comment out to disable
despeckle EedDl
```

Επίσης, η ευαισθησία για την επιλογή των διαδοχικών frames στα οποία εντοπίζεται κίνηση επιλέχθηκε στο 3.

```
# Picture frames must contain motion at least the specified number of
frames
# in a row before they are detected as true motion. At the default of
1, all
# motion is detected. Valid range: 1 to thousands, recommended 1-5
minimum_motion_frames 3
```

Τέλος, παρέμεινε απενεργοποιημένη η δυνατότητα αποθήκευσης εικόνων όταν δεν υπάρχει ανίχνευση κίνησης αφού είναι περιττό να δεσμεύεται αποθηκευτικός χώρος χωρίς αιτία.

```
# Always save images even if there was no motion (default: off)
output_all off
```

3.3 Χρησιμοποιώντας βάσεις δεδομένων

Το Motion μπορεί να μεταγλωττιστεί και με την βάση δεδομένων MySQL και με την PostgreSQL. Όταν γίνει χρήση τους, το Motion έχει τη δυνατότητα να προσθέσει μια εγγραφή σε έναν πίνακα της βάσης δεδομένων με πεδία το όνομα του αρχείου, το χρόνο του συμβάντος καθώς και τον τύπο αρχείου. Δεν έχει τη δυνατότητα να προσθέσει δυαδικές εικόνες στη βάση δεδομένων αλλά και ούτε να διαγράψει παλαιότερες εγγραφές. Με άλλα λόγια, το Motion μόνο προσθέτει σε βάση δεδομένων εγγραφές αποθηκευμένων αρχείων που δημιουργούνται σε περίπτωση που συμβεί κάποιο γεγονός στο προς μελέτη προσκήνιο. Από την επιλογή sql_log μπορούν να οριστούν οι τύποι των αρχείων που θα εισαχθούν στη βάση δεδομένων. Οι πιθανοί τύποι μπορεί να είναι triggered image file, snapshot image file, triggered mpeg file, timelapse mpeg file. Παρακάτω, απεικονίζεται το ανάλογο τμήμα κώδικα για τις επιλογές sql_log το οποίο είναι κοινό και για την MySQL και για την PostgreSQL.

```
# Common Options For MySQL and PostgreSQL database features.
# Options require the MySQL/PostgreSQL options to be active also.

#Log to the database when creating motion triggered image file
(default: on)
sql_log_image on

#Log to the database when creating a snapshot image file (default:
on)
sql_log_snapshot on
```


Η επιλογή του mpeg αρχείου παραμένει απενεργοποιημένη αφού στην προκειμένη περίπτωση μας ενδιαφέρει να καταγράψουμε μόνο εικόνες.

```
#Log to the database when creating motion triggered mpeg file
(default: off)
sql_log_mpeg off
```

```
#Log to the database when creating timelapse mpeg file (default: off)
sql_log_timelapse off
```

3.3.1 Ρυθμίσεις στην MySQL.

Αρχικά, θα πρέπει να εγκαταστήσουμε την MySQL εκτελώντας την παρακάτω εντολή, καθώς και να θέσουμε έναν κωδικό πρόσβασης για τον χρήστη root:

```
sudo apt-get install mysql-server
```

Με το πέρας της εγκατάστασης ο the MySQL server θα ξεκινήσει να εκτελείται αυτόματα.

Με τη χρήση της:

```
sudo netstat -tap | grep mysql
```

μπορούμε να ελέγξουμε αν ο server είναι ενεργός και να παρατηρήσουμε τα χαρακτηριστικά του. Εάν είναι σκόπιμο, μπορούμε να εισέλθουμε στο αρχείο configuration της MySQL και να γίνουν ρυθμίσεις σχετικά με τα log files, τις ports, την IP διεύθυνση του server για επικοινωνία με network hosts, κτλ. Αυτό επιτυγχάνεται με τη χρήση της:

```
sudo gedit /etc/mysql/my.cnf
```

Φυσικά, σε κάθε μεταβολή του παραπάνω αρχείου απαιτείται επανεκκίνηση της MySQL:

```
sudo service mysql restart
```

Στη συνέχεια, θα πρέπει να δημιουργηθεί μια νέα βάση δεδομένων με ένα οποιοδήποτε όνομα της επιλογής του χρήστη και το οποίο θα πρέπει να οριστεί στην επιλογή mysql_db του αρχείου configuration. Αφού πρώτα εισέλθουμε σαν χρήστες root (mysql -u root -p και πληκτρολογώντας τον κωδικό) επιλέγουμε τη νέα βάση δεδομένων στην MySQL ονομάζοντας την "simos" (mysql> CREATE DATABASE simos;) και την δηλώνουμε στο Motion:

```
# Database Options For MySQL
```

```
# Mysql database to log to (default: not defined)
```

```
mysql_db simos
```

```
# The host on which the database is located (default: localhost)
mysql_host localhost

# User account name for MySQL database (default: not defined)
mysql_user root

# User password for MySQL database (default: not defined)
mysql_password simos
```

Και στη συνέχεια δημιουργούμε έναν νέο πίνακα με το όνομα "simosss" στη βάση δεδομένων simos (mysql> USE simos;) με τα παρακάτω πεδία:

- camera (int) - Ο αριθμός της κάμερας (thread).
- time_stamp (varchar(60)) -Χρονοσφραγίδα (timestamp) της εικόνας σε format βάσης δεδομένων.

Το κομμάτι του κώδικα, λοιπόν, ώστε να δημιουργηθεί ο πίνακας στην SQL θα μπορεί να είναι αυτό:

```
mysql> CREATE TABLE simosss (camera INT, time_stamp VARCHAR(60));
```

Επίσης, θα πρέπει να γίνει ενημέρωση του Πίνακα grant ώστε να υπάρχει πρόσβαση στο όνομα χρήστη της MySQL που επιλέχθηκε για το Motion.

Αντίστοιχα, θα πρέπει να δηλωθεί το sql query στο αρχείο configuration του Motion:

```
sql_query insert into simosss(camera, time_stamp) values('%t', '%Y-
%m-%d %T')
```

Αν θέλουμε να δούμε το περιεχόμενο της βάσης δεδομένων που δημιουργήσαμε:

```
mysql> USE simos;
mysql> SHOW TABLES FROM simos
```

Και αν θέλουμε να δούμε το περιεχόμενο του πίνακα με τις τιμές των μεταβλητών που μας ενδιαφέρει:

```
mysql> SHOW INDEX FROM simosss;
```

Επιπρόσθετα, μπορούμε να δούμε τις καταγραφές του πίνακα που έχουμε δημιουργήσει (simosss) χρησιμοποιώντας την εντολή:

```
mysql> SELECT * FROM simosss
```

Έτσι, στην επόμενη εικόνα (Εικόνα 3.3.1-2) είναι ορατές οι καταγραφές στη βάση δεδομένων έπειτα από τη δημιουργία events- ανιχνεύσεων.

```
| 1 | 2014-06-01 14:58:07 |
| 2 | 2014-06-01 14:58:08 |
| 2 | 2014-06-01 14:58:08 |
| 2 | 2014-06-01 14:58:08 |
| 2 | 2014-06-01 14:58:08 |
| 1 | 2014-06-01 14:58:08 |
| 1 | 2014-06-01 14:58:09 |
| 2 | 2014-06-01 17:02:55 |
| 2 | 2014-06-01 17:02:55 |
| 2 | 2014-06-01 17:02:55 |
| 2 | 2014-06-01 17:02:55 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:56 |
| 2 | 2014-06-01 17:02:57 |
| 2 | 2014-06-01 17:02:57 |
| 2 | 2014-06-01 17:02:57 |
| 2 | 2014-06-01 17:02:57 |
+-----+
46 rows in set (0.00 sec)

mysql> █
```

Εικόνα (3.3.1-1): Καταγραφή events στη βάση δεδομένων

3.4 Αποστολή e-mail κατά τη δημιουργία ενός γεγονότος

Στην περίπτωση όπου θέλουμε να ενεργοποιήσουμε κάποια ενέργεια για να ειδοποιηθεί ο χρήστης όταν συμβεί κάποιο γεγονός, δηλαδή ανίχνευση κίνησης, θα πρέπει να μεταβούμε στο αρχείο configuration file και να προγραμματίσουμε τις εξής παραμέτρους:

```
# Command to be executed when an event starts. (default: none)

# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start value
```

Θα πρέπει, λοιπόν, να δημιουργήσουμε scripts για αποστολή e-mail ή γραπτού μηνύματος SMS και να τα καλέσουμε από το αρχείο configuration, τοποθετώντας απλά το όνομα του εκάστοτε script στη θέση του "value".

Στο συγκεκριμένο project, θα γίνεται αποστολή μέσω ενός λογαριασμού που ανήκει στο gmail με διεύθυνση "simos8456@gmail.com" και παραλαβή σε ένα λογαριασμό που ανήκει στο hotmail με διεύθυνση "djsimko_@hotmail.com".

Αναλυτικότερα, με τις κατάλληλες ρυθμίσεις μπορεί να ενεργοποιηθεί η δυνατότητα αποστολής e-mail κατά την έναρξη ή λήξη ενός γεγονότος. Παρακάτω, παρατίθεται το ανάλογο script (στο οποίο έχει δοθεί η ονομασία mail_motion.sh) για

την ενεργοποίηση του e-mail κατά την έναρξη γεγονότος, το οποίο το αποθηκεύουμε σε οποιοδήποτε φάκελο έχουμε πρόσβαση με δικαιώματα διαχειριστή (π.χ. /usr/local/etc/ ή /Home/Desktop κ.ο.κ).

```
#!/bin/sh

#location of 'mail' binary

MAIL="/usr/bin/mail"

#Destination e-mail address

TO="djsimko_@hotmail.com"

#Subject of the e-mail

SUBJECT="Alarm!!!Motion detected"

echo -e "This is an automated message generated by motion.\n\nMotion
detected: $1\n\n" | $MAIL -s "$SUBJECT" $TO
```

Επιπρόσθετα, θα πρέπει να παραμετροποιηθεί το λειτουργικό μας σύστημα ώστε να έχει τη δυνατότητα αποστολής email. Για να συμβεί αυτό θα πρέπει να εγκατασταθούν τα παρακάτω packages:

```
sudo apt-get install ssmtp
sudo apt-get install mailutils
sudo apt-get install libsasl2-2 ca-certificates libsasl2-modules
```

Έπειτα, θα πρέπει να γίνει παραμετροποίηση του postfix με την εντολή:

```
sudo dpkg-reconfigure postfix
```

Επιλέγοντας τη διασύνδεση Internet with Smart Host και ρυθμίζοντας στην πρώτη καρτέλα το όνομα του localhost simos-Presario-V6000-RG390UA-ABA και στη δεύτερη την επιλογή για το gmail smtp.gmail.com:587. Όλες οι άλλες επιλογές παραμένουν οι default, όπως προτείνεται από το postfix configuration script.

Στη συνέχεια, αποκτούμε πρόσβαση με δικαιώματα διαχειριστή στο αρχείο /etc/postfix/main.cf το οποίο διαμορφώνουμε ως εξής:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete
version
```

```

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database =
btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database =
btree:${queue_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package
# for
# information on enabling SSL in the smtp client.

myhostname = localhost
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = localhost, localhost.localdomain
mynetworks = 127.0.0.0/8
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = loopback-only
inet_protocols = all

#####
##### non debconf entries start here #####

##### client TLS parameters #####
smtp_tls_loglevel=1
smtp_tls_security_level=encrypt
smtp_sasl_auth_enable=yes
smtp_sasl_password_maps=hash:/etc/postfix/sasl/passwd
smtp_sasl_security_options = noanonymous

##### map simos@simos-Presario-V6000-RG390UA-ABA to
simos8456@gmail.com #####
smtp_generic_maps=hash:/etc/postfix/generic

relayhost=[smtp.gmail.com]:587

```

Δημιουργούμε το αρχείο `/etc/postfix/generic` το οποίο περιέχει μόνο την αντιστοίχιση του `localhost` με τον αποστολέα του email:

```
simos@simos-Presario-V6000-RG390UA-ABA simos8456@gmail.com
```

Δημιουργούμε το αρχείο `etc/postfix/generic.db` εκτελώντας τις εντολές:

```
cd /etc/postfix
postmap generic
```

Αντίστοιχα, δημιουργούμε το αρχείο `/etc/postfix/sasl/passwd` που περιέχει τα `credentials` του αποστολέα του email (για ευνόητους λόγους δεν απεικονίζεται το password):

```
[smtp.gmail.com]:587      simos8456@gmail.com:*****
```

Και τέλος, δημιουργούμε το αρχείο `passwd.db` με εκτέλεση των παρακάτω εντολών:

```
cd /etc/postfix/sasl
postmap passwd
chown root.root passwd passwd.db
chmod 600 passwd passwd.db
```

και κάνουμε επανεκκίνηση του postfix :

```
sudo postfix reload
```

Για ακόμη καλύτερα αποτελέσματα, ρυθμίζουμε και το αρχείο configuration του `ssmtp`:

```
/etc/ssmtp/ssmtp.conf
```

το οποίο διαμορφώνεται ως εξής:

```
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
#root=simos8456@gmail.com

# The place where the mail goes. The actual machine name is required
no
# MX records are consulted.
mailhub=smtp.gmail.com:587

AuthUser=simos8456@gmail.com
AuthPass=*****
UseSTARTTLS=YES
UseTLS=YES
AuthMethod=LOGIN

smtpd_use_tls = yes
smtpd_tls_auth_only = no

# Where will the mail seem to come from?
rewriteDomain=gmail.com

# The full hostname
```

```
hostname=smtp.gmail.com:587
# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=YES
```

Αν θέλουμε να εξετάσουμε την ομαλή λειτουργία της αποστολής υπάρχει η δυνατότητα ανάγνωσης των log files από το αρχείο:

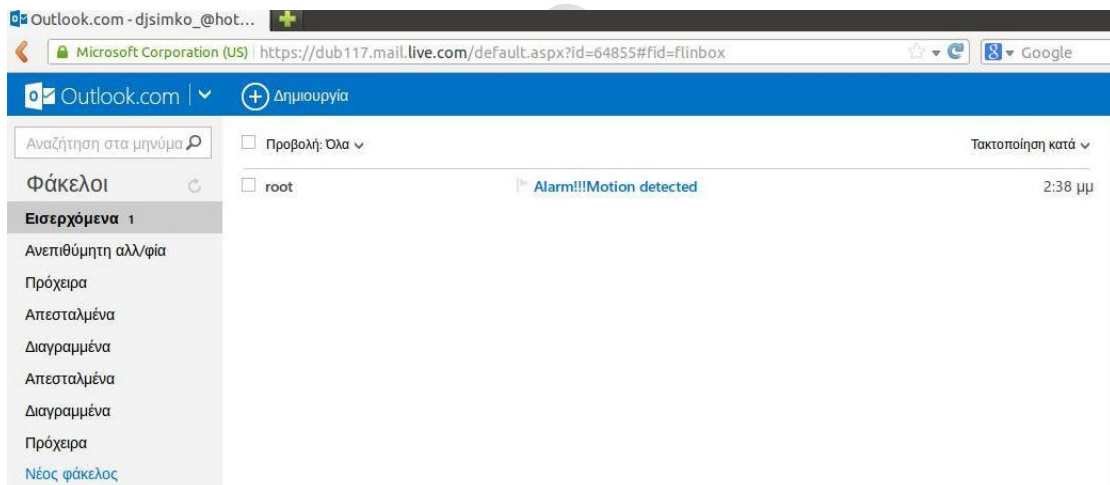
```
sudo gedit var/log/mail.log
```

Το μόνο που απομένει, λοιπόν, είναι να καλέσουμε το script από το αρχείο configuration. Η ρύθμιση θα είναι:

```
# Command to be executed when an event starts. (default: none)

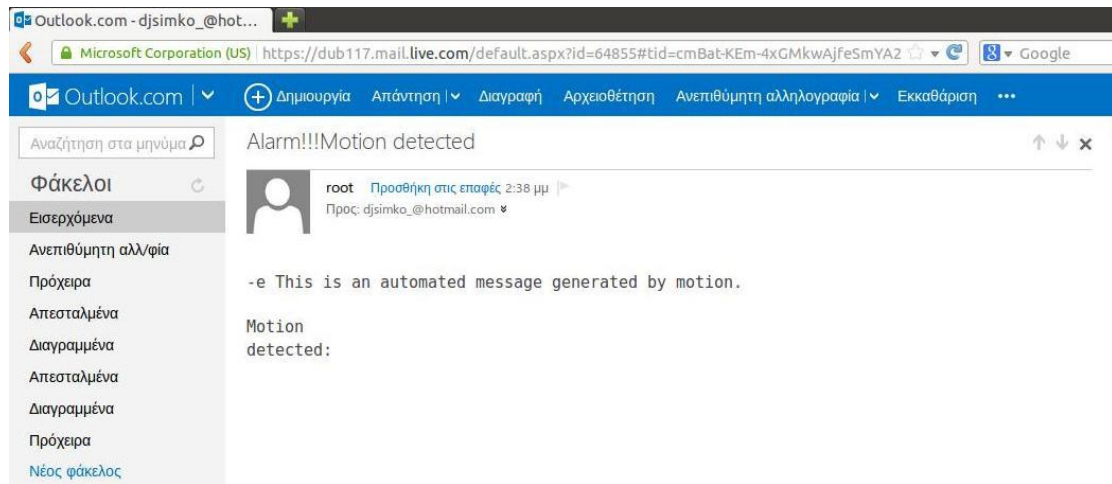
# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start /usr/local/etc/mail_motion
```

Με αυτόν τον τρόπο, από την επόμενη φορά που θα σημειωθεί ανίχνευση κίνησης θα γίνει αποστολή e-mail στο λογαριασμό "djsimko_@hotmail.com".



Εικόνα (3.4-1): Αποστολή e-mail έπειτα από ανίχνευση κίνησης

Το δε περιεχόμενο του e-mail εξασφαλίζει 100 % την ιδιωτικότητα του ασθενούς, καθώς, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 3.4-2), η μόνη πληροφορία που περιλαμβάνεται είναι η ειδοποίηση και καθόλου το στιγμιότυπο της κίνησης.



Εικόνα (3.4-2): Περιεχόμενο ληφθέντος e-mail έπειτα από ανίχνευση κίνησης

3.5 Αποστολή γραπτού μηνύματος SMS κατά τη δημιουργία ενός γεγονότος

Επίσης, με αντίστοιχες ρυθμίσεις μπορεί να ενεργοποιηθεί και η δυνατότητα αποστολής γραπτού μηνύματος SMS κατά την έναρξη ή λήξη ενός γεγονότος. Σε αυτή την περίπτωση είναι απαραίτητη η χρήση ενός USB GSM/3G MODEM (ή ακόμα και Smartphone σε λειτουργία Tethering) με κάρτα sim (οποιοδήποτε φυσικά παρόχου), ένας διαθέσιμος τηλεφωνικός αριθμός καθώς και παραμετροποίηση του λειτουργικού συστήματος με εγκατάσταση του λογισμικού gammu το οποίο προσδίδει την ικανότητα αποστολής γραπτών μηνυμάτων SMS.

Αρχικά, θα πρέπει να γίνει η παραμετροποίηση του λειτουργικού συστήματος (με την εγκατάσταση μόνο του gammu) ώστε να υπάρχει η δυνατότητα αποστολής SMS.

Συγκεκριμένα, γίνεται εγκατάσταση του gammu με την εντολή:

```
sudo apt-get install gammu
```

και αφού τοποθετηθεί το USB GSM/3G MODEM σε μια ελεύθερη θύρα USB, πληκτρολογούμε την :

```
dmesg | grep tty
```

για να διαπιστώσουμε σε ποιά θύρα έχει γίνει "mount" το USB DONGLE. Συνήθως, τα USB MODEMS γίνονται "mount" στη θύρα /dev/ttyUSB0, ενώ αν χρησιμοποιηθεί ένα κινητό τηλέφωνο Android σε λειτουργία tethering γίνεται "mount" στη θύρα /dev/ttyACM0.

Προχωρώντας στην παραμετροποίηση του gammu, αρκεί η εντολή:

```
sudo gammu-config
```


όπου το μόνο που θα πρέπει να δηλωθεί είναι η παράμετρος Port (P), λαμβάνοντας την τιμή που είχαμε λάβει παραπάνω, δηλαδή: /dev/ttyUSB0.

Τέλος, στην περίπτωση μας χρησιμοποιήθηκε ένα HUAWEI e155 USB 3G MODEM με κάρτα sim γνωστού ελληνικού παρόχου τηλεπικοινωνιών, ο αριθμός του οποίου αποθηκεύτηκε στον τηλεφωνικό αριθμό στον οποίο επιθυμούμε να γίνεται αποστολή μηνύματος κατά τη δημιουργία events.

Από τη στιγμή αυτή, είναι δυνατή η αποστολή γραπτού μηνύματος SMS με οποιοδήποτε text από τη γραμμή εντολών:

```
sudo gammu sendsms TEXT +3069***** -text "Test"
```

Επιστροφή τώρα στο λογισμικό Motion. Όπως και στην περίπτωση της αποστολής e-mail, θα δημιουργήσουμε script για αποστολή γραπτού μηνύματος SMS και θα το καλέσουμε από το αρχείο configuration, τοποθετώντας απλά το όνομα του στη θέση του "value". Παρακάτω, παρατίθεται το ανάλογο script (στο οποίο έχει δοθεί η ονομασία sms_motion.sh) για την ενεργοποίηση του γραπτού μηνύματος sms κατά την έναρξη γεγονότος, το οποίο το αποθηκεύουμε σε οποιοδήποτε φάκελο έχουμε πρόσβαση με δικαιώματα διαχειριστή (π.χ. /usr/local/etc/ ή /Home/Desktop κ.ο.κ).

```
#!/bin/sh

#location of 'gammu sms-client' binary
GAMMU="/usr/bin/gammu"
#Destination sms number
TO="69*****"
#Subject of the sms
SUBJECT="Alarm!!!Motion detected"

$GAMMU sendsms TEXT $TO -text "$SUBJECT"
```

Αν θα θέλαμε να συνδυάσουμε ταυτόχρονα την αποστολή e-mail και sms κατά την έναρξη των γεγονότων αυτό θα ήταν εφικτό με την δημιουργία ενός script (με την ονομασία mail_&_sms_motion.sh) το οποίο στην ουσία είναι συνένωση του sms_motion.sh με το mail_motion.sh που δημιουργήσαμε στην παράγραφο 3.4:

```
#!/bin/sh

#location of 'gammu sms-client' binary
GAMMU="/usr/bin/gammu"
#Destination sms number
TO="69*****"
#Subject of the sms
SUBJECT="Alarm!!!Motion detected"
$GAMMU sendsms TEXT $TO -text "$SUBJECT"
#location of 'mail' binary
MAIL="/usr/bin/mail"
#Destination e-mail address
TO="djsimko_@hotmail.com"
#Subject of the e-mail
```

```
SUBJECT="Alarm!!!Motion detected"
```

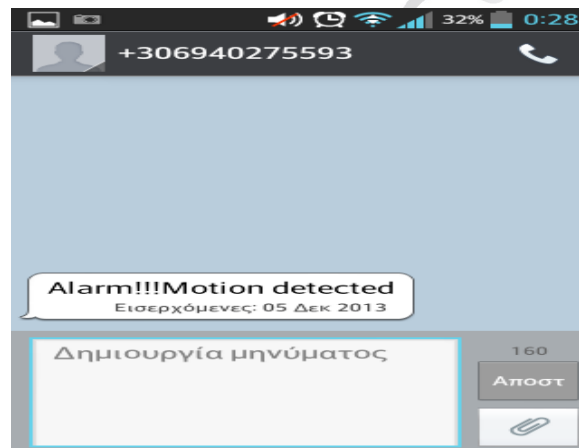
```
echo -e "This is an automated message generated by motion.\n\nMotion detected: $1\n\n" | $MAIL -s "$SUBJECT" $TO
```

Τελικά, καλούμε το script από το αρχείο configuration. Η ρύθμιση θα είναι:

```
# Command to be executed when an event starts. (default: none)

# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start /usr/local/etc/ mail_&_sms_motion
```

Έτσι, λοιπόν, ταυτόχρονα με την αποστολή e-mail, την επόμενη φορά που θα ανιχνευτεί κίνηση θα έχουμε και αποστολή SMS στον τηλεφωνικό αριθμό που έχει δηλωθεί στο script.



Εικόνα (3.5-1): Αποστολή γραπτού μηνύματος SMS σε κινητό τηλέφωνο έπειτα από ανίχνευση κίνησης

ΚΕΦΑΛΑΙΟ 4-ΠΡΟΧΩΡΗΜΕΝΕΣ ΡΥΘΜΙΣΕΙΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΚΑΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΜΕ ΤΟ ΧΡΗΣΤΗ

4.1 Ρυθμίσεις για το βασικό web interface

Το Motion έχει τη δυνατότητα να παρουσιάσει τις ληφθείσες εικόνες μέσω ενός web interface, αρκεί να οριστούν το όνομα χρήστη και ο κωδικός πρόσβασης για αυτό. Στο συγκεκριμένο project έχουν ενεργοποιηθεί τα default credentials όπου username=username και password=password. Είναι προφανές ότι μπορεί να αλλάξουν ανά πάσα στιγμή από τον διαχειριστή.

```
# HTTP Based Control

# TCP/IP port for the http server to listen on (default: 0 =
disabled)
control_port 8080

# Output for http server, select off to choose raw text plain
(default: on)
control_html_output on

# Authentication for the http based control. Syntax username:password
# Default: not defined (Disabled)
control_authentication username:password
```

Στη συνέχεια πρέπει να σωθεί το αρχείο και να προστεθεί το group motion στο όνομα χρήστη:

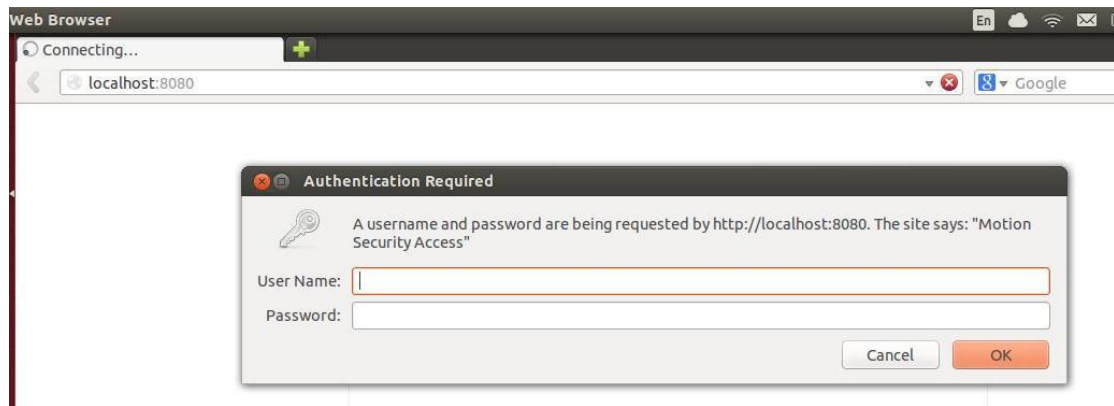
```
adduser USER motion
```

Για να λάβουν χώρα όλες οι μεταβολές θα πρέπει να γίνει επανεκκίνηση του Server, κάνοντας αλλαγή χρήστη και επανείσοδο.

Όταν γίνει εκκίνηση του daemon η διαχείριση του Motion μπορεί να γίνει μέσω του web interface αρκεί να οδηγήσουμε τον browser στην παρακάτω διεύθυνση:

```
http://localhost:8080
```

όπου για ενεργοποίηση αρκεί μόνο η πληκτρολόγηση των username και password.



Εικόνα (4.1-1): Αρχική καρτέλα του web interface και εισαγωγή ονόματος χρήστη και κωδικού πρόσβασης

Έτσι, αποκτάται πρόσβαση σε ένα βασικό αλλά χρήσιμο Motion interface από το οποίο μπορούν να γίνουν ρυθμίσεις στο αρχείο configuration file, παρατήρηση των τρεχουσών ρυθμίσεων καθώς και εκκίνηση ή παύση της ιδιότητας της ανίχνευσης κίνησης. Επιπρόσθετα, μπορεί να απεικονιστεί σε πραγματικό χρόνο η εικόνα που λαμβάνει η κάμερα από άλλο web interface, αν αυτή τη φορά πληκτρολογήσουμε στον browser τη διεύθυνση:

```
http://localhost:8081
```

```
# Live Webcam Server

# The mini-http server listens to this port for requests (default: 0
= disabled)
webcam_port 8081

# Quality of the jpeg (in percent) images produced (default: 50)
webcam_quality 50

# Output frames at 1 fps when no motion is detected and increase to
the
# rate given by webcam_maxrate when motion is detected (default: off)
webcam_motion off

# Maximum framerate for webcam streams (default: 1)
webcam_maxrate 1

# Restrict webcam connections to localhost only (default: on)
webcam_localhost on

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual webcam rate by desired
number of seconds
# Actual webcam rate is the smallest of the numbers framerate and
webcam_maxrate
webcam_limit 0
```

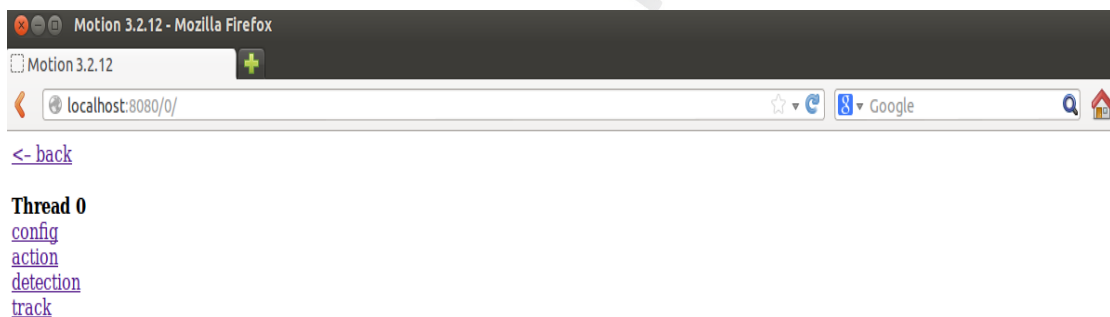
4.1.1 Δυνατότητες και επιλογές

Αμέσως μετά την πληκτρολόγηση της διεύθυνσης του localhost στον browser εμφανίζονται στην πρώτη καρτέλα τα ενεργά threads από της εν ενεργεία κάμερες ώστε να γίνει επιλογή της επιθυμητής.



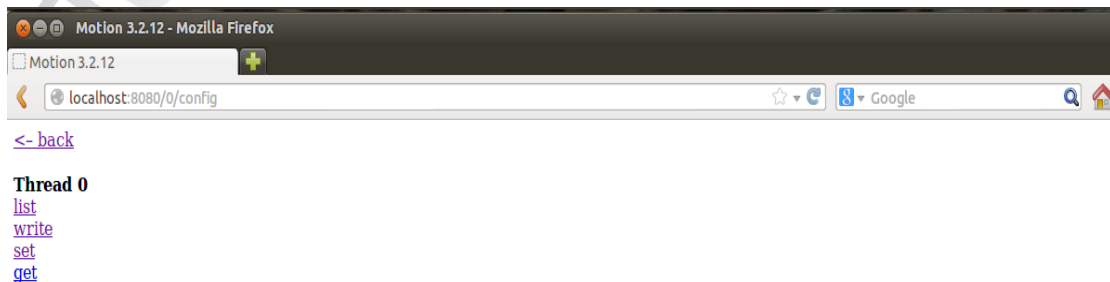
Εικόνα (4.1.1-1): Αρχική καρτέλα του web interface για επιλογή της επιθυμητής κάμερας

Επιλέγοντας τον σύνδεσμο "All" απεικονίζονται όλες οι πιθανές κατηγορίες (config,action,detection,track) που μπορεί να υπάρξει πρόσβαση από τον χρήστη σε όλα τα ενεργά threads και αναφέρονται στα επόμενα screenshots.



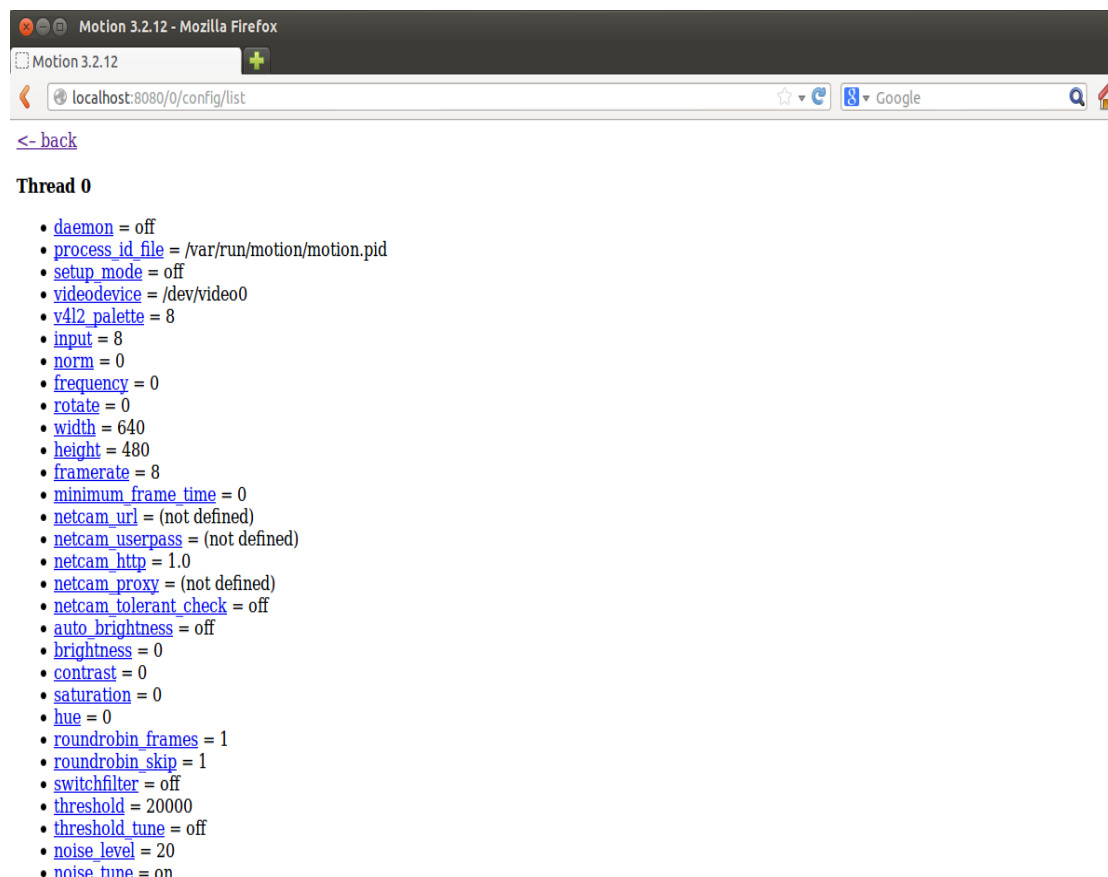
Εικόνα (4.1.1-2): Κατάλογος χαρακτηριστικών προς ρύθμιση και έλεγχο για την κάθε κάμερα

Πιο συγκεκριμένα, η πρώτη κατηγορία "config" είναι και η χρησιμότερη καθώς είναι η καρτέλα της παραμετροποίησης όπως αυτή γίνεται από το αρχείο configuration ή από τη γραμμή εντολών. Περιλαμβάνει τις επιλογές "list", "write", "set" και "get".



Εικόνα (4.1.1-3): Χαρακτηριστικά της επιλογής "config"

Με την επιλογή "list" εμφανίζεται η λίστα με όλες τις παραμέτρους όπως έχουν οριστεί από τον χρήστη από το αρχείο configuration.

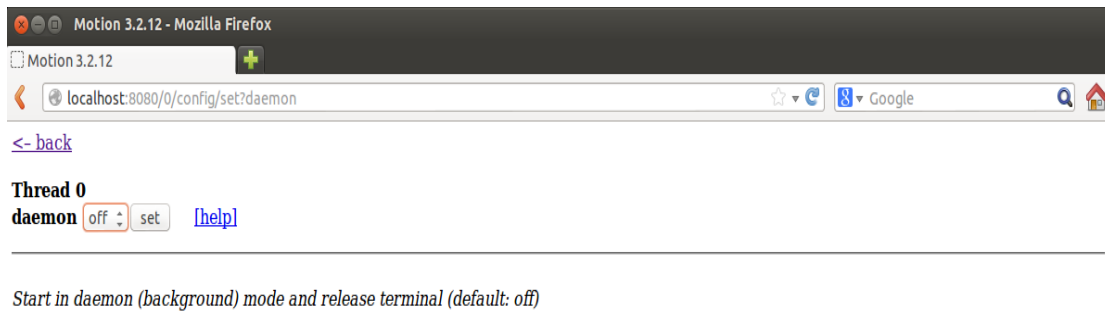


Εικόνα (4.1.1-4): Κατάλογος χαρακτηριστικών της επιλογής "list-config"

Ο χρήστης μπορεί, λοιπόν, να κάνει από εδώ οποιαδήποτε μεταβολή σε όποια παράμετρο είναι επιθυμητό απλά πατώντας πάνω και οριστικοποιώντας την μεταβολή της χρησιμοποιώντας την επιλογή "set".

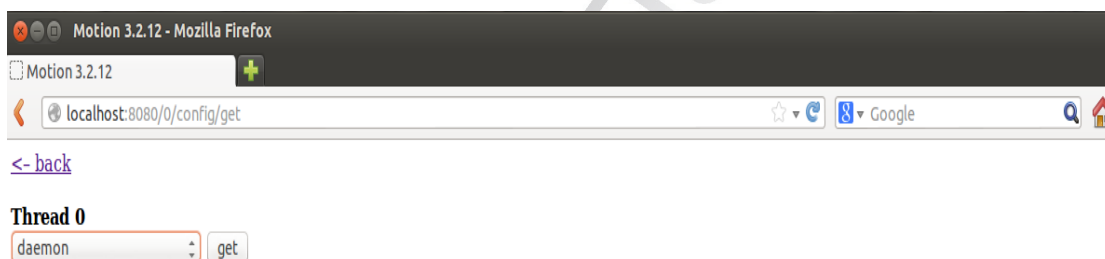


Εικόνα (4.1.1-5): Καρτέλα τροποποίησης παραμέτρων του "config"

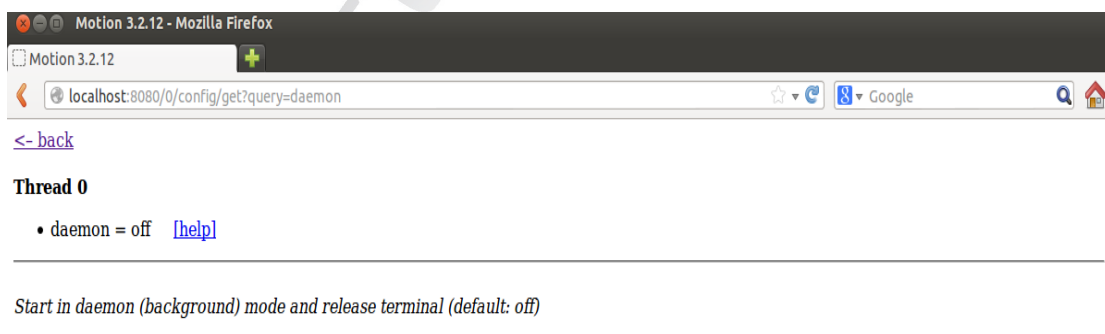


Εικόνα (4.1.1-6): Καρτέλα τροποποίησης παραμέτρων του "set"

Με την "get" απλά αντλείται πληροφορία για την τιμή που έχει κάποια επιλεχθείσα παράμετρος και με την "write" ουσιαστικά γίνεται εγγραφή του τρέχοντος configuration.

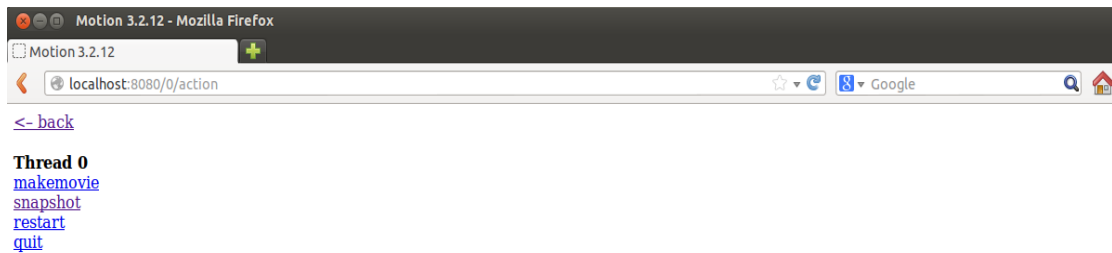


Εικόνα (4.1.1-7): Καρτέλα τροποποίησης παραμέτρων του "get"



Εικόνα (4.1.1-8): Καρτέλα τροποποίησης παραμέτρων του "get"(2)

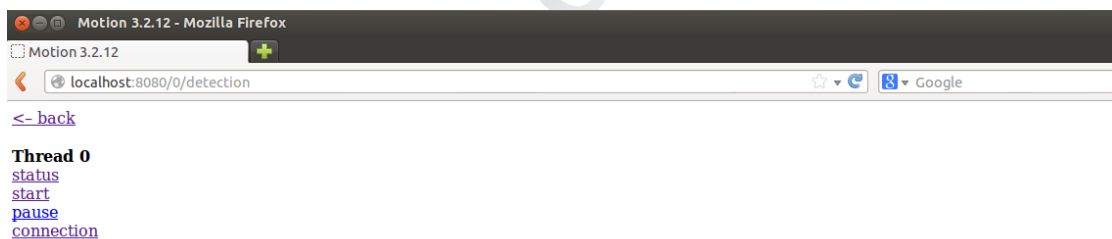
Όσον αφορά την δεύτερη επιλογή του κεντρικού μενού, "action" περιλαμβάνει τις επιλογές "makemovie", "snapshot", "restart" και "quit".



Εικόνα (4.1.1-9): Χαρακτηριστικά της επιλογής στην καρτέλα "action"

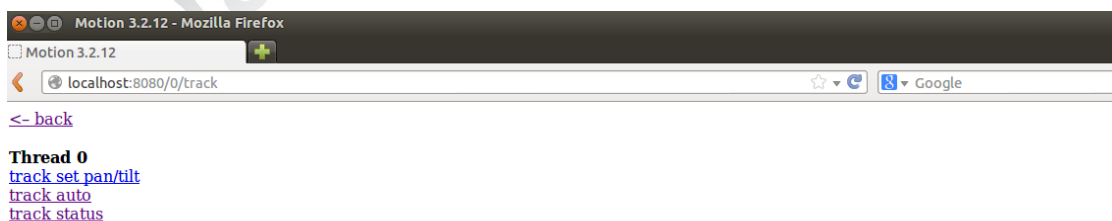
Με την "makemovie" δημιουργείται ένα μικρό αρχείο βίντεο της τρέχουσας στιγμής που λαμβάνεται από την κάμερα, με την "snapshot" λαμβάνεται ένα στιγμιότυπο της τρέχουσας εικόνας που λαμβάνεται από την κάμερα ενώ οι επιλογές "restart" και "quit" συντελούν στην επανεκκίνηση του Motion και στην έξοδο αντίστοιχα.

Η τρίτη επιλογή του κεντρικού μενού, "detection" περιλαμβάνει τις επιλογές "status", "start", "pause" και "connection". Οι "start" και "pause" επανεκκινούν η παύουν αντίστοιχα την δυνατότητα της ανίχνευσης κίνησης, η "status" απεικονίζει την κατάσταση στην οποία βρίσκεται η ανίχνευση, δηλαδή αν είναι σε start ή pause mode και η "connection" ελέγχει εάν η σύνδεση είναι ενεργή ή ανενεργή.



Εικόνα (4.1.1-10): Χαρακτηριστικά της επιλογής στην καρτέλα "detection"

Η τέταρτη και τελευταία επιλογή του κεντρικού μενού, "track" περιλαμβάνει τις επιλογές "track set pan/tilt", "track auto" και "track status".



Εικόνα (4.1.1-11): Χαρακτηριστικά της επιλογής στην καρτέλα "track"

4.2 Απομακρυσμένος έλεγχος και διαχείριση παραμέτρων

4.2.1 Από υπολογιστές του ίδιου τοπικού δικτύου

Είναι προφανές, ότι από τη στιγμή που υπάρχει η δυνατότητα της απεικόνισης και διαχείρισης του αρχείου configuration έστω και μέσω του βασικού web interface να υπάρχει και η δυνατότητα απομακρυσμένης πρόσβασης σε αυτό από υπολογιστές οι οποίοι βρίσκονται στο ίδιο τοπικό δίκτυο LAN. Έτσι, λοιπόν, είναι δυνατόν κάποιος supervisor να εισέλθει στο web interface από οποιονδήποτε υπολογιστή ο οποίος ανήκει και είναι συνδεδεμένος στο τοπικό δίκτυο, γεγονός που το καθιστά αρκετά χρήσιμο σε περιβάλλον π.χ. ενός νοσοκομείου όπου υπάρχουν αρκετοί κλινήρεις ασθενείς οι οποίοι παρακολουθούνται από ξεχωριστό υπολογιστή ο καθένας και απαιτείται η εποπτεία όλων από το αρμόδιο προσωπικό.

Για να συμβεί αυτό η μόνη μεταβολή που θα πρέπει να γίνει στο αρχείο configuration σε σχέση με την παράγραφο 4.1 είναι η απενεργοποίηση της απαγόρευσης των συνδέσεων εκτός του localhost που υπάρχει σαν προεπιλογή.

```
# Restrict control connections to localhost only (default: on)
control_localhost off
```

Στο τοπικό δίκτυο LAN που έλαβαν χώρα οι δοκιμές και η παραμετροποίηση του Motion, ο κεντρικός υπολογιστής (localhost) είχε διεύθυνση ip 192.168.1.41. Επομένως, από οποιονδήποτε υπολογιστή του τοπικού δικτύου (στο εν λόγω project υπήρχαν άλλοι 2 υπολογιστές με διευθύνσεις ip 192.168.1.40 και 192.168.1.44 αντίστοιχα) αρκεί μόνο η πληκτρολόγηση στον browser της διεύθυνσης ip του κεντρικού υπολογιστή (localhost) στον browser συνοδευόμενη από την port 8080, δηλαδή στην περίπτωση μας και οι 2 υπολογιστές του τοπικού δικτύου είχαν άμεση πρόσβαση στο web interface απλά πληκτρολογώντας στον browser τη διεύθυνση http://192.168.1.41:8080.

4.2.2 Από οπουδήποτε μέσω εικονικού ιδιωτικού δικτύου (VPN)

Είναι αυτονόητο ότι είναι ακόμα πιο χρήσιμη η απομακρυσμένη πρόσβαση από οποιονδήποτε υπολογιστή εκτός τοπικού δικτύου μέσω εικονικού ιδιωτικού δικτύου (Virtual Private Network VPN). Αυτό σημαίνει ότι μπορεί να γίνει έλεγχος της κατάστασης του κεντρικού υπολογιστή και κατ' επέκταση των κλινηρών ασθενών, από οποιονδήποτε υπολογιστή σε οποιοδήποτε σημείο του κόσμου μέσω υπολογιστή ο οποίος απλά διαθέτει σύνδεση στο internet. Από την μεριά του κεντρικού υπολογιστή ο οποίος εκτελεί το Motion, απαιτείται και εκεί σύνδεση στο internet, μια στατική ip και ένας router ο οποίος έχει τη δυνατότητα παραμετροποίησης ενός VPN.

Κατά τη διάρκεια υλοποίησης του συγκεκριμένου project, χρησιμοποιήθηκε ένας Cisco 837 ADSL Broadband Router και εκτός της διεύθυνσης ip του τοπικού δικτύου (η οποία όπως προτυπώθηκε παραπάνω ήταν η 192.168.1.41) δεσμεύτηκε και μία στατική ip για σύνδεση στο δίκτυο ευρείας περιοχής WAN η οποία ήταν η 94.70.241.142/32. Με την κατάλληλη παραμετροποίηση στον router και πάντα με γνώμονα την αυξημένη ασφάλεια καθώς πρόκειται για ευαίσθητα προσωπικά και ιατρικά δεδομένα επιτυγχάνεται η πρόσβαση στο web interface του localhost.

Παρακάτω, περιγράφονται οι σημαντικότερες ρυθμίσεις κατά την παραμετροποίηση του Cisco 837 ADSL Broadband Router:

Αρχικά, απόδοση dhcp, δήλωση primary και secondary dns server καθώς και πρωτοκόλλων που αφορούν τον πάροχο της σύνδεσης στο internet :

```
no ip dhcp use vrf connected
ip dhcp excluded-address 192.168.1.1 192.168.1.50
ip dhcp excluded-address 192.168.1.200 192.168.1.254
!
ip dhcp pool 1
  network 192.168.1.0 255.255.255.0
  default-router 192.168.1.1
  dns-server 195.170.0.1 192.170.2.2
!
!
ip cef
ip name-server 195.170.0.1
ip name-server 192.170.2.2
vpdn enable
!
vpdn-group 1
! Default PPTP VPDN group
  accept-dialin
  protocol pptp
  virtual-template 1
!
```

Ορισμός αυθεντικοποίησης και κρυπτογραφημένων credentials καθώς και περιορισμός παραμονής στη σελίδα για αυξημένη ασφάλεια:

```
username Service1 password 0 15 !55EU!44service133
!
crypto isakmp policy 10
  encr 3des
  hash md5
  authentication pre-share
  group 2
  lifetime 28800
```

Συνέχεια του προηγούμενου βήματος και δήλωση της στατικής ip για έξοδο στο WAN:

```
crypto isakmp key simoskokkovos123 address 94.70.241.142 no-xauth
!
!
crypto ipsec transform-set map-external esp-3des esp-md5-hmac
!
crypto map map-external 10 ipsec-isakmp
!
  set peer 94.70.241.142
  set transform-set map-external
  match address ipsec-acl-out
!
```

Ρυθμίσεις για interface atm0 και Ethernet0 για τον πάροχο του internet (username,password,encapsulation,vpi,vci) κατά τη σύνδεση στο DSLAM :

```
!  
interface Ethernet0  
 ip address 192.168.1.x 255.255.255.0  
 no shutdown  
 hold-queue 100 out  
!  
interface ATM0  
 no ip address  
 load-interval 30  
 no shutdown  
  
 no snmp trap link-status  
 no snmp ifindex persist  
 no atm ilmi-keepalive  
 dsl operating-mode auto  
 pvc 8/35  
  pppoe-client dial-pool-number 1  
!  
!  
interface FastEthernet1  
 duplex auto  
 speed auto  
 crypto map map-external  
!  
  
!  
interface Virtual-Template1  
 ip unnumbered Ethernet0  
 peer default ip address pool test  
 no keepalive  
 ppp encrypt mppe auto  
 ppp authentication pap chap ms-chap  
!  
interface Dialer1  
 no ip address  
 encapsulation ppp  
 no shutdown  
 dialer pool 1  
 dialer-group 1  
 ppp authentication chap pap callin  
 ppp pap sent-username o8lsj0@otenet.gr password 0 otel234!  
 ppp ipcp dns request accept  
 ppp ipcp route default  
!
```

Απόδοση συγκεκριμένων ip (μέσω συγκεκριμένου pool για αποφυγή conflict σε περίπτωση χρησιμοποίησης περισσότερων δικτυακών συσκευών) του εισερχόμενου υπολογιστή στο LAN :

```
!  
ip local pool test 192.168.1.246 192.168.1.248  
ip route 0.0.0.0 0.0.0.0 dialer1  
no ip http server  
no ip http secure-server  
!
```

4.3 Εξέλιξη του βασικού web interface

Η χρήση του βασικού Web Interface καθώς και η δυνατότητα της απομακρυσμένης πρόσβασης συντέλεσε στην ανάγκη δημιουργίας ενός εξελιγμένου Web Interface και για αισθητικούς αλλά κυρίως για πρακτικούς λόγους. Μέχρι τώρα στην παράγραφο 4.1 είδαμε ότι μέσω μιας διεύθυνσης (<http://localhost:8080>) μπορούμε να έχουμε έλεγχο του αρχείου configuration και μέσω άλλης διεύθυνσης σε διαφορετική port (<http://localhost:8081>) μπορούμε να έχουμε live παρακολούθηση των καμερών. Με την ύπαρξη μίας και μόνο κεντρικής σελίδας η οποία θα βρίσκεται σε έναν εγκατεστημένο web server είναι δυνατή η είσοδος σε ένα βελτιστοποιημένο γραφικό περιβάλλον για αισθητικούς λόγους. Όσον αφορά το πρακτικό κομμάτι, ο client της ιστοσελίδας κάθε φορά που θα εισέρχεται σε αυτή θα έχει τις επιλογές της διαχείρισης (Management) καθώς και της παρακολούθησης (Viewing). Μέσω του Management θα εισέρχεται με τον ίδιο τρόπο στις επιλογές του αρχείου configuration όπως περιγράφηκε στην παράγραφο 4.1. Επίσης, μέσω του Viewing έχει τη δυνατότητα να παρακολουθήσει τα στιγμιότυπα που λαμβάνουν οι κάμερες, να έχει πλήρη πρόσβαση για ανάγνωση της βάσης δεδομένων που δημιουργήθηκε στην παράγραφο 3.3 καθώς και την δυνατότητα να δει τις εικόνες που έχουν ληφθεί από τις κάμερες και αν θελήσει να σβήσει κάποια ή όλες από αυτές. Όλες αυτές οι επιλογές κρίνονται φοβερά χρήσιμες αν αναλογιστούμε και την ύπαρξη της απομακρυσμένης πρόσβασης, διότι είναι πλήρης και συνάμα απλούστατη η εποπτεία και η διαχείριση των ασθενών που παρακολουθούνται από το αρμόδιο προσωπικό.

Όλα τα παραπάνω χαρακτηριστικά μπορούν να υλοποιηθούν με τη χρήση των php,mysql και apache.

Σε πρώτο χρόνο γίνεται εγκατάσταση του apache:

```
sudo service apache2 restart
```

Επιπρόσθετα εγκατάσταση της php5:

```
apt-get install php5 libapache2-mod-php5
```

και στη συνέχεια επανεκκίνηση του apache:

```
sudo service apache2 restart
```

Η θέση των αρχείων web βρίσκεται by default στη θέση /var/www. Δημιουργούμε ένα PHP αρχείο με το όνομα info.php σε αυτή τη θέση και το ανοίγουμε με τον browser. Αυτό το αρχείο εμπεριέχει χρήσιμες πληροφορίες σχετικά με την εγκατάσταση της PHP, εκδόσεις, options κτλ.

```
sudo gedit /var/www/info.php
```

Για να γίνει ενεργοποίηση των δυνατοτήτων της MySQL στην PHP, μπορούμε να εγκαταστήσουμε το package php5-mysql. Συνάμα, θα ήταν καλύτερα αν γινόταν εγκατάσταση και κάποιον επιπλέον PHP5 modules. Επομένως, εγκαθιστούμε:

```
sudo apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear
php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps
php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc
php5-xsl
```

ολοκληρώνοντας με επανεκκίνηση του apache:

```
/etc/init.d/apache2 restart
```

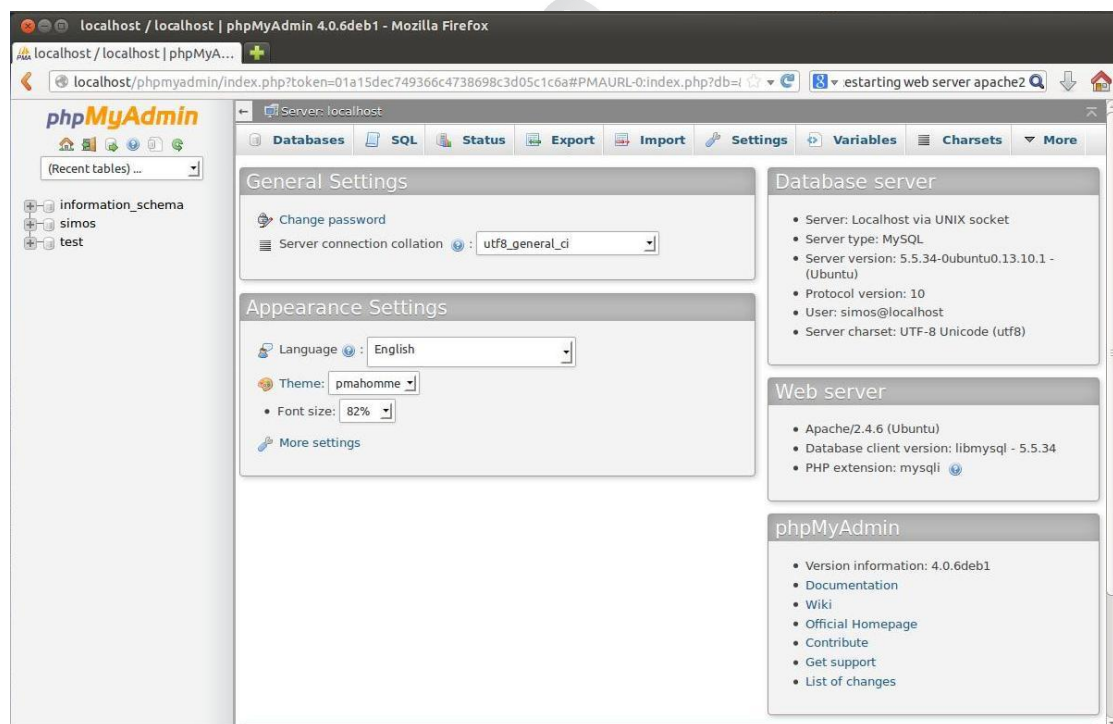
Επιπρόσθετα, είναι πολύ χρήσιμη η εγκατάσταση του option phpMyAdmin το οποίο είναι μια διεπαφή μέσω της οποίας μπορεί να γίνει διαχείριση των βάσεων δεδομένων της MySQL. Για να γίνει η εγκατάστασή του, αρκεί η παρακάτω εντολή:

```
apt-get install phpmyadmin
```

Για να γίνει παραμετροποίηση του phpMyAdmin αρκεί η παρακάτω εντολή:

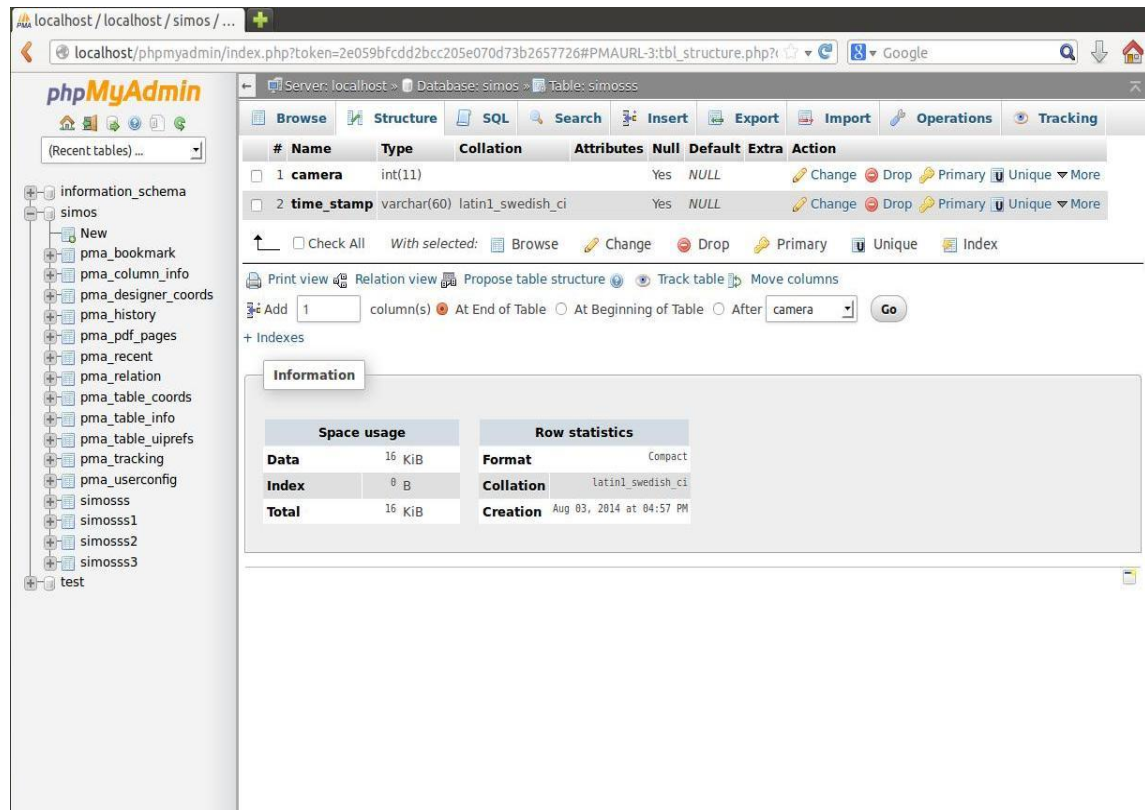
```
sudo dpkg-reconfigure phpmyadmin
```

όπου επιλέγουμε ρυθμίσεις για apache2 και δημιουργώντας τα απαραίτητα username και password για την είσοδο στην εφαρμογή. Έτσι, αφού εισάγουμε τους κωδικούς ασφαλείας καταλήγουμε στην αρχική καρτέλα του phpMyAdmin όπου είναι ορατές όλες οι βάσεις δεδομένων που έχουν δημιουργηθεί μέσω της MySQL στο σύστημα :



Εικόνα (4.3-1): Αρχική καρτέλα του phpMyAdmin

Έπειτα, επιλέγοντας την βάση δεδομένων που μας απασχολεί η οποία όπως είδαμε στην παράγραφο 3.3.1 ονομάζεται *simos*, καταλήγουμε στον πίνακα που έχουμε δημιουργήσει με το όνομα *simosss* και αποκτούμε πρόσβαση στις μεταβλητές του:



Εικόνα (4.3-2): Επιλογές της βάσης δεδομένων και των πινάκων της στο phpMyAdmin

Αφού τελειώσουν όλες οι παραμετροποιήσεις στο λειτουργικό σύστημα, το επόμενο βήμα είναι η συγγραφή κώδικα για το εξελιγμένο web interface, χρησιμοποιώντας html και php. Η αρχική οθόνη του είναι γραμμένη σε html με την ονομασία *webinterface.html*:

```
<html>
  <head>
    <title>Motion Detection</title>
    <script type='text/javascript'>/*! [CDATA[
      function popupwindow(url, title, w, h) {
        var left = (screen.width / 2) - (w / 2);
        var top = (screen.height / 2) - (h / 2) - 50;
        return window.open(url, title, 'toolbar=no,
location=no, directories=no, status=no, menubar=no, scrollbars=no,
resizable=no, copyhistory=no, width=' + w + ', height=' + h + ',
top=' + top + ', left=' + left);
      } // ]>
    </script>
    <style>
      .cssmenu {
```

```

        background: #333;
        list-style: none;
        margin: 0;
        padding: 0;
        width: 12em;
    }
    .cssmenu li {
        font: 90% 'Lucida Sans Unicode', 'Bitstream Vera
Sans', 'Trebuchet Unicode MS', 'Lucida Grande', Verdana, Helvetica,
sans-serif;
        margin: 0;
        padding: 0;
        list-style: none;
    }
    .cssmenu a {
        background: #333;
        border-bottom: 1px solid #393939;
        color: #ff0000;
        display: block;
        margin: 0;
        padding: 8px 12px;
        text-decoration: none;
        font-weight: normal;
    }
    .cssmenu a:hover {
        background: #2580a2 url('hover.gif') left center no-
repeat;
        color: #fff;
        padding-bottom: 8px;
    }
    .cssmenu li.header{
        color: #FFFFFF;
        font-size: 1.4em;
    }
</style>
</head>

<body background="/Computer_eye2.jpg" no-repeat >

    <div id="resize"></div></br>

    <h1>Motion Detection</h1>

    <div class='cssmenu'>
        <ul>
            <li class="header"><span>Management</span></li>
            <li><a
href="http://localhost:8080/">Configure</a></li>
        </ul>
    </div>

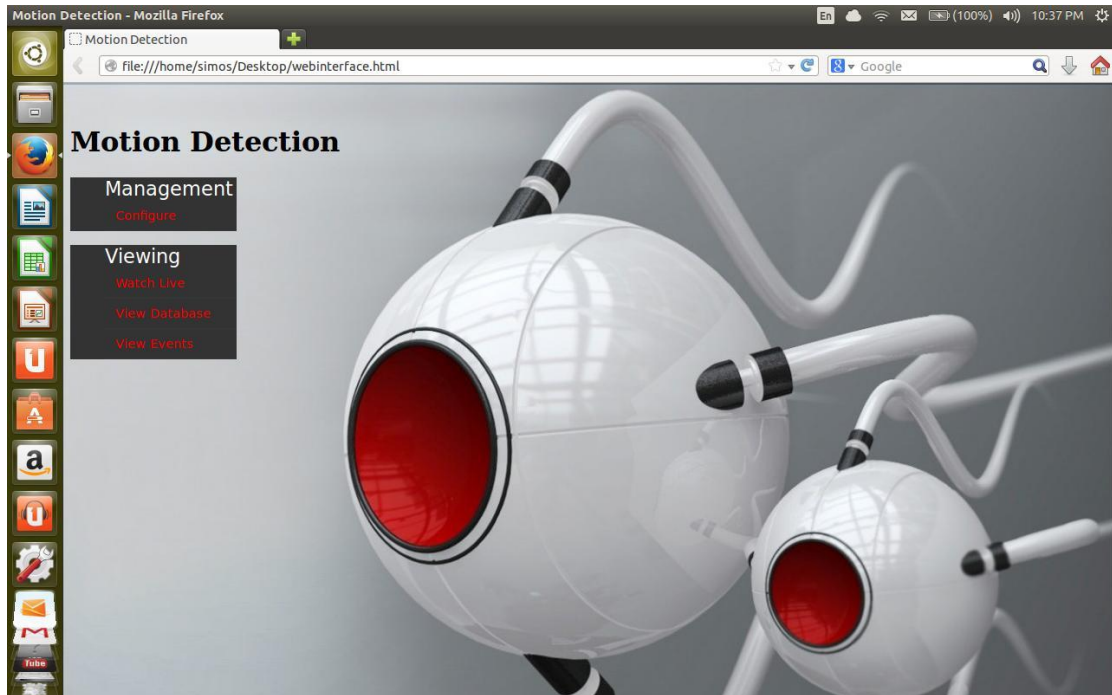
    <div class='cssmenu'>
        <ul>
            <li class="header"><span>Viewing</span></li>
            <li><a href="multiplecameras.html">Watch
Live</a></li>
            <li><a href="database.php">View Database</a></li>
            <li><a
href="javascript:popupwindow('imageviewer/imageviewer.php','Image
Viewer',1024,768)">View Events</a></li>

```

```

        </ul>
      </div>
    </body>
  </html>

```



Εικόνα (4.3-3): Αρχική καρτέλα του εξελεγμένου web interface

Όπως προτυπώθηκε, η πρώτη δυνατότητα του Viewing είναι η live παρακολούθηση της εικόνας που λαμβάνουν οι κάμερες ("Watch Live"). Για το λόγο αυτό δημιουργήθηκε σε html το αρχείο `multiplecameras.html` το οποίο καλείται μέσω του `webinterface.html`. Στο εν λόγω αρχείο καλούμε τις ports που προσδιορίζουν τις κάμερες όπως ορίστηκε στην παράγραφο 3.1.1 και προσθέτουμε ένα αρκετά χρήσιμο slider ώστε ο χρήστης να μπορεί να εστιάσει σε οποιοδήποτε σημείο ενδιαφέροντος.

```

<!DOCTYPE html>
<html>
<head>
<link
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="stylesheet" type="text/css"/>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"><
/script>
<script
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"></script>
<style type="text/css">
#resize {
margin-top: 0px;
margin-bottom: 20px;
display: inline-block;
}
#slider {

```



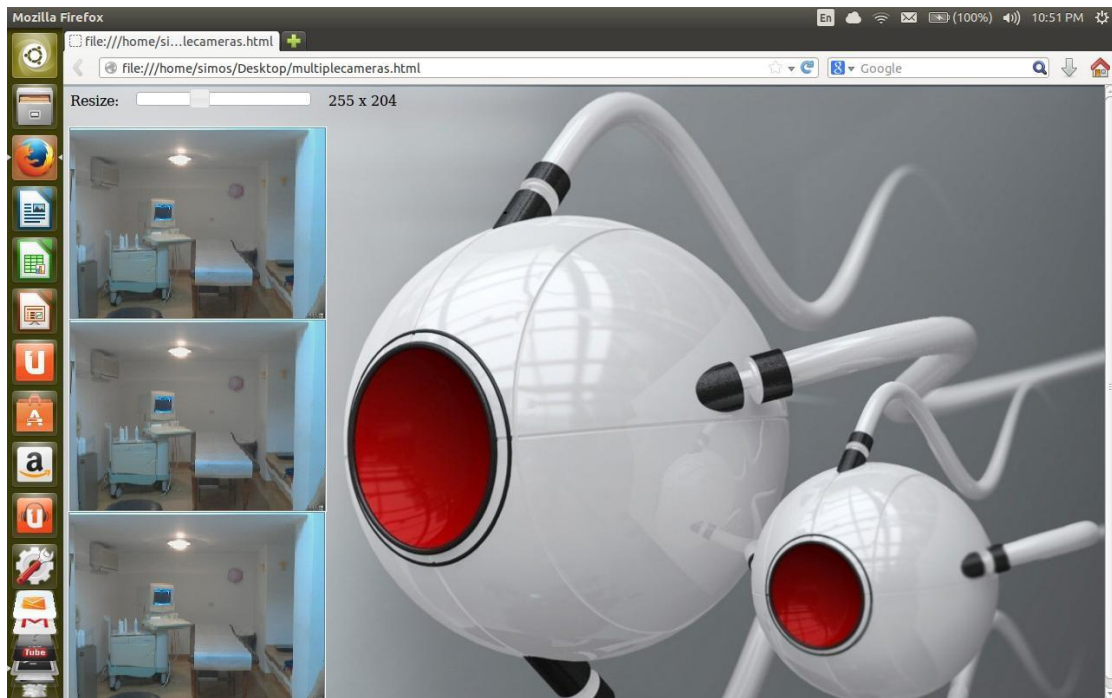
```

        margin-top: 0px;
        margin-bottom: 0px;
        margin-left: 20px;
        width: 200px;
        display: inline-block;
    }
    #sliderValue {
        margin-left: 20px;
        display: inline-block;
    }
</style>
<script>
$(document).ready(function() {
    $("#slider").slider({
        'max': 720,
        'min': 160,
        'step': 1,
        'value': 360,
        slide: function (event, ui) {
            var height = parseInt (ui.value / 5) * 4;
            $('#sliderValue').text (ui.value + ' x ' +
height);
        },
        change: function (event, ui) {
            var height = parseInt (ui.value / 5) * 4;
            $('#cam1').attr ('width', ui.value);
            $('#cam1').attr ('height', height);
            $('#cam2').attr ('width', ui.value);
            $('#cam2').attr ('height', height);
            $('#cam3').attr ('width', ui.value);
            $('#cam3').attr ('height', height);
        }
    });
});
</script>
</head>
<body background="/Computer_eye2.jpg" no-repeat >

<div id="resize">Resize:</div><div id="slider"></div><div
id="sliderValue">255 x 204</div></br>

<li><img id=cam1 src=http://localhost:8081/ width=290
border="0"></li>
<li><img id=cam2 src=http://localhost:8082/ width=290
border="0"></li>
<img id=cam3 src=http://localhost:8083/ width=290 border="0">
</body>
</html>

```



Εικόνα (4.3-4): Επιλογή "Watch Live" από το εξελεγμένο web interface

Μεταβαίνοντας στη δεύτερη δυνατότητα του Viewing είναι η ανάγνωση των συμβάντων που έχουν δημιουργηθεί από τη βάση δεδομένων. Για το σκοπό αυτό δημιουργήθηκε σε php το αρχείο database.php το οποίο καλείται μέσω του webinterface.html. Εκμεταλλευόμενοι, λοιπόν, τη δυνατότητα της php να συνδεθεί με την MySQL καθώς και τις ρυθμίσεις που έγιναν στις προηγούμενες παραγράφους έχουμε την εικόνα της βάσης δεδομένων με το πάτημα του πλήκτρου "View Database".

```
<?php
$con=mysqli_connect("localhost","root","simos","simos");
// Check connection
if (mysqli_connect_errno())
{
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$result = mysqli_query($con,"SELECT * FROM simosss");

echo "<table border='1'>
<tr>
<th>Camera</th>
<th>Timestamp</th>
</tr>";

while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['camera'] . "</td>";
    echo "<td>" . $row['time_stamp'] . "</td>";
    echo "</tr>";
}
```

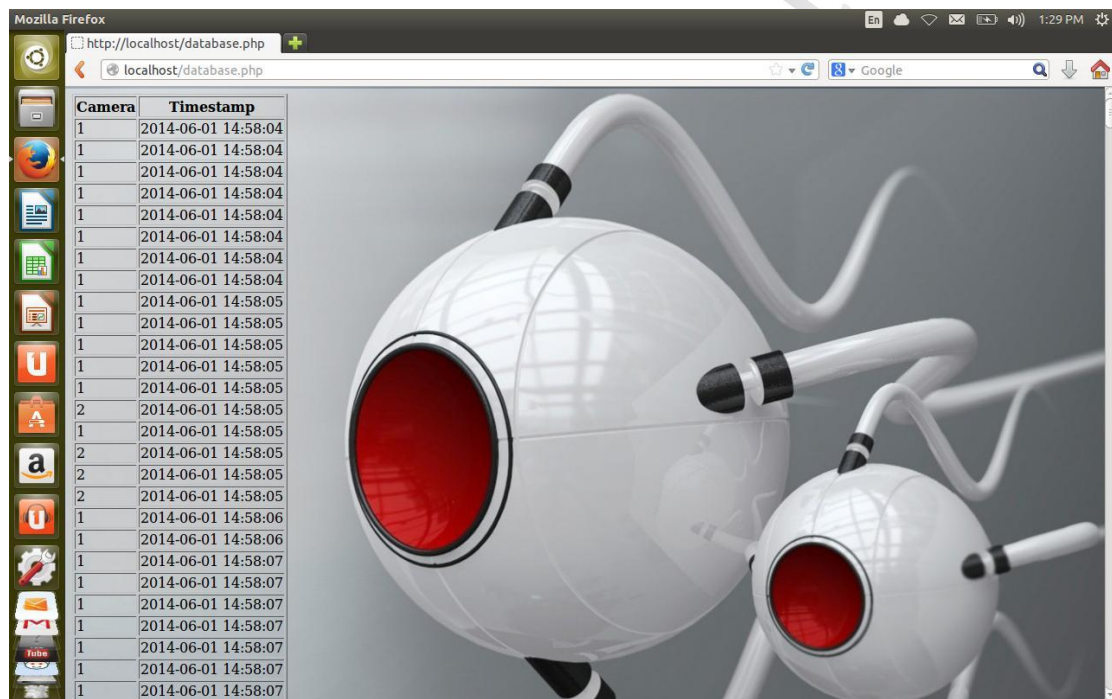
```

    }
echo "</table>";

mysqli_close($con);
?>
<!DOCTYPE html>
<html>
    <body>
<body background="/Computer_eye2.jpg" no-repeat >

        <div id="resize"></div></br>
        </body>
</html>

```



Εικόνα (4.3-5): Επιλογή "View Database" από το εξελιγμένο web interface

Η τρίτη και αρκετά χρήσιμη δυνατότητα του Viewing είναι η προβολή των εικόνων που έχουν ληφθεί κατά τη διάρκεια των συμβάντων και έχουν αποθηκευθεί στο φάκελο με το όνομα "motion" στην διεύθυνση /var/www/imageviewer/motion (όπως προτυπώθηκε στην παράγραφο 3.1) από όπου είναι δυνατή η ανάγνωση από τον apache2 web server. Συγκεκριμένα, δημιουργήθηκε ο φάκελος "imageviewer" για να εμπεριέχει εκτός των εικόνων που έχουν ληφθεί και όλα τα πηγαία αρχεία για την προβολή τους. Αναλυτικότερα, ο εν λόγω φάκελος αποτελείται από τα αρχεία getimages.php , imageviewer.php (παρατίθενται παρακάτω) και κάποια μικρά αρχεία png που είναι τα buttons του viewer.

Για να γίνεται η άντληση των εικόνων, λοιπόν, υπήρξε η συγγραφή του αρχείου getimages.php.

```

<?php

$dir = '/var/www/imageviewer/motion';

function scand() {
    global $dir;
    $files = scandir($dir);
    sort($files);
    reset($files);
    return $files;
}

function getImages() {
    global $dir;

    $output = '<div id="fader">' . PHP_EOL;
    $files = scand($dir);

    foreach ($files as $file) {
        if ($file === '.' || $file === '..') {
            continue;
        }
        $output.='' . PHP_EOL;
    }
    echo $output . '</div>' . PHP_EOL;
}

function deleteImage($imageName) {
    global $dir;
    if (file_exists($dir . '/' . $imageName)) {
        $old_umask = umask(0);
        @chmod($dir . '/' . $imageName, 0757);
        unlink($dir . '/' . $imageName);
        umask($old_umask);
    }
}

?>

```

Για τη δημιουργία του viewer συντάχθηκε σε php το αρχείο imageviewer.php το οποίο καλείται με το πάτημα του πλήκτρου "View Events" από το web interface.html. Με αυτόν τον τρόπο, ο χρήστης έχει τη δυνατότητα να παρακολουθήσει όλες τις εικόνες που έχουν εξαχθεί από το Motion (με επιλογές-πλήκτρα "previous" και "next") καθώς και να σβήσει από το φάκελο (με επιλογή "delete") όσες δεν επιθυμεί.

```

<?php
include "getimages.php";

$imageName = filter_input(INPUT_POST, 'delete',
FILTER_SANITIZE_STRING);

if ($imageName) {
    deleteImage($imageName);
}

```

```

?>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
charset=UTF-8">
    <title> galerry</title>
    <script type='text/javascript'
src='http://code.jquery.com/jquery-1.10.1.min.js'></script>
    <style type='text/css'>
      #fader {
        position: relative;
        width: 100%;
      }
      .button {
        display: inline-block;
        width: 32px;
        height: 32px;
        margin: 5px;
      }
      #next {background:url('next.png') no-repeat;}
      #next:hover {background:url('next_p.png') no-repeat;}
      #prev {background:url('prev.png') no-repeat;}
      #prev:hover {background:url('prev_p.png') no-repeat;}
      #delete_but {
        background:url('delete.png') no-repeat;
        width: 32px;
        height: 32px;
        margin: 5px;
      }
      #delete_but:hover {background:url('delete_p.png') no-
repeat;}
      #controls{
        text-align: center;
        top: -48px;
        position: relative;
        background: rgba(230, 230, 230, 0.10);
      }
      body{
        background: black;
      }
    </style>
    <script type='text/javascript'>/*! [CDATA[
      $(window).load(function() {
        $(function() {
          $('#fader img:not(:first)').hide();
          $('#fader img').css('position', 'absolute');
          $('#fader img').css('top', '0px');
          $('#fader img').css('left', '50%');

          $('#fader img').each(function() {
            var img = $(this);
            $('<img>').attr('src',
$(this).attr('src')).load(function() {
              img.css('margin-left', -this.width / 2 +
'px');
            });
          });
        });
      });
    </script>
  </body>
</html>

```

```

        $('#fader').height($('#fader
img').first().height());

        function fadeNext() {
            $('#fader
img').first().fadeOut().appendTo($('#fader'));
            $('#fader img').first().fadeIn();
            $('#fader').height($('#fader
img').first().height());
        }

        function fadePrev() {
            $('#fader img').first().fadeOut();
            $('#fader
img').last().prependTo($('#fader')).fadeIn();
            $('#fader').height($('#fader
img').first().height());
        }

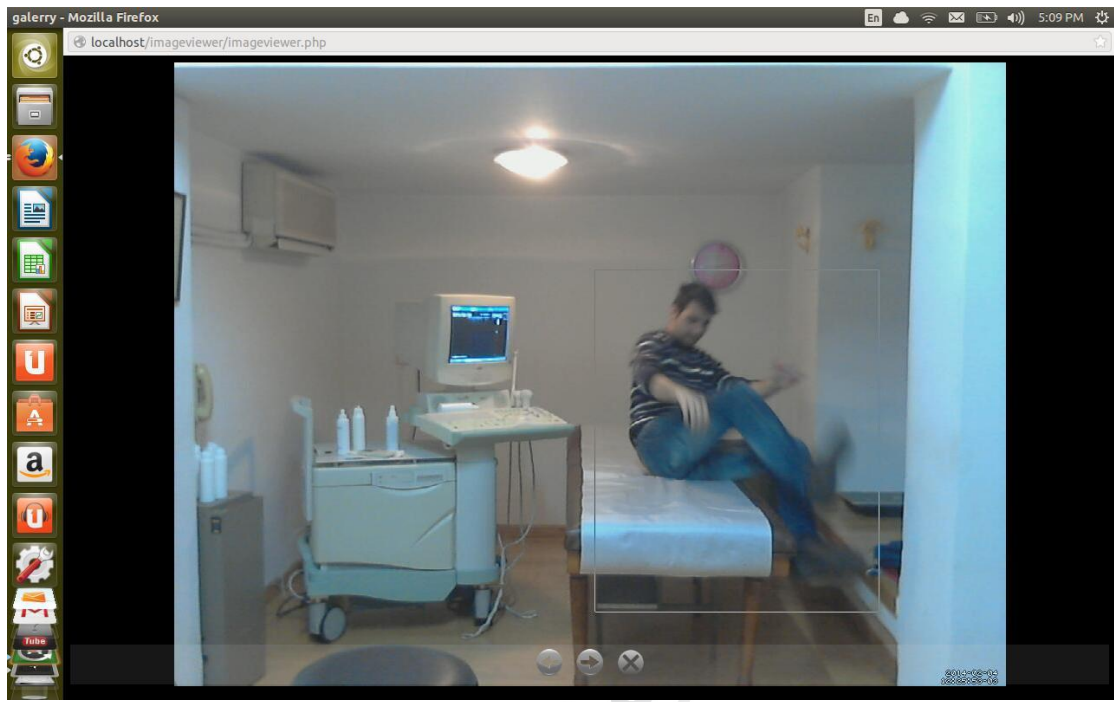
        $("#faderForm").submit(function(event) {
            if (confirm('Are you sure you want to delete
' + $('#fader img').first().attr('name') + '?')) {
                $('#delete').val($('#fader
img').first().attr('name'));
                return;
            } else {
                event.preventDefault();
            }
        });

        $('#delete_but').click(function() {
            $('#faderForm').submit();
        });

        $('#next').click(function() {
            fadeNext();
        });

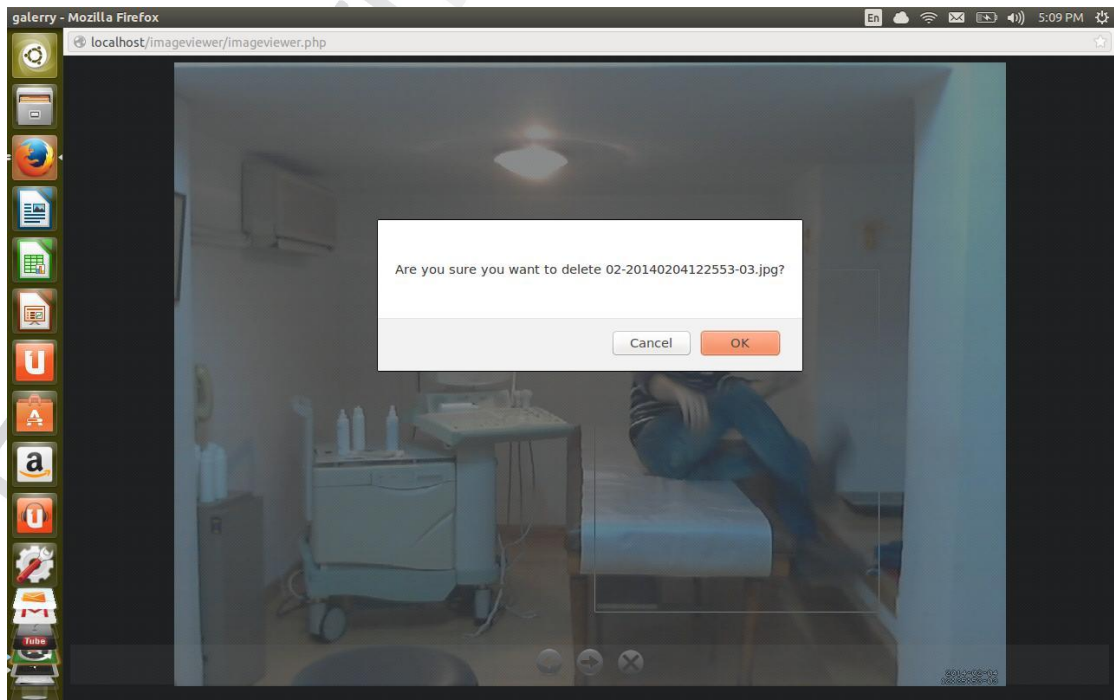
        $('#prev').click(function() {
            fadePrev();
        });
    });
}); //]]>
</script>
</head>
<body>
    <form method="post" id="faderForm">
        <?php getImages() ?>
        <div id="controls">
            <div class="button" id="prev" value="<"></div>
            <div class="button" id="next" value=">"></div>
            <div class="button" id="delete_but"
value="Delete"></div>
            <input type="hidden" name="delete" id="delete"
value="Delete">
        </div>
    </form>
</body>
</html>

```



Εικόνα (4.3-6): Επιλογή "View Events" από το εξελιγμένο web interface και προβολή των εικόνων

Από την παραπάνω εικόνα (Εικόνα 4.3-6) είναι ορατές οι επιλογές "previous" και "next" από τον viewer καθώς και η επιλογή "delete". Κατά την επιλογή "delete" υπάρχει και μήνυμα επιβεβαίωσης πριν γίνει η διαγραφή (Εικόνα 4.3-7).



Εικόνα (4.3-7): Επιβεβαίωση διαγραφής εικόνας από τον viewer

ΚΕΦΑΛΑΙΟ 5-ΠΡΟΣΘΗΚΗ ΕΥΦΥΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΣΤΟ ΣΥΣΤΗΜΑ ΜΕ ΣΤΟΧΟ ΤΗ ΔΙΑΚΡΙΤΙΚΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΤΟΥ ΑΣΘΕΝΟΥΣ

Στο κεφάλαιο αυτό, έχοντας ήδη αναπτύξει αρκετά επιπλέον χαρακτηριστικά που κάνουν το Motion πιο αξιοποιήσιμο, γίνεται προσπάθεια να αποδοθούν και κάποια με περισσότερο ευφυή χαρακτήρα τα οποία εκτός του γεγονότος ότι το καθιστούν πιο εντυπωσιακό έχουν και αρκετά χρήσιμη λειτουργία, συντελώντας στον τελικό στόχο, την αποφυγή ψευδών συναγερωμών. Για το σκοπό αυτό, προστίθενται οι δυνατότητες για ενημέρωση του Motion για αυτόματη παύση και εκκίνηση ανάλογα με περιπτώσεις που αναφέρονται τις επόμενες παραγράφους, για συνεκτίμηση του αποτελέσματος των δεδομένων από πολλαπλές κάμερες καθώς και για εκμάθηση του συστήματος της συνηθισμένης συμπεριφοράς στις κινήσεις του ασθενούς και δημιουργία συναγερωμού μόνο σε διαφορετική περίπτωση.

5.1 Αυτόματη παύση/επανεκκίνηση του Motion

Στις συνθήκες παρακολούθησης κλινηρών ασθενών είναι φυσικό να προστεθεί η παρουσία νοσηλευτικού προσωπικού στο χώρο παρακολούθησης κάποιες ώρες, είτε προγραμματισμένα είτε επειγόντως καθώς και επισκέψεις συγγενικών προσώπων των νοσηλευόμενων. Η παρουσία των προαναφερθέντων ατόμων στο χώρο παρακολούθησης φυσικά θα δημιουργήσει συναγερωμούς από το σύστημα. Στόχος είναι το σύστημα να είναι ενημερωμένο για τις εν λόγω παρουσίες και να μην αντιδρά, θεωρώντας την ανίχνευση κίνησης στο χώρο εκείνη τη στιγμή φυσιολογική.

Για να επιτευχθεί αυτό, θα πρέπει να γίνει στο σύστημα διαχωρισμός των προγραμματισμένων από των μη προγραμματισμένων παρουσιών στο χώρο επιτήρησης. Έτσι, η μια κατηγορία αποτελείται από το προγραμματισμένο επισκεπτήριο των ασθενών από συγγενικά τους πρόσωπα σε συγκεκριμένες ημέρες και ώρες ή την προγραμματισμένη εξέταση του νοσηλευτικού προσωπικού, ενώ η δεύτερη από την επείγουσα επίσκεψη του προσωπικού ή την μη προγραμματισμένη επίσκεψη συγγενικού προσώπου. Και στις δύο περιπτώσεις, το σύστημα μπορεί να είναι ενήμερο και να μην γεννά συναγερωμούς παύοντας προσωρινά την διαδικασία της ανίχνευσης.

Όλες οι ρυθμίσεις που έχουν σχέση με τον χρόνο (time schedule) πραγματοποιούνται με τη βοήθεια του Linux Crontab το οποίο είναι ένα χρήσιμο utility του Linux ικανό ώστε να προγραμματίζει μια εργασία ρουτίνας στο παρασκήνιο σε συγκεκριμένη ημέρα και ώρα. Το format για την συγγραφή και ρύθμισή του περιγράφεται παρακάτω:

MIN HOUR DOM MON DOW CMD

Στον επόμενο Πίνακα αναφέρονται όλες οι μεταβλητές που μπορούν να ρυθμιστούν στο Linux Crontab και οι επιτρεπόμενες τιμές που μπορούν να λάβουν:

Πεδίο	Περιγραφή	Επιτρεπόμενες Τιμές
MIN	Λεπτά	0-59
HOUR	Ώρες	0-23
DOM	Ημέρα του μήνα	1-31
MON	Μήνας	1-12
DOW	Ημέρα της εβδομάδας	0-6
CMD	Εντολή	Οποιαδήποτε εντολή χρήστη

Πίνακας (5.1-1): Μεταβλητές και επιτρεπόμενες τιμές του Linux Crontab

Για να αποκτήσουμε πρόσβαση στο αρχείο configuration του Linux Crontab και να προβούμε στις επιθυμητές ρυθμίσεις είναι προτιμότερο να το προγραμματίσουμε ώστε να αναπαράγει ένα αρχείο μορφής σημειωματάριου με το gedit.

Αρκεί, λοιπόν, η πληκτρολόγηση της εντολής στο terminal:

```
EDITOR=gedit crontab -e
```

ώστε να εμφανιστεί και να παραμετροποιηθεί ανάλογα με τις απαιτήσεις του χρήστη το αρχείο configuration του Linux Crontab.

5.1.1 Κατά τη διάρκεια προγραμματισμένων επισκέψεων στο χώρο βιντεοεπιτήρησης

Αρχικά, έχουμε τις προγραμματισμένες επισκέψεις νοσηλευτικού προσωπικού και συγγενών στο χώρο όπου γίνεται η βιντεοεπιτήρηση. Έστω ότι τα ωράρια έχουν ως εξής:

Εξέταση νοσηλευτικού προσωπικού: 8.00-9.00 και 17.00-18.00 καθημερινά και Σαββατοκύριακα 8.00-9.00.

Επισκεπτήριο Ασθενών: 18.00-20.00 Δευτέρα, Τετάρτη και Παρασκευή.

Για την παύση/εκκίνηση της ανίχνευσης κίνησης αρκεί να εκμεταλλευτούμε την ιδιότητα του βασικού web interface για ενεργοποίηση/απενεργοποίηση της ανίχνευσης κίνησης όπως περιγράφηκε στην παράγραφο 4.1.1 και Εικόνα 4.1.1-10 και απλά να καθοδηγήσουμε τον pointer εκεί. Απενεργοποιούμε, επίσης, και την ιδιότητα αποστολής email (`2>&1 >/dev/null`).

Επομένως, για την παύση θα έχουμε:

```
wget -q --delete-after http://localhost:8080/0/detection/pause 2>&1 >/dev/null
```

Και αντίστοιχα, για την επανεκκίνηση θα έχουμε:

```
wget -q --delete-after http://localhost:8080/0/detection/start 2>&1 >/dev/null
```

Στο αρχείο configuration του Linux Crontab οι παραπάνω ημέρες και ώρες απεικονίζονται και ρυθμίζονται ως εξής, ενώ προστίθενται και οι εντολές για παύση/εκκίνηση:

```
# m h dom mon dow    command
00 08-09 * * * wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 17-18 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 18-20 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
01 09 * * * wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 18 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 20 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
```

5.1.2 Σε περίπτωση έκτακτης εισόδου στο χώρο βιντεοεπιτήρησης

Όσον αφορά στις μη προγραμματισμένες και επείγουσες επισκέψεις είτε από νοσηλευτικό προσωπικό είτε από συγγενείς των ασθενών, η παύση της ανίχνευσης γίνεται με τη βοήθεια συσκευών που μπορούν να διαθέτουν ρύθμιση για σύνδεση στο δίκτυο και διεύθυνση ip (π.χ. PDAs, smartphones). Συγκεκριμένα, οποιοσδήποτε σκοπεύει να εισέλθει στο χώρο επιτήρησης θα πρέπει να είναι εφοδιασμένος με μια συσκευή η οποία θα έχει διεύθυνση ip η οποία θα αναγνωρίζεται όντας καταχωρημένη στο σύστημα αυτό έπειτα από έλεγχο. Εάν η διεύθυνση ip της νεοεισελθείσας στο χώρο συσκευής (και κατ' επέκταση ατόμου) συμφωνεί με αυτή με την οποία το σύστημα είναι ρυθμισμένο να αναγνωρίζει σημαίνει ότι είναι εξουσιοδοτημένο να υπάρχει στο χώρο και η διαδικασία της ανίχνευσης παύει προσωρινά. Όταν απομακρυνθεί η συσκευή, το σύστημα αντιλαμβάνεται την απουσία της και επανεκκινεί την διαδικασία της επιτήρησης κανονικά.

Οι παραπάνω σκέψεις υλοποιούνται με τη βοήθεια του script `ipcheck.sh`:

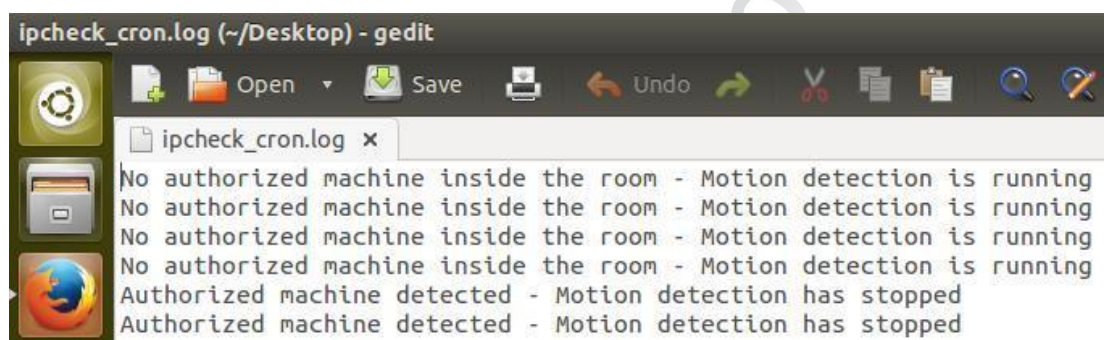
```
#!/bin/bash
# IP address of the entry machine
ENTRYIP="192.168.1.80"
# Ping entry machine's ip
IPCHECK="$(ping $ENTRYIP -c 1)"
# If the machine's ip is not found, resume detection on all cameras
if [ "$?" -eq 0 ]; then
echo Authorized machine detected - Motion detection has stopped
wget -q --delete-after http://localhost:8080/0/detection/pause 2>&1
>/dev/null
# Else if the machine's ip is found, stop detection on all cameras
else
echo No authorized machine inside the room - Motion detection is
running
wget -q --delete-after http://localhost:8080/0/detection/start 2>&1
>/dev/null
fi
```

Αντίστοιχα, θα πρέπει να γίνουν οι κατάλληλες ρυθμίσεις στο `crontab` υποδεικνύοντας την τοποθεσία του script `ipcheck.sh` και θέτοντας το χρόνο

επανάληψης της διαδικασίας ταυτοποίησης της ip (εκτέλεση του script). Κατά τη διάρκεια εκπόνησης του συγκεκριμένου project, η εκτέλεση του script χρησιμοποιήθηκε κάθε ένα λεπτό.

```
# m h dom mon dow   command
*/1 * * * * /home/simos/Desktop/ipcheck.sh 2>&1
>>/home/simos/Desktop/ipcheck_cron.log
```

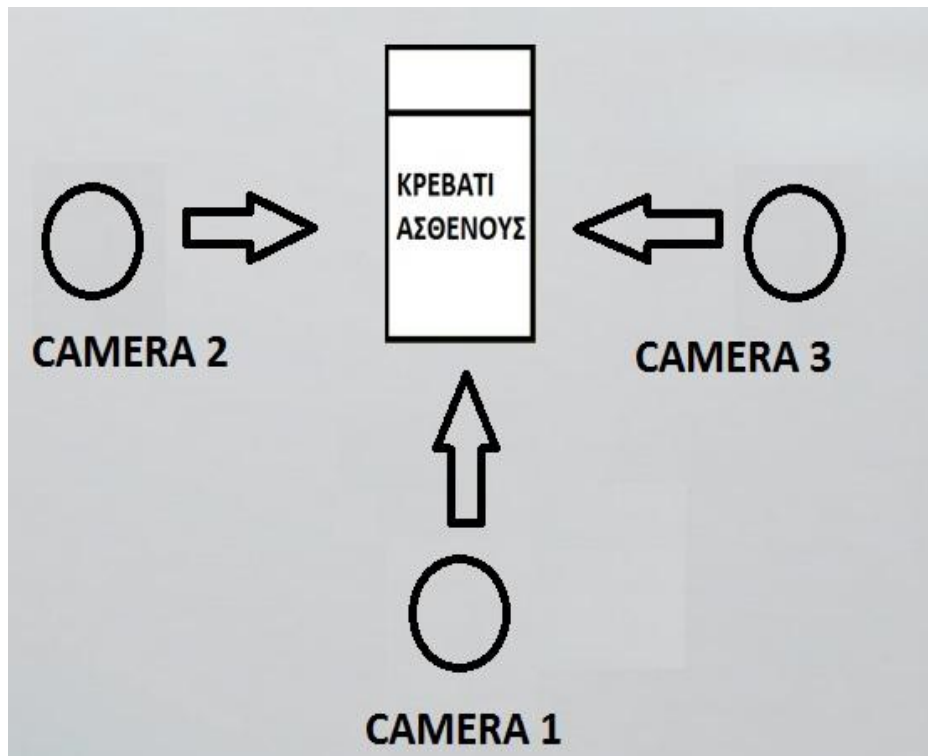
Στην παρακάτω εικόνα (Εικόνα 5.1.2-1) είναι ορατά τα αποτελέσματα έπειτα από την εκτέλεση κάθε λεπτό μέσω του cron του script `ipcheck.sh`. Σε περίπτωση που υπάρχει ταύτιση της νεοεισελθείσας συσκευής με αυτή που έχει δηλωθεί στο σύστημα έχουμε το μήνυμα "ip detected Disarming detection", ενώ στην αντίθετη περίπτωση όπου η συσκευή έχει απομακρυνθεί λαμβάνεται το μήνυμα "ip not detected Arming detection".



Εικόνα (5.1.2-1): Μηνύματα αρχείων log του cron έπειτα από εκτελέσεις για το script ipcheck.sh

5.2 : Συνδυασμός και συνεκτίμηση αποτελέσματος δεδομένων από πολλαπλές κάμερες

Μια ακόμη διεύρυνση των ποιοτικών χαρακτηριστικών του Motion αποτελεί ο συνυπολογισμός του αποτελέσματος σε περίπτωση που χρησιμοποιούνται πολλαπλές κάμερες προς τον ίδιο στόχο. Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκαν τρεις διαφορετικές κάμερες στοχεύοντας προς τον ίδιο τελικό προορισμό (τον ασθενή) από διαφορετική, όμως, γωνία λήψης η καθεμία. Για να ελαχιστοποιηθεί η πιθανότητα ψευδών συναγερμών, υπήρξε η σκέψη να μην θεωρηθεί ως κίνηση η ανίχνευση μόνο από μια κάμερα (αφού αυτό μπορεί να συμβεί σε πολλές περιπτώσεις π.χ. σε περίπτωση αλλαγής επιπέδου φωτισμού στο δωμάτιο βιντεοεπιτήρησης) αλλά να συνεκτιμηθεί το αποτέλεσμα και των τριών καμερών και μόνο εάν όλες τους (οι οποίες στοχεύουν σε ένα κοινό στόχο) ταυτόχρονα ή εντός ενός αποδεκτού χρονικού διαστήματος σκανδαλιστούν, να υπάρχει η αναγγελία της ανίχνευσης κίνησης. Με αυτόν τον τρόπο, εξασφαλίζεται η γνησιότητα της ύπαρξης κίνησης αφού εφόσον η κίνηση έχει αναγνωριστεί από όλες τις κάμερες είναι πλέον σχεδόν αδύνατον αυτή να οφείλεται σε εξωτερικούς παράγοντες, αλλά σε πραγματική ύπαρξη κίνησης στο χώρο βιντεοεπιτήρησης.



Εικόνα (5.2-1): Διάταξη των καμερών στο χώρο για το συνδυασμό και συνεκτίμηση δεδομένων από αυτές

Συνεχίζοντας στην συγγραφή των προγραμμάτων, θα κατευθυνθούμε αρχικά στα αρχεία `thread.conf` όπως περιγράφηκε στην παράγραφο 3.1-1. Εκεί, ενεργοποιείται η επιλογή `on_event_start` όχι όμως στο αρχείο `configuration` όπως έγινε στην παράγραφο 3.4, αλλά στα ίδια τα αρχεία `thread.conf`. Με αυτόν τον τρόπο, κάθε φορά που ανιχνεύεται κίνηση από κάθε μία από τις τρεις κάμερες (`thread1.conf`, `thread2.conf`, `thread3.conf`) θα εξάγεται η εντολή ώστε να ενεργοποιείται ένα πρόγραμμα της αρεσκείας μας. Στην εν λόγω περίπτωση, κατασκευάστηκε σε C το αρχείο `inform.c` το οποίο έχει τη δυνατότητα δημιουργίας ενός αρχείου `text` και απόδοση χρονοσφραγίδας (`timestamp`) σε αυτό. Συγκεκριμένα, τοποθετώντας την εντολή ώστε να εξαχθεί το αρχείο `inform.c` στις επιλογές `on_event_start` των αρχείων `thread1.conf`, `thread2.conf`, `thread3.conf` αντίστοιχα, κάθε φορά που θα ανιχνεύεται κίνηση θα έχουμε την καταγραφή σε αρχείο `text` της ώρας που συνέβη η κίνηση καθώς και από ποια κάμερα, δηλαδή από ποιο αρχείο `thread.conf` προήλθε. Για να ξεχωρίζει το σύστημα από ποια κάμερα προήλθε η δημιουργία του αρχείου, αρκεί να δηλωθεί ένα ξεχωριστό όνομα αρχείου δίπλα από την εντολή εκτέλεσης του `inform.c` σε κάθε αρχείο `thread.conf`. Επομένως, για το αρχείο `thread1.conf` θα έχουμε την εξαγωγή του αρχείου `text` `output1`, για το αρχείο `thread2.conf` θα έχουμε την εξαγωγή του αρχείου `text` `output2` κ.ο.κ.

Παρακάτω, παρατίθεται η μεταβολή-πρόσθεση της επιπλέον εντολής στα αρχεία `thread1.conf`, `thread2.conf`, `thread3.conf` σε σχέση με τη μορφή που είχαν στην παράγραφο 3.1-1:

Έτσι, η προσθήκη στο αρχείο `thread1.conf` για την πρώτη κάμερα είναι η εξής:

```
# Command to be executed when an event starts. (default: none)

# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start /home/simos/Desktop/inform output1
```

Επίσης, στο αρχείο `thread2.conf` για την δεύτερη κάμερα είναι η εξής:

```
# Command to be executed when an event starts. (default: none)

# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start /home/simos/Desktop/inform output2
```

Και τέλος, στο αρχείο `thread3.conf` για την τρίτη κάμερα είναι η εξής:

```
# Command to be executed when an event starts. (default: none)

# An event starts at first motion detected after a period of no
motion defined by gap
on_event_start /home/simos/Desktop/inform output3
```

Ακόμη, ακολουθεί το αρχείο `inform.c`:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main (int argc, char *argv[]) {

    time_t now;
    struct tm *tm;

    now = time(0);
    if ((tm = localtime (&now)) == NULL) {
        printf ("Error extracting time stuff\n");
        return 1;
    }

    printf ("%04d-%02d-%02d %02d:%02d:%02d\n",
            tm->tm_year+1900, tm->tm_mon+1, tm->tm_mday,
            tm->tm_hour, tm->tm_min, tm->tm_sec);

    FILE *f = fopen(argv[1], "w");
    if (f == NULL)
    {
        printf("Error opening file!\n");
        exit(1);
    }

    if (argc>2)

    {
        fprintf(f, "%04d-%02d-%02d %02d:%02d:%02d\n", 0, 0, 0, 0, 0, 0);
    }
}
```

```

else {

fprintf (f,"%04d-%02d-%02d %02d:%02d:%02d\n",
        tm->tm_year+1900, tm->tm_mon+1, tm->tm_mday,
        tm->tm_hour, tm->tm_min, tm->tm_sec);
}
fclose(f);

return 0;
}

```

Στη συνέχεια, έχοντας αποκτήσει την πληροφορία για την ακριβή ώρα που ανίχνευσε την κίνηση μια συγκεκριμένη κάμερα χρειάζεται ένα πρόγραμμα το οποίο θα ελέγχει τα αποτελέσματα και όταν εντοπίζεται κίνηση από όλες τις κάμερες ταυτόχρονα μέσα σε ένα προκαθορισμένο χρονικό όριο (time threshold), θα αναγγέλλει την ανίχνευση κίνησης ως alarm. Το εν λόγω αρχείο, δημιουργήθηκε και αυτό σε C, ονομάστηκε `check.c` και ακολουθείται από ένα δικό του αρχείο `configuration` με την ονομασία `check.conf` στο οποίο θα πρέπει να δηλώσει αρχικά ο χρήστης το συνολικό αριθμό των καμερών που χρησιμοποιούνται (`totalcameras`), τα ονόματα των αρχείων `text` που εξάγονται από την κάθε κάμερα σε κάθε μεμονωμένη ανίχνευση από την καθεμία με τη χρήση του `inform.c` (`camera_output`), καθώς και το επιθυμητό χρονικό όριο (`threshold`, σε δευτερόλεπτα) μέσα στο οποίο θα είναι ανεκτή η χρονική διαφορά μεταξύ των ανιχνεύσεων ώστε να θεωρηθεί σωστή ανίχνευση κίνησης και να δημιουργηθεί το `alarm`.

Έτσι, λοιπόν, το αρχείο `check.conf` στην περίπτωση χρησιμοποίησης τριών καμερών και χρονικού ορίου δέκα δευτερολέπτων έχει ως εξής:

```

##### f
#configuration file for check
##### f

totalcameras 3

threshold 10

camera_output output1
camera_output output2
camera_output output3

```

Επιπρόσθετα, παρατίθεται το αρχείο `check.c`:

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>

int main (int argc, char *argv[]) {
int day, hour, min, sec, yy;

FILE *f[11]; //we can accept up to 10 cameras

char str1[30];
char str2[30];
char filenames[11][30];
int value;

```

```

int threshold;//keeping the value of threshold
int totalcameras;//keeping the total number of cameras

long css[30];
int i=0;
int tempre;
int tempnow;

//Reads the configuration file
FILE *readconf = fopen("check.conf", "r");
if (readconf == NULL)
{
    printf("Error opening file!\n");
    return 0;
}

while (!feof(readconf))
{

fscanf(readconf,"%s %s",str1,str2);

if (!(strcmp(str1,"totalcameras"))) {
value=atoi(str2); totalcameras=value;
}

if (!(strcmp(str1,"threshold"))) { value=atoi(str2);threshold=value;
;}

if (!(strcmp(str1,"camera_output")))
{strcpy(filenamees[counter++],str2);
}
}

for (i=1;i<=totalcameras;i++)
{

f[i] = fopen(filenamees[i], "r");
if (f == NULL)
{

printf("Error opening file!\n");
return 0;
}

}

while(0<1)
{
for (i=1;i<=totalcameras;i++)
{
fscanf(f[i],"%04d-%02d-%02d %02d:%02d:%02d\n",
&yy,&yy,&day,&hour,&min,&sec);

//the following array holds the timestamp in seconds
//

css[i]=day*24*60*60+hour*60*60+min*60+sec;

}

// THE FOLLOWING CODE

```



```

// READS THE PREVIOUS ARRAY AND CHECKS
// IF WE HAVE A MATCH WITHIN THE THRESHOLD

tempre=css[1];

int hit=1;

for (i=2;i<=totalcameras;i++)
{
tempnow=css[i];

if (abs(tempre-tempnow)<=threshold) {
hit++;

if ((hit==totalcameras) &&(tempre!=0)) {printf("MOTION ON ALL
CAMERAS \n");};

//EXTERNAL COMMAND
system("/home/simos/Desktop/mail_motion");

int ct;
for (ct=1;ct<=totalcameras;ct++)
{

char informstring[100];
strcpy(informstring, "./inform ");
strcat(informstring, filenames[ct]);
strcat(informstring, " -null");
system(informstring);
}
}
else{
}

}
} //for end

for (i=1;i<=totalcameras;i++)
{
rewind(f[i]);
}
}

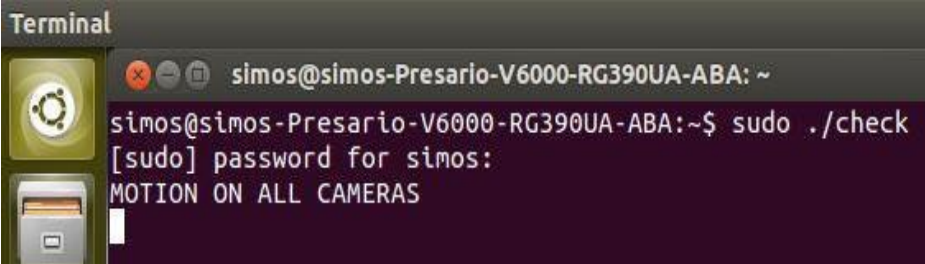
return 0;
}

```

Συνοψίζοντας, το αρχείο `check.c` διαβάζει τα αρχεία `text` που έχουν εξαχθεί έπειτα από ανιχνεύσεις από την κάθε κάμερα ξεχωριστά, τα καταχωρεί και τα συγκρίνει ως προς το χρόνο που έχουν δημιουργηθεί σύμφωνα με το `threshold` που έχει οριστεί στο αρχείο `check.conf`. Αν μέσα στο κατώφλι αυτό δεν υπάρχει κίνηση, συνεχίζει να εκτελείται στο παρασκήνιο και να συγκρίνει συνεχώς τα αρχεία που εξάγονται από τις κάμερες. Αντίθετα, αν έχει αναγγελθεί κίνηση από όλες τις κάμερες το εντοπίζει και σημαίνει συναγερμό δημιουργώντας στη γραμμή εντολών το μήνυμα "MOTION ON ALL CAMERAS". Επίσης, αποδόθηκε στο αρχείο `check.c` η πολύ σημαντική ιδιότητα του μηδενισμού της χρονοσφραγίδας των αρχείων `text` έπειτα από κάθε σύγκριση έτσι ώστε να υπάρχει συνεχής ανανέωση σε περίπτωση νέων ανιχνεύσεων.

Τέλος, χρησιμοποιώντας το script που γράφτηκε στην παράγραφο 3.5, μπορούν να ενταχθούν στο πρόγραμμα και να αξιοποιηθούν οι δυνατότητες αποστολής email και SMS σε κάθε συναγερμό.

Με αυτές τις προσθήκες, όλες οι ειδοποιήσεις πλέον εξάγονται και διαχειρίζονται από το αρχείο `check.c` και την οθόνη αποτελεσμάτων αυτού και όχι από το ίδιο το Motion.



```
Terminal
simos@simos-Presario-V6000-RG390UA-ABA: ~
simos@simos-Presario-V6000-RG390UA-ABA:~$ sudo ./check
[sudo] password for simos:
MOTION ON ALL CAMERAS
```

Εικόνα (5.2-2): Μηνύματα στη γραμμή εντολών έπειτα από την εκτέλεση του προγράμματος `check.c`

5.3 : Εκμάθηση από το σύστημα της συνηθισμένης συμπεριφοράς στις κινήσεις του ασθενούς και δημιουργία συναγερμού μόνο σε διαφορετική περίπτωση

Η τελευταία πολύ χρήσιμη προσθήκη η οποία προσδίδει έντονο ευφυή χαρακτήρα στο όλο project είναι η δημιουργία συναγερμών χρησιμοποιώντας ουσιαστικά την ανάστροφη διαδικασία σχετικά με τα όσα μελετήθηκαν έως τώρα. Συγκεκριμένα, θεωρείται λογικό να υπάρχει κίνηση σε συγκεκριμένους χώρους για ορισμένα χρονικά διαστήματα αφού πρώτα μελετηθεί η φυσιολογική καθημερινή συμπεριφορά του παρακολουθούμενου, απεικονίζοντας έτσι τη φυσιολογική συμπεριφορά του, προσαρμοσμένο στο χώρο βιντεοεπιτήρησης.

Έτσι, δεν περιορίζεται μόνο σε παρακολούθηση κλινηρών ασθενών αλλά γενικότερα ατόμων που απαιτούν παρακολούθηση δίχως να παραβιάζεται η ιδιωτικότητα τους. Το σύστημα αποκτά την ελαστικότητα να προσαρμόζεται σε διαφορετικούς χώρους βιντεοεπιτήρησης καθώς και κατηγορίες παρακολουθούμενων και να γεννά μόνο τους συναγερμούς εκείνους που ενδιαφέρουν τον διαχειριστή.

Με άλλα λόγια, το σύστημα ενημερώνεται μέσω του διαχειριστή και προσαρμόζεται κατάλληλα σύμφωνα με ένα καθημερινό πρόγραμμα από φυσιολογικές κινήσεις οι οποίες αν δεν συνάδουν με αυτό θα ελέγχονται και θα δημιουργείται ο συναγερμός. Ο διαχειριστής επιλέγει τον αριθμό των καμερών που θα χρησιμοποιήσει, τη θέση και τον τύπο τους, τα αποδεκτά χρονικά όρια παραμονής του παρακολουθούμενου στο πεδίο παρατήρησής της καθεμίας, τον χρόνο αδράνειας πριν τη λήψη απόφασης, τον συνδυασμό του αποτελέσματος δύο καμερών για την εξαγωγή αποφάσεων καθώς και τα μηνύματα που εξάγονται σε περίπτωση φυσιολογικής ή μη συμπεριφοράς. Με αυτή τη λογική, το σύστημα αναπτύσσεται και γίνεται προσαρμόσιμο και ελαστικό σε διαφορετικά σενάρια παρακολούθησης τα οποία επιλέγονται αποκλειστικά από τον διαχειριστή του συστήματος.

Στις επόμενες παραγράφους, εκτός από τη συγγραφή των προγραμμάτων για την υλοποίηση του παραπάνω εγχειρήματος, παρουσιάζονται και δύο πιθανά σενάρια βιντεοεπιτήρησης (το πρώτο σε συνθήκες δωματίου νοσοκομείου και το

δεύτερο σε συνθήκες κατ' οίκον παρακολούθησης) με πολλαπλές κάμερες σε συνδυασμό με όλα τα χαρακτηριστικά που αναπτύχθηκαν σε όλες τις προηγούμενες παραγράφους των κεφαλαίων 3 και 4.

Αναλυτικότερα, κατασκευάζεται ένα πρόγραμμα php σε δύο σκέλη: το αρχείο διαχείρισης (intelligentfeaturesconf.php) και το κύριο αρχείο (intelligentfeatures.php), τα οποία καλούνται απευθείας από την αρχική σελίδα του εξελιγμένου web interface (αποτελώντας την κατηγορία "Intelligent Features") όπως περιγράφηκε στην παράγραφο 4.3 και δρουν σε συνεργασία με τις καταχωρήσεις στη βάση δεδομένων που αναλύθηκαν στην παράγραφο 3.3 (camera, time_stamp, έτσι ώστε να λαμβάνουν τις πληροφορίες για την κάθε κάμερα που ανίχνευσε την κίνηση με την ανάλογη χρονοσφραγίδα, να ενημερώνονται από τον χρήστη για τα φυσιολογικά ή μη γεγονότα και να εξάγουν αποφάσεις σε μορφή μηνύματος.

Στο web interface αρχικά, ενεργοποιείται η δυνατότητα της επιλογής "Enable Intelligent Features" και έτσι ο χρήστης μπορεί να καλεί το αρχείο διαχείρισης intelligentfeaturesconf.php από την επιλογή "Configure" καθώς και μια παραλλαγή (που θα αναλυθεί αργότερα) του αρχείου intelligentfeatures.php (run.php) από την επιλογή "Show Results" για την απεικόνιση των αποτελεσμάτων.

Παρακάτω, παρατίθεται η μεταβολή ώστε να υπάρχει η δυνατότητα της επιλογής "Enable Intelligent Features" (με επιλογές "Configure" και "Show Results") στο αρχείο webinterface.html:

```
<html>
  <head>
    <title>Motion Detection</title>
    <script type='text/javascript'>
      function popupwindow(url, title, w, h) {
        var left = (screen.width / 2) - (w / 2);
        var top = (screen.height / 2) - (h / 2) - 50;
        return window.open(url, title, 'toolbar=no,
location=no, directories=no, status=no, menubar=no, scrollbars=no,
resizable=no, copyhistory=no, width=' + w + ', height=' + h + ',
top=' + top + ', left=' + left);
      }]]&gt;
    &lt;/script&gt;
    &lt;style&gt;
      .cssmenu {
        background: #333;
        list-style: none;
        margin: 0;
        padding: 0;
        width: 12em;
      }
      .cssmenu li {
        font: 90% 'Lucida Sans Unicode', 'Bitstream Vera
Sans', 'Trebuchet Unicode MS', 'Lucida Grande', Verdana, Helvetica,
sans-serif;
        margin: 0;
        padding: 0;
        list-style: none;
      }
      .cssmenu a {
        background: #333;
        border-bottom: 1px solid #393939;</pre></div><div data-bbox="479 923 516 940" data-label="Page-Footer"><p>[99]</p></div>
```

```

        color: #ff0000;
        display: block;
        margin: 0;
        padding: 8px 12px;
        text-decoration: none;
        font-weight: normal;
    }
    .cssmenu a:hover {
        background: #2580a2 url('hover.gif') left center no-
repeat;
        color: #fff;
        padding-bottom: 8px;
    }
    .cssmenu li.header{
        color: #FFFFFF;
        font-size: 1.4em;
    }
</style>
</head>

<body background="/Computer_eye2.jpg" no-repeat >

    <div id="resize"></div></br>

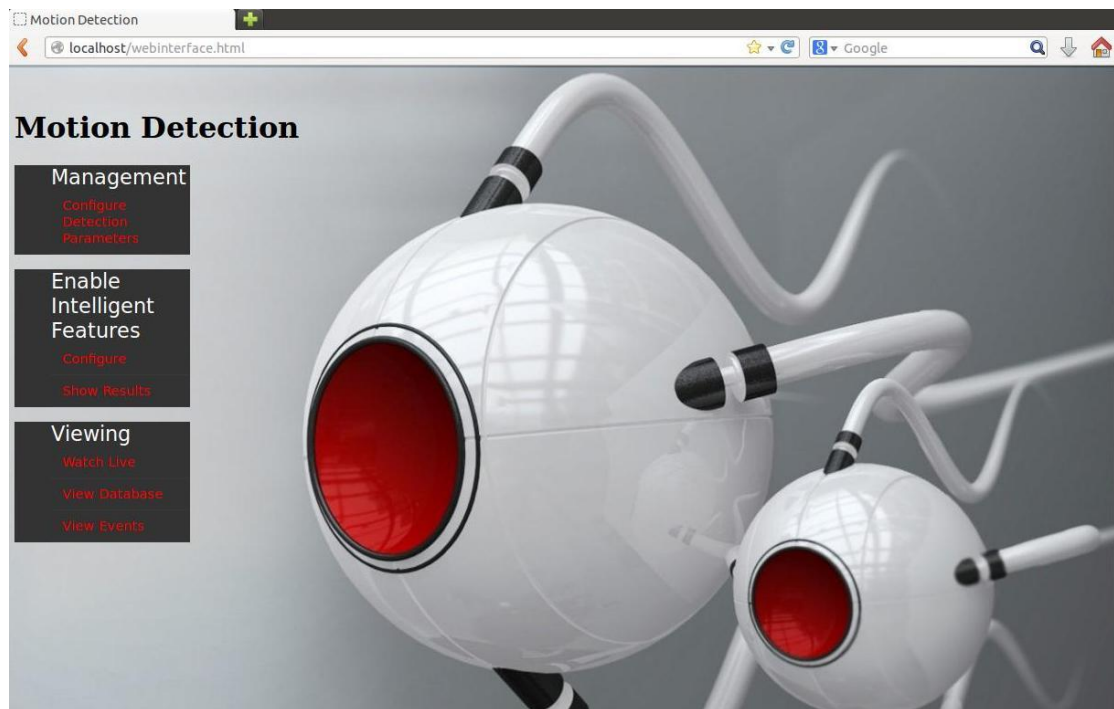
    <h1>Motion Detection</h1>

    <div class='cssmenu'>
        <ul>
            <li class="header"><span>Management</span></li>
            <li><a href="http://localhost:8080/">Configure
Detection Parameters</a></li>
        </ul>
    </div>

    <div class='cssmenu'>
        <ul>
            <li class="header"><span>Enable Intelligent
Features</span></li>
            <li><a href="intelligentfeaturesconf".php
">Configure</a></li>
            <li><a href="run.php">Show Results</a></li>
        </ul>
    </div>

    <div class='cssmenu'>
        <ul>
            <li class="header"><span>Viewing</span></li>
            <li><a href="multiplecameras.html">Watch
Live</a></li>
            <li><a href="database.php">View Database</a></li>
            <li><a
href="javascript:popupwindow('imageviewer/imageviewer.php','Image
Viewer',1024,768)">View Events</a></li>
        </ul>
    </div>
</body>
</html>

```



Εικόνα (5.3-1): Προσθήκη της επιλογής Intelligent Features στην κατηγορία Management στην αρχική καρτέλα του εξελεγμένου web interface

Συνεχίζοντας με το αρχείο διαχείρισης `intelligentfeaturesconf.php` το οποίο όπως προτυπώθηκε αναδύεται μέσω του web interface, πρέπει να αναφερθεί ότι είναι το αρχείο εκείνο που δίνει τη δυνατότητα στον διαχειριστή να επιλέγει τον χώρο που τοποθετείται η κάθε κάμερα (Location καθώς και τον τύπο της (Type)), ορίζοντας τα αποδεκτά χρονικά όρια παραμονής του παρακολουθούμενου στο πεδίο παρατήρησής της (Normal Time Threshold), τον χρόνο αδράνειας πριν τη λήψη απόφασης (Idle Time), τη συνεργασία μεταξύ δύο καμερών για τη λήψη αποφάσεων (Cooperation With Camera) καθώς και τα μηνύματα που εξάγονται σε περίπτωση φυσιολογικής (Normal Situation Text) ή μη συμπεριφοράς (Abnormal Situation Text). Θα πρέπει πρωταρχικά να δημιουργηθεί ένας πίνακας στη βάση δεδομένων `simos` στην `MySQL` στον οποίο θα καταχωρούνται μέσω του `intelligentfeaturesconf.php` οι επιλογές που αναφέρθηκαν παραπάνω. Ο πίνακας ονομάζεται `intelligentfeaturestable` και περιλαμβάνει τις εξής μεταβλητές προς καταχώρηση: `id` κάμερας, `location` (χώρος τοποθέτησης της εκάστοτε κάμερας), `type` (τύπος κάμερας, π.χ. `primary` ή `secondary`), `normal_time` (μέγιστο αποδεκτό χρονικό όριο παραμονής στο πεδίο παρατήρησης), `idle_time` (χρόνος αδράνειας πριν τη λήψη αποφάσεων), `cooperate` (συνεκτίμηση του αποτελέσματος και από δεύτερη κάμερα πριν τη λήψη απόφασης), `normal_text` (περιεχόμενο μηνύματος φυσιολογικής κατάστασης), `abnormal_text` (περιεχόμενο μηνύματος μη φυσιολογικής κατάστασης).

```
CREATE TABLE IF NOT EXISTS `intelligentfeaturestable` (
  `location` int(11) NOT NULL,
  `type` int(11) NOT NULL,
  `normal_time` int(11) NOT NULL,
  `idle_time` int(11) NOT NULL,
  `cooperate` int(11) NOT NULL,
  `normal_text` varchar(100) NOT NULL,
  `abnormal_text` varchar(100) NOT NULL,
```

```

`enabled` tinyint(4) NOT NULL,
`id` int(11) NOT NULL,
PRIMARY KEY (`id`)
)

```

Έτσι, λοιπόν, παρατίθεται ο κώδικας για το αρχείο διαχείρισης intelligentfeaturesconf.php, το οποίο αφού δεχθεί από τον χρήστη μέσω της γραφικής του διεπαφής τις παραμέτρους που αναφέρθηκαν παραπάνω στην κατασκευή του πίνακα intelligentfeaturestable, αναμένει για την ενεργοποίηση του πλήκτρου "Submit" το οποίο εν τέλει θα καταχωρήσει τις παραμέτρους στον πίνακα. Ακόμη, χρησιμοποιώντας παράλληλα και τις ιδιότητες για jquery του javascript, απεικονίζει σε λίστα html τις επιλογές Camera, Location, Type, Normal Time Threshold, Idle Time, Cooperate with Camera, Normal Situation Text, Abnormal Situation Text:

```

<?php
$con = mysqli_connect("localhost", "root", "simos", "simos");
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
// Check connection
}

if($_POST["Submit"]){ //Record values to intelligentfeaturestable
    if (!mysqli_query($con, "delete FROM
intelligentfeaturestable;")){
        echo "Error: " . mysqli_error($con);
    }
    for ($i = 2; $i <= 21; $i++) { //We can have up to 20 cameras
        if (!mysqli_query($con, "Insert into intelligentfeaturestable
values('".$_POST["enabled$i"]."', '".$_.$i."', '".$_POST["normal_time$i"].
"', '".$_POST["idle_time$i"]."', '".$_POST["cooperate$i"]."'".$_POST["
normal_text$i"]."', '".$_POST["abnormal_text$i"]."'")) {
            echo "Error: " . mysqli_error($con);
        }
    }
}

$select = mysqli_query($con, "SELECT * FROM
intelligentfeaturestable;");
if (!$select){
    echo "Error: " . mysqli_error($con);
}

?>
<!DOCTYPE html>
<html>
<head>
<script src="js/script.js"></script>
<script src="js/jquery.min.js"></script>
</head>
<body>
<body background="/Computer_eye2.jpg" no-repeat >

    <div id="resize"></div></br>
    <form method="post">
        <table>
            <tbody>
                <tr><th></th><th>Camera #</th><th
style='width:80px'>Location</th><th style='width:80px'>Type</th><th

```

```

style='width:80px'>Normal Time Threshold</th><th
style='width:80px'>Idle Time</th><th style='width:80px'>Cooperation
With Camera</th><th>Normal Situation Text</th><th>Abnormal Situation
Text</th></tr>

<?php
for ($i = 2; $i <= 21; $i++) {
    $row =
mysqli_fetch_array($intelligentfeaturestable);
    if ($row && $row["enabled"]==1){
        echo "<tr><td><input checked
class='selection' name='enabled$i' type='checkbox'
value='1' /></td><td align='right'>$i</td><td><input
style='width:80px' name='location$i'
value='\".$row[\"location\"].\"' /></td><td><input style='width:80px'
name='type$i' value='\".$row[\"type\"].\"' /></td><td><input
style='width:80px' name='normal_time$i'
value='\".$row[\"normal_time\"].\"' /></td><td><input style='width:80px'
name='idle_time$i' value='\".$row[\"idle_time\"].\"' /></td><td><input
style='width:80px' name='cooperate$i'
value='\".$row[\"cooperate\"].\"' /></td><td><input name='normal_text$i'
value='\".$row[\"normal_text\"].\"' /></td><td><input
name='abnormal_text$i'
value='\".$row[\"abnormal_text\"].\"' /></td></tr>";
    }
    else{
        echo "<tr><td><input
class='selection' name='enabled$i' type='checkbox'
value='1' /></td><td align='right'>$i</td><td><input disabled
style='width:80px' name='location$i' /></td><td><input disabled
style='width:80px' name='type$i' /></td><td><input disabled
style='width:80px' name='normal_time$i' /></td><td><input disabled
style='width:80px' name='idle_time$i' /></td><td><input disabled
style='width:80px' name='cooperate$i' /></td><td><input disabled
name='normal_text$i' /></td><td><input disabled
name='abnormal_text$i' /></td></tr>";
    }
}
?>

</table>
<br/><input name="Submit" type="Submit" value="Submit" />
</form>
</body>
</html>

```

Η κατασκευή του αρχείου script.js, για το jquery με τη βοήθεια του javascript ώστε σε περίπτωση που δεν έχει επιλεγθεί κάποιος αριθμός κάμερας να μην μπορούν να επιλεγθούν και οι υπόλοιπες επιλογές για την κάμερα αυτή, φαίνεται παρακάτω. Απαραίτητη είναι και η αποθήκευση της βιβλιοθήκης jquery.min.js στον ίδιο φάκελο με το script.js του Apache server.

```

$(document).ready(function() {
    $(".selection").click(function() {
        if (!$ (this).is(':checked'))

$(this).parent().parent().find("td").children("input").attr("disabled
", "disabled");
        else

```

```

$(this).parent().parent().find("td").children("input").removeAttr("disabled");
    $(this).removeAttr("disabled");
  })
});

```

Έτσι, λοιπόν, προκύπτει η παρακάτω πλατφόρμα που επιτρέπει στον διαχειριστή του συστήματος να επιλέγει τον αριθμό των καμερών που θα χρησιμοποιηθούν, την τοποθεσία και τον τύπο της εκάστοτε κάμερας, να αυξομειώνει τα χρονικά όρια μεταξύ ανίχνευσης, μη ανίχνευσης και του χρόνου λήψεως αποφάσεων, να επιλέγει μια δεύτερη κάμερα για συνεκτίμηση του αποτελέσματος πριν εξάγει απόφαση καθώς και να ορίζει τα μηνύματα που θα εξάγονται για την κάθε κάμερα στις περιπτώσεις φυσιολογικής κατάστασης και κατάστασης συναγερμού.

Camera #	Location	Type	Normal Time Threshold	Idle Time	Cooperation With Camera	Normal Situation Text	Abnormal Situation Text
<input type="checkbox"/>	2						
<input type="checkbox"/>	3						
<input type="checkbox"/>	4						
<input type="checkbox"/>	5						
<input type="checkbox"/>	6						
<input type="checkbox"/>	7						
<input type="checkbox"/>	8						
<input type="checkbox"/>	9						
<input type="checkbox"/>	10						
<input type="checkbox"/>	11						
<input type="checkbox"/>	12						
<input type="checkbox"/>	13						
<input type="checkbox"/>	14						
<input type="checkbox"/>	15						
<input type="checkbox"/>	16						
<input type="checkbox"/>	17						
<input type="checkbox"/>	18						
<input type="checkbox"/>	19						
<input type="checkbox"/>	20						
<input type="checkbox"/>	21						

Εικόνα (5.3-2): Επιλογές του αρχείου διαχείρισης

Στη συνέχεια, το αρχείο διαχείρισης `intelligentfeaturesconf.php` θα παραδώσει την σκυτάλη στο κύριο αρχείο `intelligentfeatures.php` το οποίο έχοντας δεχθεί όλες τις παραμέτρους από το διαχειριστή είναι αυτό που αναλαμβάνει να υλοποιήσει όλες τις εικασίες που τυπώθηκαν αρχικά. Συγκεκριμένα, το αρχείο `intelligentfeatures.php` συνδέεται με τη βάση δεδομένων και με τον πίνακα `simosss` όπως αυτός δημιουργήθηκε στην παράγραφο 3.3 και ο οποίος εμπεριέχει τις μεταβλητές `camera`, `time_stamp` στις οποίες καταχωρείται αντίστοιχα ο αριθμός της κάμερας και η ώρα που συνέβη η κάθε ανίχνευση. Έτσι, λοιπόν, ελέγχει εάν υπάρχουν καταχωρήσεις από την κάμερα 1 (κύρια κάμερα στην οποία είναι φυσιολογικό να υπάρχουν ανιχνεύσεις τον περισσότερο χρόνο της βιντεοεπιτήρησης)

και αν δεν υπάρχουν ελέγχει τις καταχωρήσεις για τα τελευταία λεπτά (εδώ χρησιμοποιούνται 300 λεπτά=5 ώρες διότι στα σενάρια που εξετάζονται παρακάτω υπάρχουν μεγάλοι χρόνοι αναμονής και πολλές κάμερες) ταξινομημένα με το χρόνο. Έχοντας αντλήσει τη χρονοσφραγίδα από την κάμερα που έχει κάνει την τελευταία καταχώρηση ελέγχει τις προηγούμενες καταχωρήσεις για την εν λόγω κάμερα και εάν υπάρχουν, καταχωρεί τις χρονοσφραγίδες της πρώτης και της τελευταίας καταχώρησης. Τελικά, ελέγχει τον πίνακα intelligentfeaturestable ώστε να ενημερωθεί για το configuration που έχει ορίσει ο διαχειριστής και αν ικανοποιούνται οι χρόνοι Normal Time Threshold και Idle Time για την τρέχουσα κάμερα και οι χρόνοι που έχουν οριστεί για την συνεκτίμηση με την δεύτερη κάμερα (εάν υπάρχει), αποφασίζει ότι πρόκειται για φυσιολογική κατάσταση. Σε αντίθετη περίπτωση αποφασίζει ότι πρόκειται για μη φυσιολογική κατάσταση και συγχρόνως κάνει κλήση του script που γράφτηκε στην παράγραφο 3.5, ώστε να ενεργοποιηθούν οι δυνατότητες αποστολής email και SMS σε κάθε συναγερμό. Επίσης, σε περίπτωση που δεν υπάρχουν καθόλου καταχωρήσεις από καμία κάμερα τα τελευταία λεπτά και πάλι σημάνει συναγερμό καθώς αποτελεί μη φυσιολογικό γεγονός.

Ο κώδικας για την κατασκευή του αρχείου intelligentfeatures.php είναι:

```
<?php
date_default_timezone_set('Europe/Athens');
//Set timezone
$con = mysqli_connect("localhost", "root", "simos", "simos");
//Check connection
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " .
    mysqli_connect_error()."\n";
}
//If there is no record for camera 1
$getvalues = mysqli_query($con, "SELECT * FROM simosss where
time_stamp = CURRENT_TIMESTAMP and camera = 1");
if (!$getvalues) {
    echo "Error: " . mysqli_error($con)."\n";
}
//Find records for the last 300 minutes, sorted by time
if (!mysqli_fetch_array($getvalues)) {
    $last300minvalues = mysqli_query($con, "SELECT * FROM simosss
where time_stamp >= (CURRENT_TIMESTAMP - INTERVAL 300 MINUTE) order
by time_stamp desc;");
    $row = mysqli_fetch_array($last300minvalues);
    if ($row) {
        //Take the last camera's record
        $current = $row['camera'];
        //If the last record is not from camera 1 means that this is
the current camera
        if ($current != 1) {
            //Store the timestamp of the last record
            $last_time = $first_time = $row['time_stamp'];
            //Search the timestamp of the first record for the
current camera
            while ($row = mysqli_fetch_array($last300minvalues)) {
                $first_time = $row['time_stamp'];
                //If there is record for another camera, the previous
one is the first one record for the current camera
                if ($row['camera'] != $current) {
                    break;
                }
            }
        }
    }
}
```

```

        } }
        //Get the configuration for the last current camera
        $intelligentfeaturestableresult = mysqli_query($con,
"SELECT * FROM intelligentfeaturestable where id=" . $current);
        $intelligentfeaturestable=
mysqli_fetch_array($intelligentfeaturestableresult);
        if (!$intelligentfeaturestable||
$intelligentfeaturestable['enabled'] == 0) { // Check if there is
configuration available
            echo date("Y/m/d H:i:s").": ". "No configuration
for camera ".$current."\n";
        }else{
            $normal_time =
$intelligentfeaturestable['normal_time']; //Normal Time Threshold
            $idle_time = $intelligentfeaturestable['idle_time'];
//Idle Time
            $last_time = strtotime($last_time);
//Convert time php
            $first_time = strtotime($first_time); //Convert time
php
            //If has no passed more time than Normal Time
Threshold and Idle Time print normal situation
            if ($last_time - $first_time <= $normal_time +
$idle_time) {
                echo date("d/m/Y H:i:s").":
".$intelligentfeaturestable['normal_text']."\n"; //Print normal
situation
            }

            //Otherwise check if there is record
from the cooperation camera
            else {
                $current2 = $row['camera'];
                if ($current2 == $intelligentfeaturestable['cooperate']) {
//If it is the cooperation camera make it the new current
camera and store the timestamp of the last record
                $last_time2 = $first_time2 = $row['time_stamp'];
                //Search the timestamp of the first record for the
current camera
                while ($row = mysqli_fetch_array($last300minvalues)) {
                    $first_time2 = $row['time_stamp'];
                    //If there is record for another camera, the previous
one is the first one record for the current camera
                    if ($row['camera'] != $current2) {
                        break;
                    }
                }
                //Get the configuration for the cooperation camera
                $intelligentfeaturestableresult2 = mysqli_query($con,
"SELECT * FROM intelligentfeaturestable where id=" . $current2);
                $intelligentfeaturestable2=
mysqli_fetch_array($intelligentfeaturestableresult2);
                if (!$intelligentfeaturestable2||
$intelligentfeaturestable2['enabled'] == 0) { // Check if there is
configuration available
                    echo date("Y/m/d H:i:s").": ". "No configuration
for camera ".$current2."\n";
                }else{
                    $idle_time2 =
$intelligentfeaturestable2['idle_time']; //Idle Time of cooperation
camera

```

```

                                $last_time2 = strtotime($last_time2);
//Convert time php
                                $first_time2 = strtotime($first_time2); //Convert
time php
                                //If has passed more time than cooperation camera's
Idle Time print abnormal situation and call alarm's script
                                if ($last_time2 - $first_time2 >= $idle_time2) {
                                        echo date("d/m/Y H:i:s").":
". $intelligentfeaturestable2['abnormal_text']."\n"; //Print abnormal
situation
                                $outcome =
shell_exec('/home/simos/Desktop/mail_motion'); //External command
                                echo $outcome; }
                                //Otherwise print normal situation
                                else {
                                        echo date("d/m/Y H:i:s").":
". $intelligentfeaturestable2['normal_text']."\n"; //Print normal
situation}
                                }
                                }
                                }
                                }
                                } else { //If there is no record find for the last 300 minutes
                                        echo date("d/m/Y H:i:s").": ". "Alarm!!!No motion recorded in
all cameras for the last minutes ". "\n";
                                        $outcome =
shell_exec('/home/simos/Desktop/mail_motion'); //External command
                                        echo $outcome;
                                }
                                }
?>

```

Για να υπάρχει η δυνατότητα να εκτελείται κάθε δευτερόλεπτο το αρχείο `intelligentfeatures.php` καθώς και να εξάγει τα μηνύματα Normal Situation Text και Abnormal Situation Text σε ένα αρχείο `text` για πλήρη καταγραφή και παρακολούθηση των γεγονότων σε οποιοδήποτε χρόνο (το οποίο ονομάζεται `intelligentfeatures.txt`) κατασκευάζεται το script `runbash.sh`:

```

#!/bin/bash

while true
do
    php /var/www/intelligentfeatures.php >> intelligentfeatures.txt
    sleep 1
done

```

Τέλος, για πρακτικούς λόγους ώστε να είναι δυνατή η αξιοποίηση του παραπάνω script μέσω κλήσης της επιλογής "Show Results" του "Enable Intelligent Features" του εξελιγμένου web interface (αφού δεν είναι δυνατόν να γίνει απευθείας κλήση bash script από αρχείο html) κατασκευάζεται σε php το αρχείο `run.php` το οποίο απλά κάνει κλήση του `runbash.sh` και όπως προτυπώθηκε και στις αρχές της παραγράφου 5.3 είναι αυτό που καλείται εν τέλει από το αρχείο `webinterface.html`. Το περιεχόμενο του `run.php` έχει ως εξής:

```

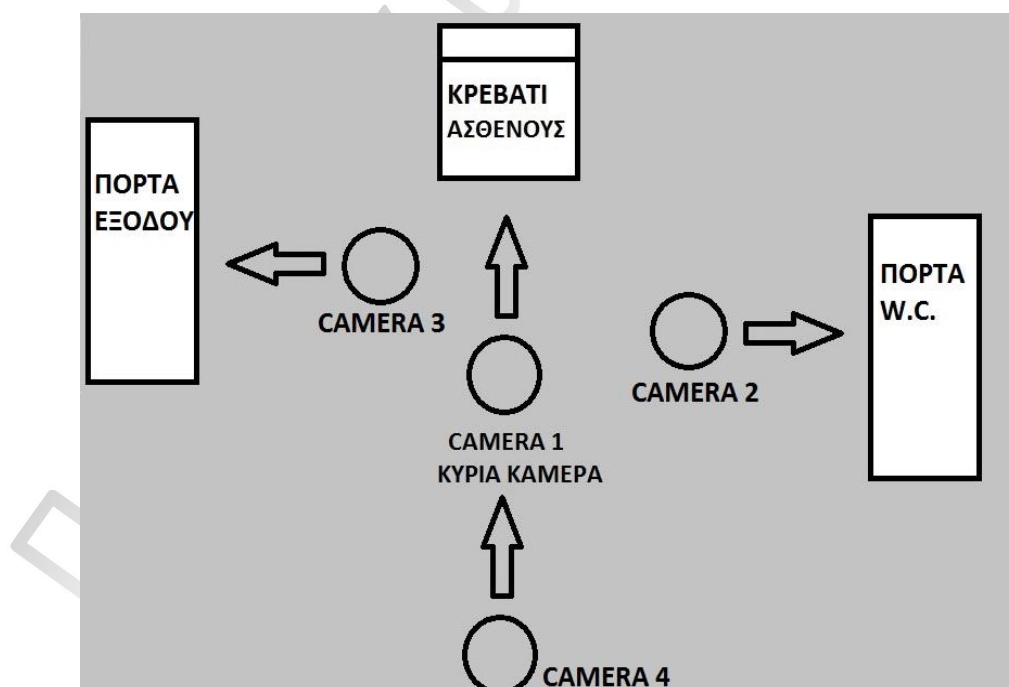
<?php
$outcome = shell_exec('/var/www/runbash.sh');
                                echo $outcome;
?>

```

Για να γίνει περισσότερο κατανοητή η χρήση της επιλογής "Enable Intelligent Features" ακολουθούν δύο πιθανά σενάρια βιντεοεπιτήρησης (το πρώτο σε συνθήκες δωματίου νοσοκομείου και το δεύτερο σε συνθήκες κατ' οίκον παρακολούθησης) με πολλαπλές κάμερες.

5.3.1 : Σενάριο 1 - Παρακολούθηση ασθενούς με πολλαπλές κάμερες σε ένα δωμάτιο νοσοκομείου

Στην πρώτη περίπτωση, επιλέχθηκε ο έλεγχος ενός ασθενή σε ένα δωμάτιο νοσοκομείου που παρακολουθείται από τέσσερις κάμερες. Αφού πρωτίστως παρατηρήθηκε η καθημερινή δραστηριότητά του ασθενούς αποφασίστηκε να καταχωρηθεί στο σύστημα μια σειρά από ρυθμίσεις οι οποίες αποτελούν τη φυσιολογική συμπεριφορά. Έτσι, λοιπόν, τοποθετήθηκε η κάθε κάμερα με πεδίο παρατήρησης σε αντίστοιχα κρίσιμα σημεία κοντά στα περιθώρια κίνησης του παρακολουθούμενου στο δωμάτιο (η θέση και ο τύπος τους ορίζεται με την καταχώρηση των επιλογών Location και Type αντίστοιχα) και ορίστηκαν στο αρχείο διαχείρισης `intelligentfeaturesconf.php` τα χρονικά αποδεκτά περιθώρια για την παραμονή στα σημεία αυτά και την εναλλαγή της κίνησης μεταξύ των προαναφερθέντων σημείων (Normal Time Threshold και Idle Time) καθώς και η εξαγωγή των ανάλογων μηνυμάτων (Normal Situation Text, Abnormal Situation Text). Συγκεκριμένα, η πρώτη κάμερα αποτελεί την κύρια κάμερα και κοιτάζει απευθείας προς το κρεβάτι του ασθενούς. Η δεύτερη κάμερα κοιτάζει την είσοδο της τουαλέτας (οπότε επιλέγεται Location=bathroom και Type=Door), η τρίτη την πόρτα της εξόδου από το χώρο (οπότε επιλέγεται Location=exit και Type=Door) και η τέταρτη και πάλι προς τον ασθενή αλλά υπό κλίση προς τη μεριά του πατώματος ώστε να εντοπίσει τυχόν πτώση (οπότε επιλέγεται Location=bed και Type=Room).



Εικόνα (5.3.1-1): Σενάριο 1- Παρακολούθηση ασθενούς σε ένα δωμάτιο νοσοκομείου με πολλαπλές κάμερες

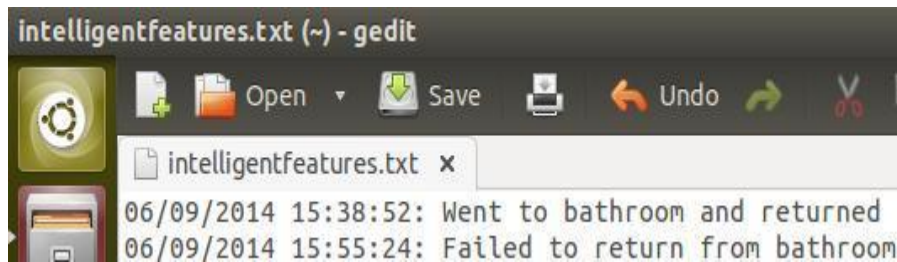
Φυσικά, μόλις γίνει εκκίνηση του Motion η κάμερα 1 θα εντοπίσει την κίνηση αλλά δεν θα δημιουργηθεί alarm και θα γίνει εκμετάλλευση μόνο των στοιχείων της κίνησης. Με τη βοήθεια των καταχωρήσεων της βάσης δεδομένων θα καταχωρηθεί ο αριθμός της κάμερας που καταγράφει την κίνηση και η χρονοσφραγίδα.

Από εκεί και πέρα, με τη χρήση της επιλογής "Enable Intelligent Features-Show Results", το σύστημα μέσω του αρχείου `intelligentfeatures.php` ελέγχει αν υπάρχει κίνηση σε κάποια από τις άλλες κάμερες ή αν δεν υπάρχει και είναι μέσα στα αποδεκτά χρονικά όρια που ορίστηκαν από τον χρήστη στο αρχείο διαχείρισης `intelligentfeaturesconf.php` και έτσι αποφασίζει αν πρόκειται για φυσιολογική ή μη κατάσταση έχοντας δεχθεί τα μηνύματα που θα πρέπει να εξάγει σε περίπτωση κανονικής κατάστασης ή κατάστασης συναγερμού, εργαζόμενο ως εξής:

Εάν δεν υπάρχει ανίχνευση κίνησης από την κάμερα 1, σημαίνει ότι ο ασθενής σηκώθηκε από το κρεβάτι και το σύστημα δρα με τον ακόλουθο τρόπο:

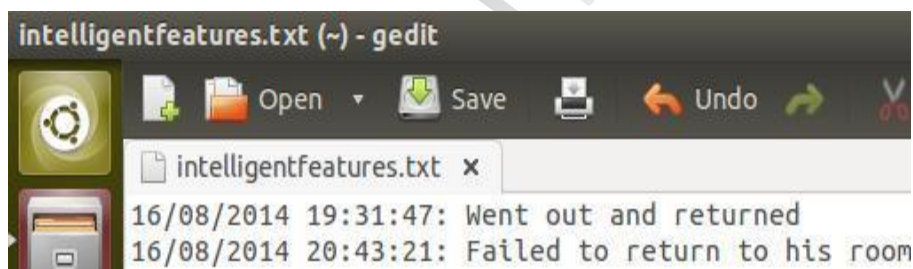
- Εάν εντοπιστεί κίνηση στην κάμερα 2, που όπως προτυπώθηκε βρίσκεται στην είσοδο της τουαλέτας (Location=Bathroom, Type=Door), σημαίνει ότι ο ασθενής εισήλθε στην τουαλέτα. Η παραμονή του ακριβώς στην είσοδο σημαίνει ανιχνεύσεις από την τρέχουσα κάμερα αλλά για ένα μικρό χρονικό διάστημα καθώς η κάμερα βρίσκεται μπροστά από την πόρτα, που σημαίνει ότι έπειτα από κάποια ελάχιστα δευτερόλεπτα εάν δεν υπάρχει πρόβλημα θα έχει καταφέρει να εισέλθει στο χώρο, καθώς για λόγους ιδιωτικότητας δεν τοποθετείται και άλλη κάμερα μέσα στο χώρο της τουαλέτας. Για το λόγο αυτό το χρονικό όριο (Normal Time Threshold) ορίστηκε στα 30 δευτερόλεπτα.

Μπορεί να μην υπάρχει κάμερα μέσα στο χώρο της τουαλέτας, όμως, η εφαρμογή θα μπορέσει να εντοπίσει εάν υπάρχει πρόβλημα μέσα σε αυτόν αφού έπειτα από ένα εύλογο χρονικό διάστημα (Idle Time το οποίο για το τρέχον σενάριο ορίστηκε στα 600 δευτερόλεπτα) προφανώς θα πρέπει να υπάρξει πάλι κίνηση στο οπτικό πεδίο της κάμερας, αφού ο επιτηρούμενος επιστρέφοντας θα κινηθεί προς την πόρτα. Εάν το χρονικό διάστημα ικανοποιεί τις επιλογές του αρχείου διαχείρισης `intelligentfeaturesconf.php` θα θεωρείται φυσιολογικό γεγονός και επομένως θα πρέπει να υπάρξουν καταχωρήσεις στο αρχείο `intelligentfeatures.txt` που καλείται από το `intelligentfeatures.php` με μήνυμα "Went to bathroom and returned". Εάν δεν εντοπιστεί κίνηση μέσα στο χρονικό όριο που ορίστηκε σημαίνει ότι ο παρακολουθούμενος δεν έχει βγει, υπάρχει κάποιο πρόβλημα, οπότε θα δημιουργείται alarm (κλήση script email/sms) και καταχωρήσεις στο αρχείο `intelligentfeatures.txt` με μήνυμα "Failed to return from bathroom".



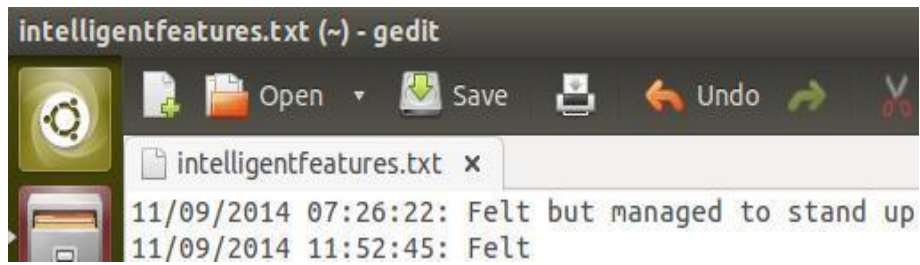
Εικόνα (5.3.1-2): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 2

- Το ίδιο και στην περίπτωση της κάμερας 3 που βρίσκεται εγκατεστημένη στην πόρτα εξόδου του δωματίου (Location=Exit, Type=Door) όπου με ακριβώς την ίδια λογική ορίστηκε το χρονικό όριο (Normal Time Threshold) παραμονής εντός εμβέλειας της κάμερας σε 30 δευτερόλεπτα και το Idle Time (εκτός εμβελείας αυτής μέχρι την επάνοδο) σε 600 δευτερόλεπτα με φυσιολογικό μήνυμα "Went out and returned" και μήνυμα alarm "Failed to return to his room".



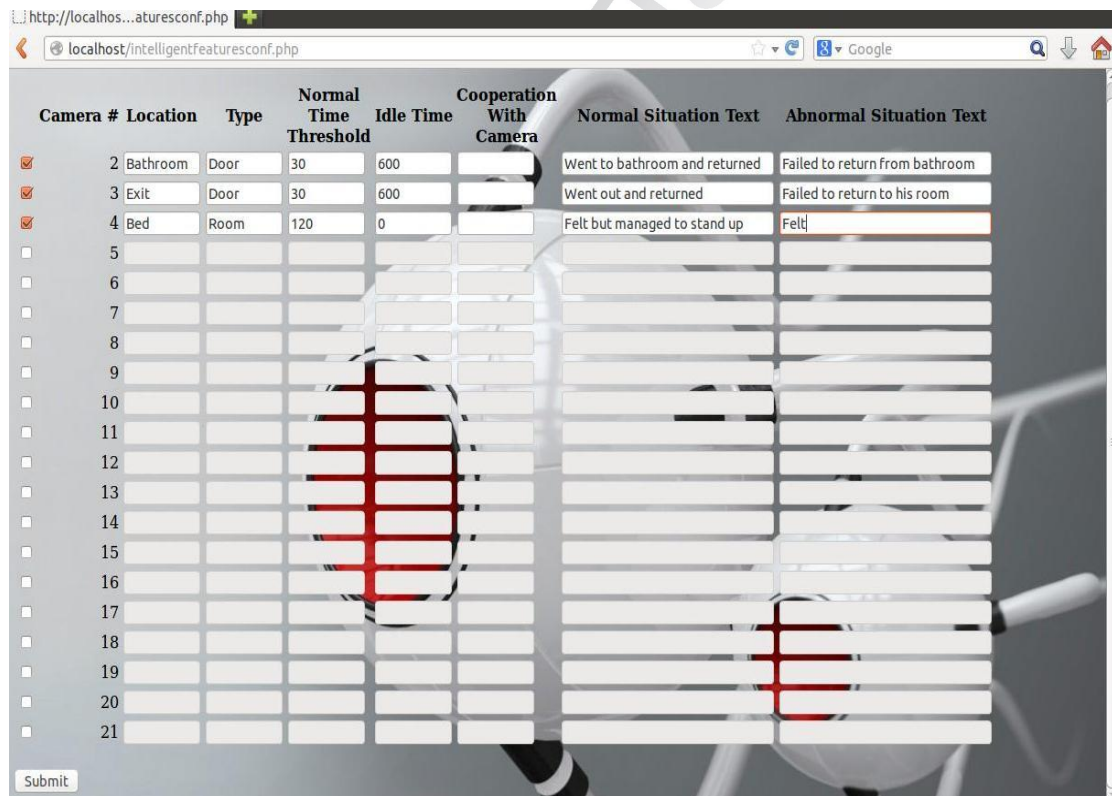
Εικόνα (5.3.1-3): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 3

- Στην περίπτωση της κάμερας 4 που στοχεύει χαμηλότερα της κύριας λήψης της κάμερας 1 στο κρεβάτι του ασθενούς (Location=Bed, Type=Door), εάν εντοπιστεί κίνηση σημαίνει ότι ο ασθενής έπεσε. Εάν συνεχίζεται να ανιχνεύεται κίνηση εκτός του χρονικού ορίου (Normal Time Threshold) που ορίστηκε για αυτήν την περίπτωση στα 120 δευτερόλεπτα, σημαίνει ότι δεν μπορεί να σηκωθεί, οπότε δημιουργία alarm (κλήση script email/sms) και καταχωρήσεις μηνυμάτων "Felt". Εάν, όμως, σταματήσει να ανιχνεύεται αποτελεί φυσιολογικό γεγονός καθώς κατάφερε να σηκωθεί και θα έχουμε εξαγωγή του μηνύματος "Felt but managed to stand up". Είναι ευνόητο, ότι στην συγκεκριμένη περίπτωση δεν είναι αναγκαίος ο ορισμός Idle Time καθώς δεν υπάρχει εναλλαγή στην κίνηση και ο έλεγχος μπορεί να γίνει μόνο μέσω του Normal Time Threshold.



Εικόνα (5.3.1-4): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 4

Στην παρακάτω εικόνα, παρατίθενται όλες οι επιλογές που έγιναν στο αρχείο διαχείρισης `intelligentfeaturesconf.php` για την περίπτωση του πρώτου σεναρίου: χώρος τοποθέτησης και τύπος της κάθε κάμερας, χρονικά όρια παραμονής του παρακολουθούμενου στο πεδίο παρατήρησής της καθεμίας καθώς και τα μηνύματα που εξάγονται σε περίπτωση φυσιολογικής ή μη συμπεριφοράς:



Εικόνα (5.3.1-5): Επιλογές του διαχειριστή στο αναδυόμενο αρχείο διαχείρισης για το σενάριο 1

5.3.2 : Σενάριο 2- Κατ'οίκον παρακολούθηση ασθενούς με πολλαπλές κάμερες σε πολλαπλά δωμάτια

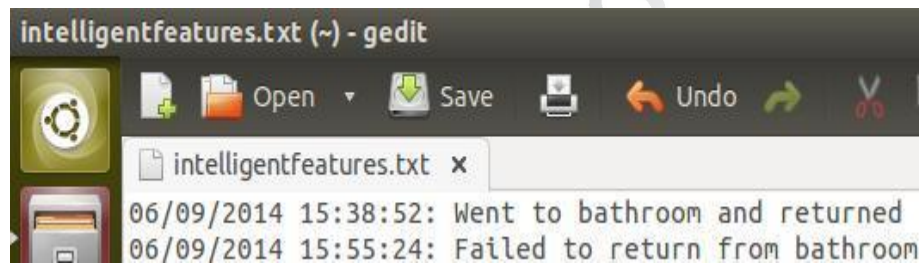
Στη δεύτερη περίπτωση, το σύστημα προσαρμόστηκε σε συνθήκες βιντεοεπιτήρησης σε περιβάλλον κατ' οίκον παρακολούθησης. Η ύπαρξη περισσότερων διεξόδων και δωματίων αυξάνουν την πιθανή καθημερινή δραστηριότητα του ατόμου που παρακολουθείται και επομένως και την πολυπλοκότητα στην υλοποίηση, αφού χρησιμοποιούνται αρκετά περισσότερες κάμερες, καθώς και στην ρυθμισιμότητα του συστήματος. Συνολικά, χρησιμοποιούνται δεκατρείς (13) κάμερες. Ο υποθετικός χώρος του σεναρίου αυτού αποτελείται από μια οικία χωρισμένη σε δυο διαφορετικά επίπεδα. Φυσικά, και στα δυο τοποθετούνται κάμερες οι οποίες για τεχνικούς λόγους είναι πλέον ασύρματες ip κάμερες. Στο πρώτο επίπεδο τοποθετείται η κύρια κάμερα όπου είναι το κρεβάτι του ασθενούς και είναι λογικό να περνά την περνάει την περισσότερη ώρα της ημέρας εκεί. Οι υπόλοιπες κάμερες τοποθετούνται προσαρμοσμένες στο καθημερινό πρόγραμμα όπως έχει αναλυθεί από τους διαχειριστές είτε σε εισόδους δωματίων είτε εντός των δωματίων δίνοντας τη δυνατότητα επιλογής του τύπου της κάμερας μεταξύ Door και Room ώστε (όταν κρίνεται χρήσιμο) να μπορεί να γίνεται η συνεκτίμηση και από τις δύο κάμερες πριν τη λήψη απόφασης σε κάθε υποσενάριο. Επιπρόσθετα, έχουν μελετηθεί και τα χρονικά όρια μέσα στα οποία θεωρούνται φυσιολογικές οι ανιχνεύσεις και οι εναλλαγές κίνησης σε κάθε μια από τις κάμερες που αποδίδουν τα όρια κίνησης και έχει ενεργοποιηθεί και η ανάλογη εξαγωγή μηνυμάτων Normal Situation Text και Abnormal Situation Text.

Έτσι, λοιπόν, το σύστημα ξεκινώντας ελέγχει αν υπάρχει κίνηση στην κάμερα 1 ή αν δεν υπάρχει και είναι μέσα στο αποδεκτό χρονικό όριο που ορίζεται από τον χρήστη μέχρι να ξαναυπάρξει και αποφασίζει αν πρόκειται για φυσιολογική ή μη κατάσταση.

Εάν δεν υπάρχει ανίχνευση κίνησης στην κάμερα 1 που σημαίνει ότι ο παρακολουθούμενος έχει εξέλθει από το πεδίο παρατήρησης της, υπάρχουν τα εξής ενδεχόμενα που αφορούν τα χρονικά όρια και τα μηνύματα που ελέγχονται από τα πεδία επιτήρησης των υπολοίπων καμερών και τα οποία ο διαχειριστής δηλώνει στην πλατφόρμα του αρχείου διαχείρισης `intelligentfeaturesconf.php`:

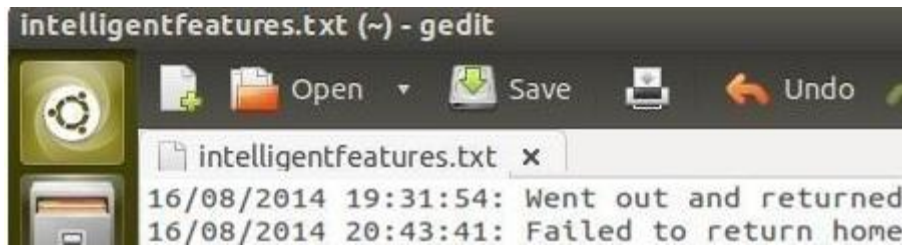
- Εντοπισμός κίνησης στην κάμερα 2 που είναι τοποθετημένη στην είσοδο της τουαλέτας (Location=Bathroom, Type=Door), σημαίνει ότι ο παρακολουθούμενος εισήλθε στην τουαλέτα. Η παραμονή του ακριβώς στην είσοδο σημαίνει ανιχνεύσεις από την τρέχουσα κάμερα αλλά για ένα μικρό χρονικό διάστημα καθώς η κάμερα βρίσκεται μπροστά από την πόρτα, που σημαίνει ότι έπειτα από κάποια ελάχιστα δευτερόλεπτα εάν δεν υπάρχει πρόβλημα θα έχει καταφέρει να εισέλθει στο χώρο, καθώς για λόγους ιδιωτικότητας δεν τοποθετείται και άλλη κάμερα μέσα στο χώρο της τουαλέτας. Για το λόγο αυτό, το χρονικό όριο (Normal Time Threshold) ορίστηκε στα 30 δευτερόλεπτα.

Μπορεί να μην υπάρχει κάμερα μέσα στο χώρο της τουαλέτας, όμως, η εφαρμογή θα μπορέσει να εντοπίσει εάν υπάρχει πρόβλημα μέσα σε αυτόν αφού έπειτα από ένα εύλογο χρονικό διάστημα (Idle Time το οποίο για το τρέχον σενάριο ορίστηκε στα 600 δευτερόλεπτα) προφανώς θα πρέπει να υπάρξει πάλι κίνηση στο οπτικό πεδίο της κάμερας, αφού ο επιτηρούμενος επιστρέφοντας θα κινηθεί προς την πόρτα. Εάν το χρονικό διάστημα ικανοποιεί τις επιλογές του αρχείου διαχείρισης `intelligentfeaturesconf.php` θα θεωρείται φυσιολογικό γεγονός και επομένως θα πρέπει να υπάρξουν καταχωρήσεις στο αρχείο `intelligentfeatures.txt` που καλείται από το `intelligentfeatures.php` με μήνυμα "Went to bathroom and returned". Εάν δεν εντοπιστεί κίνηση μέσα στο χρονικό όριο που ορίστηκε σημαίνει ότι ο παρακολουθούμενος δεν έχει βγει, υπάρχει κάποιο πρόβλημα, οπότε θα δημιουργείται alarm (κλήση script email/sms) και καταχωρήσεις στο αρχείο `intelligentfeatures.txt` με μήνυμα "Failed to return from bathroom".



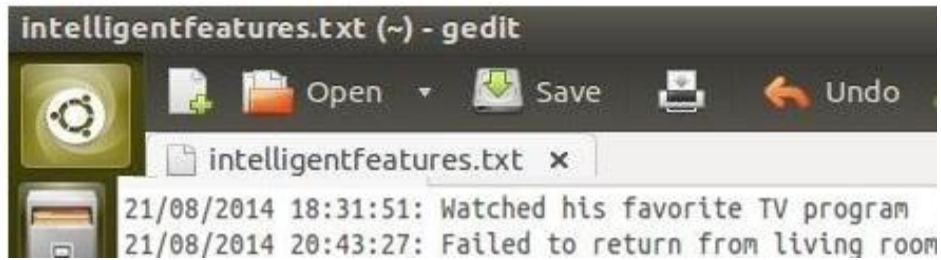
Εικόνα (5.3.2-1): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 2

- Εντοπισμός κίνησης στην κάμερα 3 που βρίσκεται στην είσοδο/έξοδο της οικίας (Location=Exit, Type=Door) όπου με ακριβώς την ίδια λογική ορίστηκε το χρονικό όριο (Normal Time Threshold) παραμονής εντός εμβέλειας της κάμερας σε 30 δευτερόλεπτα και το Idle Time σε 600 δευτερόλεπτα (που είναι αρκετά σε περίπτωση που απλά άνοιξε την πόρτα για οποιοδήποτε λόγο) με φυσιολογικό μήνυμα "Went out and returned", αλλά αν τα λεπτά αυτά ξεπεραστούν σημαίνει ότι ο παρακολουθούμενος κινήθηκε προς την έξοδο του σπιτιού ασυνόδευτος και γίνεται εξαγωγή μηνύματος alarm " Failed to return home".



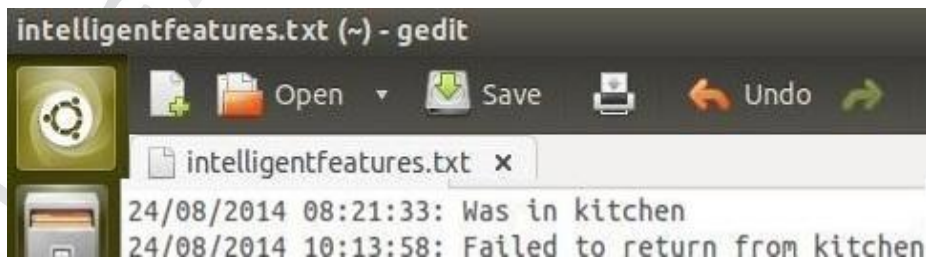
Εικόνα (5.3.2-2): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 3

- Εντοπισμός κίνησης στην κάμερα 4 που βρίσκεται στην είσοδο του σαλονιού (Location=Living Room, Type=Door). Σε αυτή την περίπτωση χρησιμοποιείται και δεύτερη κάμερα, (η κάμερα 5) η οποία βρίσκεται μπροστά από τον καναπέ και την τηλεόραση (Location=Living Room, Type=Room). Το σύστημα θα συνεκτιμήσει τα αποτελέσματα και από τις δύο κάμερες πριν εξάγει την τελική απόφαση. Για το λόγο αυτό, ο διαχειριστής σε αυτή την περίπτωση στο αρχείο configuration επιλέγει για την κάμερα 4 Cooperation With Camera 5 και θέτει την κάμερα 4 σαν Door κάμερα, ενώ την 5 σαν Room κάμερα. Όσον αφορά την κάμερα 4, με ίδια λογική όπως στις προηγούμενες περιπτώσεις, ορίστηκε το χρονικό όριο (Normal Time Threshold) παραμονής εντός εμβέλειας της κάμερας σε 30 δευτερόλεπτα (αρκετά για να εισέλθει από το κατώφλι της πόρτας) και το Idle Time σε 7200 δευτερόλεπτα διότι, υποθετικά, είναι γνωστό ότι ο ασθενής παρακολουθεί ένα συγκεκριμένο πρόγραμμα για 2 ώρες στην τηλεόραση οπότε η παραμονή του στο χώρο του σαλονιού θεωρείται λογική. Για να ελαχιστοποιηθεί το ενδεχόμενο ψευδούς συναγερμού η κάμερα 4 θα συνεκτιμήσει το αποτέλεσμα από την κάμερα 5 η οποία είναι μέσα στο δωμάτιο και στην οποία είναι λογικό να υπάρχει ανίχνευση και εναλλαγή της κίνησης με χρονικό όριο (Idle Time) των 5 ωρών (18000 δευτερόλεπτα). Είναι προφανές ότι δίνεται περιθώριο πολύ περισσότερου χρόνου παραμονής αφού για οποιοδήποτε λόγο ο παρακολουθούμενος μπορεί να παραμείνει στο δωμάτιο χωρίς να υπάρχει κάποιο πρόβλημα αφού θα υπάρχουν ανιχνεύσεις. Η παραμονή εντός αυτών των χρονικών ορίων θεωρείται φυσιολογική με φυσιολογικά μηνύματα "Watched his favorite TV program", ενώ αν ο χρόνος ξεπεραστεί και δεν υπάρχει ανίχνευση από καμία από τις δύο κάμερες, εξάγεται μήνυμα alarm "Failed to return from living room".



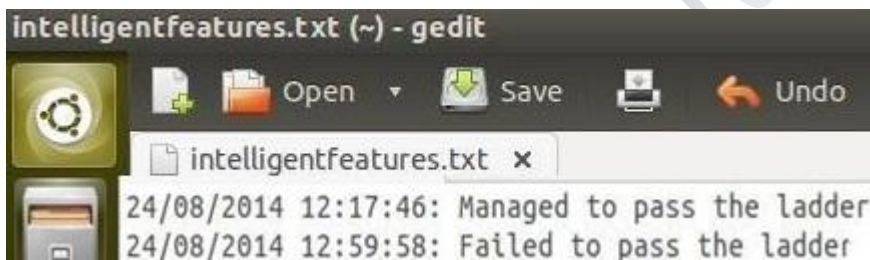
Εικόνα (5.3.2-3): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση των καμερών 4 και 5

- Εντοπισμός κίνησης στην κάμερα 6 που βρίσκεται στην είσοδο της κουζίνας (Location=Kitchen, Type=Door). Όπως προηγουμένως, και σε αυτή την περίπτωση χρησιμοποιείται και δεύτερη κάμερα, (η κάμερα 7) η οποία βρίσκεται μέσα στο χώρο της κουζίνας (Location=Kitchen, Type=Room). Και εδώ, το σύστημα θα συνεκτιμήσει τα αποτελέσματα και από τις δύο κάμερες πριν εξάγει την τελική απόφαση. Για το λόγο αυτό, ο διαχειριστής στο αρχείο configuration επιλέγει για την κάμερα 6 Cooperation With Camera 7 και θέτει την κάμερα 6 σαν Door κάμερα, ενώ την 7 σαν Room κάμερα. Έτσι, η κάμερα 6 σαν Door κάμερα θα λάβει ένα χρονικό όριο (Normal Time Threshold) παραμονής εντός της εμβέλειας της 30 δευτερόλεπτων, ενώ Idle Time 1800 δευτερόλεπτα αφού η παραμονή του ασθενούς εντός του χώρου για το πολύ 30 λεπτά, κρίνεται λογική. Αντίστοιχα, στην κάμερα 7 που βρίσκεται εντός του δωματίου είναι λογικό να υπάρχουν συνέχεια εναλλαγές στην ανίχνευση κίνησης οπότε εάν ικανοποιείται και αυτή η συνθήκη με ένα χρονικό όριο (Idle Time) της τάξης των 9000 δευτερολέπτων (δίνεται περιθώριο πολύ περισσότερου χρόνου παραμονής αφού για οποιοδήποτε λόγο ο παρακολουθούμενος μπορεί να παραμείνει στο δωμάτιο χωρίς να υπάρχει κάποιο πρόβλημα αφού θα υπάρχουν ανιχνεύσεις) γίνεται εξαγωγή φυσιολογικού μηνύματος "Was is in the kitchen" ενώ σε αντίθετη περίπτωση εξάγεται συναγερμός με μήνυμα "Failed to return from kitchen".



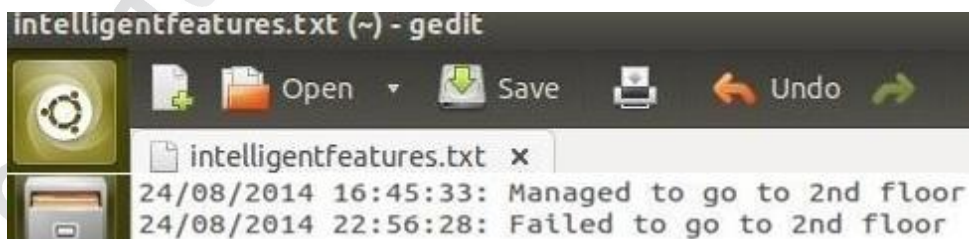
Εικόνα (5.3.2-4): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση των καμερών 6 και 7

- Εντοπισμός κίνησης στην κάμερα 8 που βρίσκεται στην αρχή της σκάλας στον κάτω όροφο (Location=Down Stair) που σημαίνει ότι ο παρακολουθούμενος ξεκίνησε την άνοδο προς το δεύτερο όροφο. Η παραμονή του εντός της εμβέλειας της κάμερας και κατ' επέκταση των ανιχνεύσεων-καταχωρήσεων είναι λογικό να υφίσταται για ένα μέγιστο χρονικό όριο (Normal Time Threshold) της τάξης των 120 δευτερολέπτων, καθώς θα πρέπει μετά από 2 λεπτά να σταματήσει να υπάρχει ανίχνευση που σημαίνει ότι ο παρακολουθούμενος κατάφερε να ξεκινήσει την άνοδο προς την σκάλα και επομένως θα εξαχθεί φυσιολογικό μήνυμα με τίτλο "Managed to pass the ladder". Σε αντίθετη περίπτωση, έχουμε φυσικά δημιουργία συναγερμού που θα απεικονίζει το μήνυμα "Failed to pass the ladder".

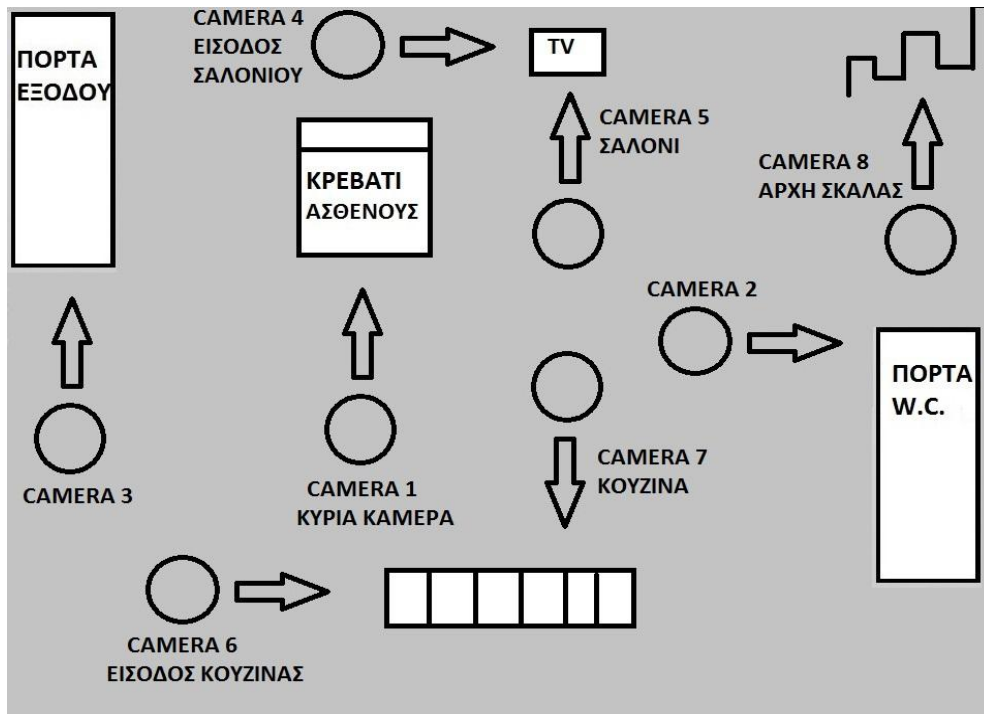


Εικόνα (5.3.2-5): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 8

- Εντοπισμός κίνησης στην κάμερα 9 που βρίσκεται στο τέλος της σκάλας στον δεύτερο όροφο (Location=Up Stair) που σημαίνει ότι ο παρακολουθούμενος προσπαθεί να ολοκληρώσει την άνοδο προς το δεύτερο όροφο. Θα πρέπει και πάλι μέσα σε 2 λεπτά να σταματήσει να υπάρχει ανίχνευση που σημαίνει ότι ανέβηκε τις σκάλες επιτυχώς και επομένως θα εξαχθεί φυσιολογικό μήνυμα με τίτλο "Managed to go to 2nd floor". Σε αντίθετη περίπτωση, έχουμε φυσικά δημιουργία συναγερμού που θα απεικονίζει το μήνυμα "Failed to go to 2nd floor".



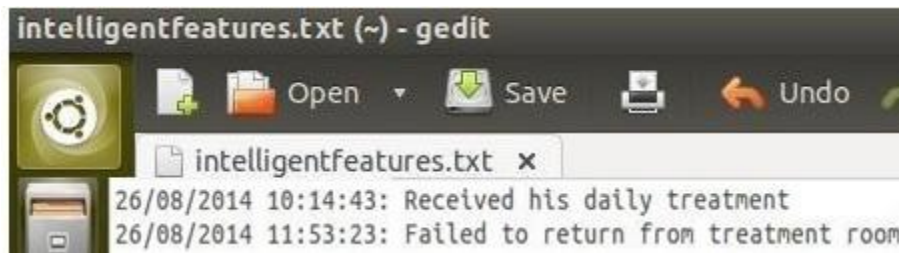
Εικόνα (5.3.2-6): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση της κάμερας 9



Εικόνα (5.3.2-7): Σενάριο 2- Κατ' οίκον παρακολούθηση ασθενούς με πολλαπλές κάμερες σε πολλαπλά δωμάτια (κάτω όροφος)

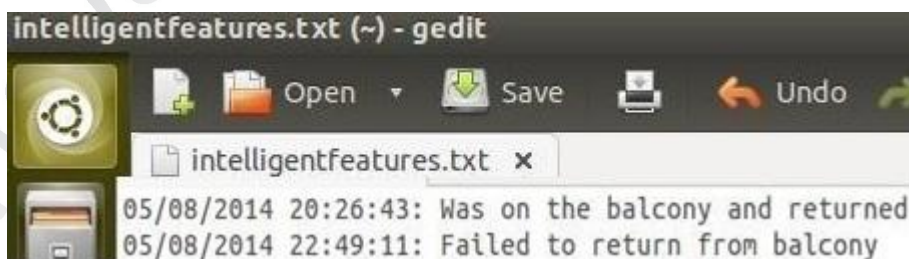
- Εντοπισμός κίνησης στην κάμερα 10 που βρίσκεται στο δεύτερο όροφο στο χώρο του treatment room και ακριβώς στην είσοδο αυτού (Location=Treatment Room, Type=Door), όπου είναι γνωστό ότι ο ασθενής παραμένει καθημερινά λόγω παρακολούθησης του από συγκεκριμένα όργανα μετρήσεων καθώς και για λήψη φαρμακευτικής αγωγής. Επίσης, και εδώ γίνεται συνεκτίμηση του αποτελέσματος με δεύτερη κάμερα (κάμερα 11) η οποία βρίσκεται μέσα στο χώρο (Location=Treatment Room, Type=Room). Για το λόγο αυτό, ο διαχειριστής στο αρχείο configuration επιλέγει για την κάμερα 10 Cooperation With Camera 11 και ορίζει την κάμερα 10 σαν Door κάμερα, ενώ την 11 σαν Room κάμερα. Έτσι, η παραμονή στην κάμερα 10 αφού πρόκειται για Door κάμερα σημαίνει ανιχνεύσεις από την τρέχουσα κάμερα αλλά για ένα μικρό χρονικό διάστημα καθώς η κάμερα βρίσκεται μπροστά από την πόρτα, που σημαίνει ότι έπειτα από κάποια ελάχιστα δευτερόλεπτα εάν δεν υπάρχει πρόβλημα θα έχει καταφέρει να εισέλθει στο χώρο. Για το λόγο αυτό το χρονικό όριο (Normal Time Threshold) ορίστηκε στα 30 δευτερόλεπτα. Στη συνέχεια, οι ανιχνεύσεις από το πεδίο της κάμερας είναι λογικό να σταματήσουν. Η παραμονή του ασθενούς εντός του χώρου (που σημαίνει ότι θα υπάρχουν ανιχνεύσεις μόνο από την κάμερα 11) για μια ώρα σημαίνει ότι ακολούθησε την διαδικασία κανονικά και θα κινηθεί και πάλι προς την έξοδο, δηλαδή προς την εμβέλεια της κάμερας 10 όπου και θα υπάρξουν καταχωρήσεις. Με βάση αυτή τη σκέψη, το Idle Time για την κάμερα 10 ορίστηκε στα 3600 δευτερόλεπτα, ενώ για την κάμερα 11 που βρίσκεται εντός του δωματίου είναι λογικό να υπάρχουν συνέχεια εναλλαγές στην ανίχνευση κίνησης οπότε δόθηκε ένα χρονικό όριο (Idle Time) της τάξης των 9000 δευτερολέπτων (δίνεται περιθώριο πολύ περισσότερο χρόνου παραμονής) και σε φυσιολογική κατάσταση (όπου δεν θα υπάρχει υπέρβαση

των ορίων και των 2 καμερών) θα εξαχθεί το μήνυμα "Received his daily treatment", ενώ σε αντίθετη περίπτωση θα υπάρξει το μήνυμα συναγερμού "Failed to return from treatment room".



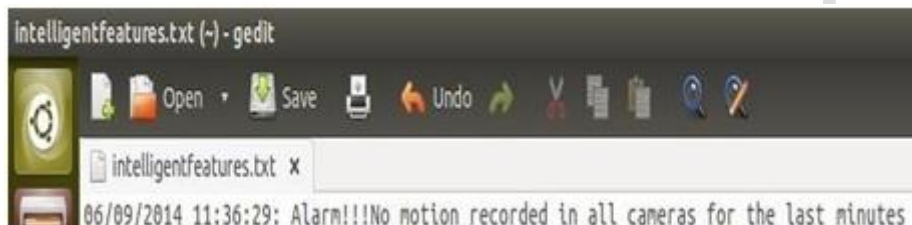
Εικόνα (5.3.2-8): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση των καμερών 10 και 11

- Εντοπισμός κίνησης στην κάμερα 12 που βρίσκεται στο δεύτερο όροφο στην έξοδο της μπαλκονόπορτας (Location=Balcony, Type=Door) που σημαίνει κίνηση προς την έξοδο του μπαλκονιού. Στο χώρο του μπαλκονιού υπάρχει και δεύτερη κάμερα (κάμερα 13) για συνεκτίμηση του αποτελέσματος (Location=Balcony, Type=Room). Για το λόγο αυτό, ο διαχειριστής στο αρχείο configuration επιλέγει για την κάμερα 12 Cooperation With Camera 13 και θέτει την κάμερα 12 σαν Door κάμερα, ενώ την 13 σαν Room κάμερα. Έτσι, για την κάμερα 12 θα υπάρχουν ανιχνεύσεις για ένα μικρό χρονικό διάστημα με χρονικό όριο (Normal Time Threshold) στα 30 δευτερόλεπτα. Στη συνέχεια, οι ανιχνεύσεις από το πεδίο της κάμερας 12 είναι λογικό να σταματήσουν. Η παραμονή του ασθενούς εντός του χώρου του μπαλκονιού και επομένως εντός της εμβέλειας της κάμερας 13 για 15 λεπτά κρίνεται φυσιολογική οπότε ορίζεται Idle Time της τάξης των 900 δευτερολέπτων για την κάμερα 12 αφού είναι λογικό να ξαναυπάρξουν ανιχνεύσεις για αυτή κατά την κίνηση προς την πόρτα. Με αυτή τη λογική, δίνοντας ένα περιθώριο πολύ περισσότερου χρόνου παραμονής στο χώρο του μπαλκονιού ορίζεται για την κάμερα 13 ένα Idle Time 3600 δευτερολέπτων και σε φυσιολογική κατάσταση (όπου δεν θα υπάρχει υπέρβαση των ορίων και των 2 καμερών) θα εξαχθεί το μήνυμα "Was on the balcony and returned" ενώ το μήνυμα συναγερμού αν ξεπεραστούν τα προαναφερθέντα χρονικά όρια θα είναι "Failed to return from balcony".



Εικόνα (5.3.2-9): Εξαγωγή αποτελεσμάτων μέσω φυσιολογικών ή μη μηνυμάτων στη διεπαφή για την περίπτωση των καμερών 12 και 13

- Κανένας εντοπισμός κίνησης σε περίπτωση που δεν υπάρχουν καθόλου καταχωρήσεις από καμία κάμερα τα τελευταία 300 λεπτά και πάλι σημάνει συναγερμό καθώς αποτελεί μη φυσιολογικό γεγονός, αφού είναι αδύνατον υπό φυσιολογικές συνθήκες καμία κάμερα να μην ανιχνεύει κίνηση για μεγάλο χρονικό διάστημα και επομένως παραπέμπει πρόβλημα στην κατάσταση του παρακολουθούμενου, με μήνυμα συναγερμού "Alarm!!!No motion recorded in all cameras for the last minutes".

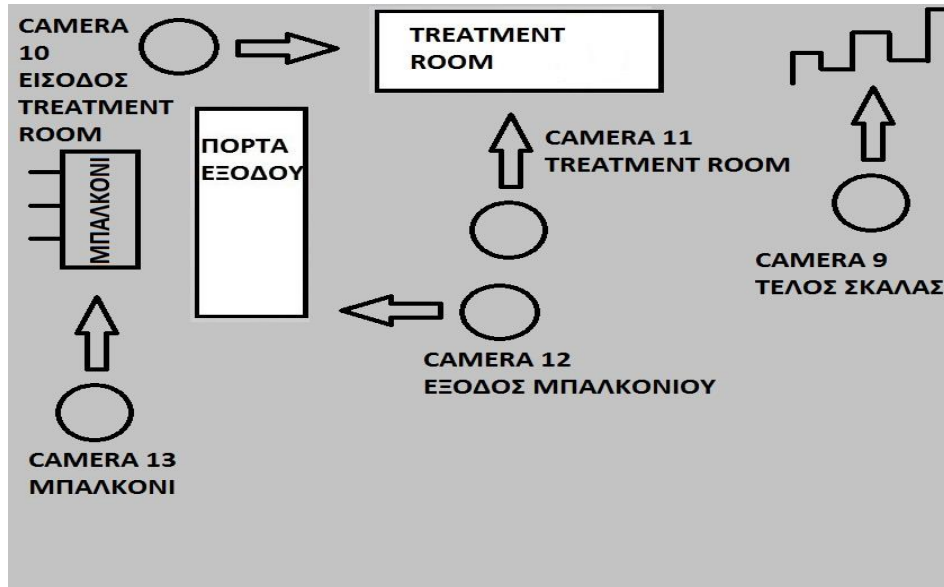


Εικόνα (5.3.2-10): Εξαγωγή μηνύματος συναγερμού στη διεπαφή για την περίπτωση μη καταχώρησης από καμία κάμερα

Στην παρακάτω εικόνα, παρατίθενται όλες οι επιλογές που έγιναν στο αρχείο διαχείρισης intelligentfeaturesconf.php για την περίπτωση του δευτέρου σεναρίου: (χώρος τοποθέτησης της κάθε κάμερας, χρονικά όρια παραμονής του παρακολουθούμενου στο πεδίο παρατήρησής της κάθε μιας καθώς και μηνύματα που εξάγονται σε περίπτωση φυσιολογικής ή μη συμπεριφοράς):

Camera #	Location	Type	Normal Time Threshold	Idle Time	Cooperation With Camera	Normal Situation Text	Abnormal Situation Text	
<input checked="" type="checkbox"/>	2	Bathroom	Door	30	600		Went to bathroom and returned	Failed to return from bathroom
<input checked="" type="checkbox"/>	3	Exit	Door	30	600		Went out and returned	Failed to return home
<input checked="" type="checkbox"/>	4	Living room	Door	30	7200	5	Watched his favourite TV program	Failed to return from living room
<input checked="" type="checkbox"/>	5	Living room	Room		18000	4	Watched his favourite TV program	Failed to return from living room
<input checked="" type="checkbox"/>	6	Kitchen	Door	30	1800	7	Was in kitchen	Failed to return from kitchen
<input checked="" type="checkbox"/>	7	Kitchen	Room		9000	6	Was in kitchen	Failed to return from kitchen
<input checked="" type="checkbox"/>	8	Down Stair	Standalone	120	0		Managed to pass the ladder	Failed to pass the ladder
<input checked="" type="checkbox"/>	9	Up Stair	Standalone	120	0		Managed to go to 2nd floor	Failed to go to 2nd floor
<input checked="" type="checkbox"/>	10	Treatment r	Door	30	3600	11	Received his daily treatment	Failed to return from treatment r
<input checked="" type="checkbox"/>	11	Treatment r	Room		9000	10	Received his daily treatment	Failed to return from treatment r
<input checked="" type="checkbox"/>	12	Balcony	Door	30	900	13	Was on the balcony and returned	Failed to return from balcony
<input checked="" type="checkbox"/>	13	Balcony	Room		3600	12	Was on the balcony and returned	Failed to return from balcony
<input type="checkbox"/>	14							
<input type="checkbox"/>	15							
<input type="checkbox"/>	16							
<input type="checkbox"/>	17							
<input type="checkbox"/>	18							
<input type="checkbox"/>	19							
<input type="checkbox"/>	20							
<input type="checkbox"/>	21							

Εικόνα (5.3.2-11): Επιλογές του διαχειριστή στο αναδυόμενο αρχείο διαχείρισης για το σενάριο 2



Εικόνα (5.3.2-12): Σενάριο 2- Κατ' οίκον παρακολούθηση ασθενούς με πολλαπλές κάμερες σε πολλαπλά δωμάτιο (επάνω όροφος)

- Επιπρόσθετα, με τη βοήθεια του check.c αν ενεργοποιηθούν περισσότερες από 2 κάμερες ταυτόχρονα έχουμε και πάλι δημιουργία συναγερμού καθώς και σε αυτήν την περίπτωση απεικονίζεται μια μη φυσιολογική κατάσταση (βρίσκεται στο χώρο κάποιος μη εξουσιοδοτημένος, τεχνική βλάβη κτλ).

Έτσι, λοιπόν, μεταβάλλεται το αρχείο check.conf στην περίπτωση χρησιμοποίησης όλων των καμερών του σεναρίου 2 (13 κάμερες) και χρονικού ορίου εξήντα δευτερολέπτων:

```
##### f
#configuration file for check
##### f

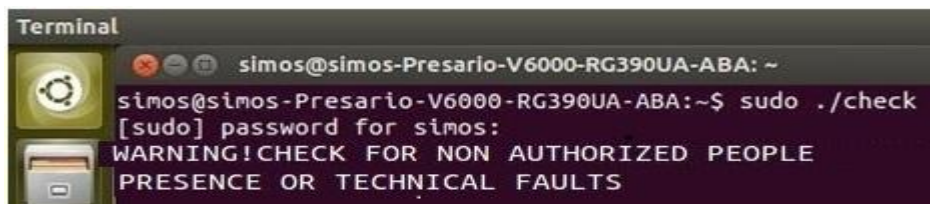
totalcameras 13

threshold 60

camera_output output1
camera_output output2
camera_output output3
camera_output output4
camera_output output5
camera_output output6
camera_output output7
camera_output output8
camera_output output9
camera_output output10
camera_output output11
camera_output output12
camera_output output13
```


Και η μόνη αλλαγή στο αρχείο `check.c` είναι το μήνυμα εκτύπωσης στη γραμμή εντολών:

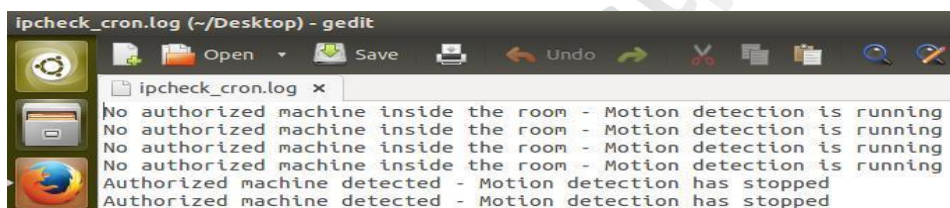
```
if ((hit==totalcameras) &&(tempre!=0)) {printf("WARNING!CHECK FOR  
NON AUTHORIZED PEOPLE PRESENCE OR TECHNICAL FAULTS \n");};;
```



```
Terminal
simos@simos-Presario-V6000-RG390UA-ABA: ~
simos@simos-Presario-V6000-RG390UA-ABA:~$ sudo ./check
[sudo] password for simos:
WARNING!CHECK FOR NON AUTHORIZED PEOPLE
PRESENCE OR TECHNICAL FAULTS
```

Εικόνα (5.3.2-13): Μηνύματα στη γραμμή εντολών έπειτα από την εκτέλεση του προγράμματος `check.c` για έλεγχο του συστήματος

- Η ύπαρξη εξουσιοδοτημένων ανθρώπων στο χώρο εξασφαλίζεται με τη βοήθεια του script `ipcheck.sh` που περιγράφηκε στην παράγραφο 5.1.2.



```
ipcheck_cron.log (~/Desktop) - gedit
ipcheck_cron.log x
No authorized machine inside the room - Motion detection is running
No authorized machine inside the room - Motion detection is running
No authorized machine inside the room - Motion detection is running
No authorized machine inside the room - Motion detection is running
Authorized machine detected - Motion detection has stopped
Authorized machine detected - Motion detection has stopped
```

Εικόνα (5.3.2-14): Μηνύματα στο log του Crontab έπειτα από την εκτέλεση του προγράμματος `ipcheck.sh`

- Με τη βοήθεια των τεχνικών που αναλύθηκαν στην παράγραφο 5.1.1 αποφασίζεται η παύση της ανίχνευσης από όλες τις κάμερες αν σε συγκεκριμένες ώρες δεν είναι αναγκαίο να υπάρχει ανίχνευση καθώς είναι γνωστό ότι ο παρακολουθούμενος δέχεται προγραμματισμένες επισκέψεις, απουσιάζει από το χώρο, αναπαύεται κτλ.

Έστω ότι τα ωράρια για παύση της ανίχνευσης στο συγκεκριμένο σενάριο έχουν ως εξής:

Προγραμματισμένες επισκέψεις: Τετάρτη 20.00-22.00
και Σαββατοκύριακα 9.00-11.00

Απουσία από το χώρο: 8.00-9.00 και 17.00-18.00 καθημερινά
εκτός Σαββατοκύριακου

Ώρες ανάπαυσης: 0.00-8.00 και 15.00-17.00 καθημερινά

Στο αρχείο configuration του Linux Crontab οι παραπάνω ημέρες και ώρες απεικονίζονται και ρυθμίζονται ως εξής, ενώ προστίθενται και οι εντολές για παύση/εκκίνηση:

```
# m h dom mon dow    command
00 08-09 * * 3 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 09-11 * * 6-7 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 08-09 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 17-18 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 00-08 * * * wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
00 15-17 * * * wget -q --delete-after
http://localhost:8080/0/detection/pause 2>&1 >/dev/null
01 09 * * 3 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 11 * * 6-7 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 09 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 18 * * 1-5 wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 08 * * * wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
01 17 * * * wget -q --delete-after
http://localhost:8080/0/detection/start 2>&1 >/dev/null
```

- Ακόμη, είναι απαραίτητο να υλοποιηθεί ένα φίλτρο πτώσης ώστε το σύστημα να ανιχνεύει πιθανές πτώσεις του παρακολουθούμενου και να γεννά συναγερμούς. Το λογισμικό Motion, όπως προτυπώθηκε στην παράγραφο 3.1, σε κάθε ανίχνευση έχει τη δυνατότητα να εξάγει πληροφορίες για την εικόνα που δημιουργήθηκε έπειτα από την ανίχνευση. Μερικές από τις πληροφορίες αυτές είναι η ημερομηνία και ώρα ανίχνευσης, ο αριθμός της κάμερας που προχώρησε σε αυτή, ο αριθμός των αλλαγμένων pixels και καρέ καθώς και το επίπεδο του θορύβου.

Επιπρόσθετα, στην ίδια παράγραφο αναλύθηκε η δυνατότητα του Motion να σχηματίζει ένα ορθογώνιο στο σημείο της εικόνας που εντοπίζεται κίνηση, που όπως είδαμε, ενεργοποιείται με την επιλογή Locate.



Εικόνα (5.3.2.-15): Εικόνα που δημιουργήθηκε έπειτα από ανίχνευση κίνησης έχοντας ενεργοποιημένες τις επιλογές locate και text

Εκτός από την προσθήκη των πληροφοριών που αναφέρθηκαν παραπάνω, το Motion έχει την δυνατότητα (και πάλι αναφέρθηκε στην παράγραφο 3.1) να εξάγει πληροφορίες και για το ορθογώνιο στο οποίο εντοπίζεται η κίνηση. Συγκεκριμένα, μπορεί να εντοπίσει και να εξάγει το μήκος και το πλάτος του ορθογωνίου καθώς και τις συντεταγμένες του κέντρου της κίνησης, δηλαδή του ορθογωνίου.

```
# Text Display
# %Y = year, %m = month, %d = date,
# %H = hour, %M = minute, %S = second, %T = HH:MM:SS,
# %v = event, %q = frame number, %t = thread (camera) number,
# %D = changed pixels, %N = noise level, \n = new line,
# %i and %j = width and height of motion area,
# %K and %L = X and Y coordinates of motion center
# %C = value defined by text_event - do not use with
text_event!
# You can put quotation marks around the text to allow
# leading spaces
```

Επομένως, για την περίπτωση που θέλουμε να μελετήσουμε τις διαστάσεις του ορθογωνίου αρκεί να εκμεταλλευτούμε τις μεταβλητές %i και %j, ενώ για τις συντεταγμένες όπου ανιχνεύεται η κίνηση υπεύθυνες είναι οι μεταβλητές %K και %L.

Για να αντλήσουμε και να αποθηκεύσουμε τις πληροφορίες αυτές, αρκεί να δηλωθούν στη βάση δεδομένων, όπως έγινε στην παράγραφο 3.1.1, τα νέα πεδία.

Όπως έχει αναλυθεί σε παλαιότερη ερευνητική προσπάθεια [5], όταν ο παρακολουθούμενος είναι σε όρθια στάση οι διαστάσεις του ορθογωνίου που ανιχνεύει την κίνηση διαφέρουν από τις αντίστοιχες όταν είναι πεσμένος.

Αναλυτικότερα, όταν ο παρακολουθούμενος είναι σε όρθια στάση, προφανώς το ορθογώνιο θα έχει μεγαλύτερο μήκος (μεταβλητή %i) και μικρότερο πλάτος (μεταβλητή %J). Σε αντίθετη περίπτωση, θα έχει μικρότερο μήκος και μεγαλύτερο πλάτος και έτσι είναι δυνατόν να ανιχνευθεί η πτώση.

Για να προστεθεί, λοιπόν, αυτό το φίλτρο στο σύστημα που έχει υλοποιηθεί αρκεί να προστεθούν στον πίνακα "simosss" της βάσης δεδομένων simos τα πεδία:

- height_dimension (int) - Μήκος (%i) του ορθογωνίου που ανιχνεύει την κίνηση.
- width_dimension (int) - Πλάτος (%J) του ορθογωνίου που ανιχνεύει την κίνηση.

```
mysql> CREATE TABLE simosss (camera INT, time_stamp  
VARCHAR(60), height_dimension INT, width_dimension INT );
```

Αντίστοιχα, θα πρέπει να δηλωθεί το sql query στο αρχείο configuration του Motion:

```
sql_query insert into simosss(camera, time_stamp,  
height_dimension, width_dimension) values('%t', '%Y-%m-%d %T',  
'%i', '%J')
```

Με αυτόν τον τρόπο, κάθε φορά που ανιχνεύεται κίνηση θα καταχωρούνται έκτος από τον αριθμό της κάμερας και τη χρονοσφραγίδα και οι διαστάσεις του ορθογωνίου που λαμβάνει χώρα η ανίχνευση.

Για να ενσωματωθεί το φίλτρο αυτό, απομένει να δηλωθεί στο κύριο αρχείο (που αναπτύχθηκε στην παράγραφο 5.3) intelligentfeatures.php ότι όταν ο λόγος μήκος ορθογωνίου προς πλάτος είναι μεγαλύτερος ή ίσος του ένα (μεγαλύτερο μήκος, μικρότερο πλάτος) αποτελεί φυσιολογική κατάσταση καθώς ο παρακολουθούμενος είναι σε όρθια στάση, ενώ όταν είναι μικρότερος σημαίνει ότι ο παρακολουθούμενος έπεσε. Στην περίπτωση που έχει ανιχνευθεί η πτώση, το σύστημα θα περιμένει για ένα χρονικό διάστημα έως ότου θα προχωρήσει ξανά στον υπολογισμό του λόγου του ορθογώνιου ώστε να διαπιστωθεί εάν ο παρακολουθούμενος κατάφερε να σηκωθεί μετά την πτώση. Εάν και πάλι ο λόγος είναι μικρότερος σημαίνει συναγερμό (κλήση script email/sms) καθώς η σύγκριση αυτή οδηγεί στο συμπέρασμα ότι ο παρακολουθούμενος έπεσε και δεν μπορεί να σηκωθεί.

Παρακάτω, παρατίθενται οι προσθήκες που έγιναν στο αρχείο intelligentfeatures.php προκειμένου να ολοκληρωθεί και η προσθήκη του φίλτρου πτώσης:

```

$con = mysqli_connect("localhost", "root", "simos", "simos");
//Check connection
if (mysqli_connect_errno()) {
echo "Failed to connect to MySQL: " .
mysqli_connect_error()."\n";
}
$getdimenvalues = mysqli_query($con, "SELECT * FROM simosss
where height_dimension != 0 and width_dimension != 0");
if (!$getdimenvalues) {
echo "Error: " . mysqli_error($con)."\n";
}
$dimensionrow = mysqli_fetch_array($getdimenvalues);

$height = $dimensionrow['height_dimension'];
$width = $dimensionrow['width_dimension'];

if ($height / $width < 1) {
//If width is bigger than height check dimensions again after 2
minutes
sleep(120)
$getdimenvalues2 = mysqli_query($con, "SELECT * FROM simosss
where height_dimension != 0 and width_dimension != 0");
if (!$getdimenvalues2) {
echo "Error: " . mysqli_error($con)."\n";
}
$dimensionrow2 = mysqli_fetch_array($getdimenvalues2);

$height2 = $dimensionrow2['height_dimension'];
$width2 = $dimensionrow2['width_dimension'];
}
if ($height2 / $width2 < 1) {
echo
date("d/m/Y
H:i:s").".".$intelligentfeaturestable['abnormal_text']."\n";
//Print abnormal situation
$outcome=shell_exec('/home/simos/Desktop/mail_motion');
//External command
echo $outcome;
}
//Otherwise continue like before
else {.....}

```

- Επιπρόσθετα, εκμεταλλευόμενο την πληροφορία που δίνουν οι μεταβλητές %K και %L για τις συντεταγμένες όπου ανιχνεύεται η κίνηση, το σύστημα είναι ικανό (έπειτα από μελέτη του χώρου και παραμετροποίηση από το διαχειριστή) να γνωρίζει το περιεχόμενο του περιβάλλοντος (context awareness) της εικόνας που λαμβάνεται και έπειτα να καταγράφει το περιεχόμενο μόνο που είναι χρήσιμο στην ανίχνευση συναγερμών. Για παράδειγμα, στις κάμερες δωματίου που χρησιμοποιούνται στα σενάρια που δημιουργήθηκαν μπορεί να προστεθεί ένα ακόμη φίλτρο για παύση της

ανίχνευσης σε περιοχή που υπάρχει κρεβάτι, καναπές, τραπέζι κλπ, διότι είναι φυσιολογικό να υπάρχει κίνηση στα σημεία αυτά του δωματίου.

Οι διαχειριστές έπειτα από μελέτη του χώρου μπορούν να προσδιορίσουν και να καταγράψουν τις συντεταγμένες όπου υπάρχουν τα παραπάνω αντικείμενα, να τις καταχωρήσουν σε νέα πεδία του αρχείου διαχείρισης `intelligentfeaturesconf.php`, όπως αυτό αναλύθηκε στην παράγραφο 5.3, και έπειτα αφού ενημερωθεί και το κύριο αρχείο `intelligentfeatures.php` να θεωρείται φυσιολογικό γεγονός η ανίχνευση στις συντεταγμένες αυτές και να εξάγεται συναγερμός μόνο σε διαφορετική περίπτωση όπου η κίνηση καταγράφεται εκτός των συντεταγμένων και ικανοποιούνται όλες οι υπόλοιπες συνθήκες που αναφέρθηκαν κατά την κατασκευή του `intelligentfeatures.php`.

- Τέλος, με βάση τον γνωστό τύπο για τον υπολογισμό της ταχύτητας $v = \Delta x / \Delta t$, εάν οριστεί η απόσταση μεταξύ δύο καμερών είναι δυνατόν να υπολογιστεί η ταχύτητα του παρακολουθούμενου εάν ληφθούν υπόψη οι χρονοσφραγίδες από τις οποίες αυτός πέρασε από την εμβέλεια κάθε κάμερας. Έτσι, θα μπορεί να λαμβάνεται υπόψη και η ταχύτητα του παρακολουθούμενου πριν εξαχθεί συναγερμός.

Η σκέψη αυτή αποδεικνύεται αρκετά χρήσιμη κατά τη μελέτη των καμερών 8 και 9 που βρίσκονται στην αρχή και στο τέλος της σκάλας, αντίστοιχα. Σε περίπτωση που ο παρακολουθούμενος ξεκινήσει την κάθοδο από τη σκάλα από τον δεύτερο όροφο (δηλαδή στην εμβέλεια της κάμερας 9) προς τον πρώτο (εμβέλεια της κάμερας 8) και αναπτύξει μια ταχύτητα μεγαλύτερη από τη συνηθισμένη που έχει μελετηθεί, μπορεί να εξάγεται συναγερμός καθώς είναι πολύ πιθανό να σημαίνει πτώση.

Σε αυτήν την περίπτωση, οι διαχειριστές του συστήματος αρκεί να γνωρίζουν την απόσταση που έχουν τοποθετηθεί οι δύο κάμερες. Έτσι, γίνεται καταχώρηση της απόστασης x σε νέο πεδίο του αρχείου διαχείρισης `intelligentfeaturesconf.php` (όπως αυτό αναλύθηκε στην παράγραφο 5.3). Επίσης, σε νέο πεδίο καταχωρείται και η μέγιστη ταχύτητα v που θεωρείται φυσιολογική. Εάν αυτή η τιμή ξεπεραστεί θα δημιουργείται συναγερμός. Στη συνέχεια, το κύριο αρχείο `intelligentfeatures.php` λαμβάνοντας τις χρονοσφραγίδες που ανίχνευσε κίνηση η καθεμία από τις δυο προς μελέτη κάμερες, προγραμματίζεται ώστε να αφαιρεί τις χρονοσφραγίδες και να τις διαιρεί με την απόσταση x που έχει ήδη δοθεί από τους διαχειριστές στο αρχείο διαχείρισης. Με αυτόν τον τρόπο, υπολογίζει την ταχύτητα v που κινήθηκε ο παρακολουθούμενος και εκτελώντας μια σύγκριση με την μέγιστη αποδεκτή ταχύτητα, που έχει δοθεί στο αρχείο διαχείρισης, εάν είναι μικρότερη από αυτή θεωρείται φυσιολογικό γεγονός ενώ σε αντίθετη περίπτωση που υπερβαίνει την προκαθορισμένη ταχύτητα γεννάται συναγερμός.

ΚΕΦΑΛΑΙΟ 6-ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 Αξιολόγηση των προσθηκών

Συνοψίζοντας, από τα προηγούμενα κεφάλαια γίνεται αντιληπτή η χρησιμότητα του λογισμικού Motion σε συνθήκες βιντεοεπιτήρησης με σκοπό τη διασφάλιση της διακριτικότητας, ακόμα και σε λειτουργία με τις προεπιλεγμένες ρυθμίσεις.

Είναι προφανές, βέβαια, ότι έπειτα από τις προτάσεις και τις ρυθμίσεις που έλαβαν χώρα στα προηγούμενα κεφάλαια (Κεφ. 3, 4 και 5), δημιουργήθηκε ένα ευφυές δικτυοκεντρικό σύστημα βιντεοεπιτήρησης με χρήση του λογισμικού Motion το οποίο έγινε περισσότερο αποδοτικό, αφού αφετέρου μεν εκμηδενίστηκαν οι πιθανότητες ψευδών συναγερμών (false alarms) αφετέρου δε, διασφαλίστηκε η ιδιωτικότητα των παρακολουθούμενων αφού δεν υπάρχει καμία πληροφορία της εικόνας τους, προσαρμοζόμενο συνάμα σε συνθήκες παρακολούθησης ασθενών. Ένα ακόμα πλεονέκτημα είναι η μικρή πτώση της απόδοσης του συστήματος (αναλύεται στην παράγραφο 6.2) σε περίπτωση σύνδεσης πολλαπλών καμερών σε σχέση με αντίστοιχες τεχνικές που έχουν εφαρμοστεί (αναφέρθηκαν στην παράγραφο 1.5) και χρησιμοποιούν τεχνικές επεξεργασίας εικόνας καταναλώνοντας μεγάλο ποσοστό των πόρων του συστήματος στο οποίο εκτελούνται.

Στην περίπτωση της ελαχιστοποίησης έως και εκμηδενισμού των ψευδών συναγερμών, σημαντικό ρόλο κατέχει η προσθήκη των ευφυών χαρακτηριστικών που έγιναν στο Κεφάλαιο 5, καθώς είναι ιδιαίτερα χρήσιμο το σύστημα να γνωρίζει τη συνηθισμένη συμπεριφορά στην κίνηση του ασθενούς σε έναν ορισμένο χώρο βιντεοεπιτήρησης και να δημιουργεί συναγερμούς μόνο αν διακρίνει μη φυσιολογικές συμπεριφορές, να γνωρίζει το χρόνο των προγραμματισμένων επισκέψεων στο χώρο παρακολούθησης και να μη γεννά άσκοπους συναγερμούς (alarm events) και να αναγνωρίζει επίσης το έγκριτο προσωπικό που εισέρχεται στο χώρο. Επιπρόσθετα, πολύ μεγάλο μερίδιο αυτής της προσπάθειας λαμβάνει η ρύθμιση ώστε να συνεκτιμάται το αποτέλεσμα από όλες τις κάμερες που υπάρχουν στο χώρο και να γεννάται συναγερμός μόνο όταν όλες από αυτές (οι οποίες στοχεύουν στο ίδιο σημείο φυσικά) ανιχνεύσουν κίνηση.

Όλες οι παραπάνω ρυθμίσεις δεν θα είχαν το ίδιο αποτέλεσμα αν δεν είχαν οριστεί με προσοχή οι επιλογές για το κατώφλι, το θόρυβο και την ευαισθησία που έγιναν στο Κεφάλαιο 3 και οι οποίες διαφέρουν για κάθε χώρο παρακολούθησης ξεχωριστά. Αν για παράδειγμα, εφαρμοστεί το όλο εγχείρημα σε ένα νοσοκομείο, θα πρέπει να ληφθούν υπόψη από την αρχή όλες οι παράμετροι για το κάθε δωμάτιο

(όπως η φωτεινότητα, ο χώρος που είναι τοποθετημένο το κρεβάτι του ασθενή κτλ) και να ρυθμιστούν κατάλληλα.

Στο κομμάτι της διασφάλισης της διακριτικότητας, έγιναν αρκετές βελτιώσεις στο ήδη "διακριτικό" Motion. Αρχικά, χρησιμοποιήθηκε κατά κόρον η εξ' ορισμού ιδιότητά του η οποία είναι η εξαγωγή εικόνων τύπου Motion και όχι κανονικών εικόνων, έτσι ώστε στον τελικό χρήστη να απεικονίζεται μόνο η πληροφορία των αλλαγμένων καρέ αν υπάρχει κίνηση και όχι η εικόνα του ασθενούς (Κεφάλαιο 2, παράγραφος 3). Ακόμη, αναπτύχθηκαν οι ειδοποιήσεις του Motion όταν αυτό ανιχνεύει την κίνηση (Κεφάλαιο 3), έτσι ώστε να καταχωρεί τις ειδοποιήσεις σε βάση δεδομένων καθώς και να αποστέλλει μηνύματα email και SMS στο χρήστη με μοναδική πληροφορία την ύπαρξη της κίνησης και όχι την πληροφορία αυτής (πραγματική εικόνα). Με αυτόν τον τρόπο, εξασφαλίζεται η αμεσότητα για επέμβαση αν συμβεί κάποιο περιστατικό και ο ασθενής χρειαστεί βοήθεια, αφού ο χρήστης θα ειδοποιηθεί κατευθείαν όπου και να βρίσκεται και παράλληλα διατηρείται η ιδιωτικότητα του ασθενούς.

Τέλος, κρίνεται πάρα πολύ χρήσιμη η εξέλιξη του βασικού web interface (Κεφάλαιο 4), διότι όλες οι επιλογές που έχει στη διάθεσή του ο χρήστης συγκεντρώθηκαν σε μια κεντρική σελίδα διαχείρισης. Επιπλέον, προστέθηκαν οι δυνατότητες απεικόνισης και των συμβάντων που έχουν δημιουργηθεί, είτε σαν απλή ειδοποίηση είτε με τη μορφή της εικόνας Motion (ppm) ή της κανονικής εικόνας τη στιγμή που συνέβη το περιστατικό. Σε συνδυασμό με την προσθήκη των ρυθμίσεων για απομακρυσμένη πρόσβαση από οποιοδήποτε δίκτυο, διαπιστώνεται ότι εκτός από την ευκολία για διαχείριση που αποδόθηκε στο χρήστη, το λογισμικό προσαρμόστηκε πλήρως για την παρακολούθηση ασθενών και υπέστη αρκετές αλλαγές ώστε να γίνει ακόμη πιο χρήσιμο.

6.2 Αξιολόγηση απόδοσης του συστήματος

Στο ευφυές δικτυοκεντρικό σύστημα βιντεοεπιτήρησης που σχεδιάστηκε σε αυτή την διπλωματική εργασία λήφθηκε σοβαρά υπόψη και η απόδοση (performance) του συστήματος κατά την εκτέλεσή του. Όλες οι παραμετροποιήσεις και οι δοκιμές πραγματοποιήθηκαν σε έναν ηλεκτρονικό υπολογιστή με μονοπύρρηνο επεξεργαστή AMD Sempron 3200 χρονισμένο στα 1.6 GHz, μνήμη RAM της τάξης των 2048 MB, και σκληρό δίσκο χωρητικότητας 60 GB με λειτουργικό σύστημα Ubuntu (έκδοση 13.10 Saucy Salamander).

Σχετικά με την πτώση στην απόδοση του συστήματος που προκαλεί η χρήση των καμερών, όπως προτυπώθηκε και στην παράγραφο 2.4, η χρησιμοποίηση USB καμερών αποτελεί μειονέκτημα καθώς μια κάμερα USB καταλαμβάνει μεγάλο bandwidth ακόμα και αν χρησιμοποιηθεί χαμηλός ρυθμός frame οπότε εάν χρειαστούν περισσότερες κάμερες, θα πρέπει να προστεθούν επιπλέον USB PCI κάρτες στον υπολογιστή που εκτελεί το λογισμικό. Παρόλα αυτά, και με την παραπάνω λύση το φορτίο του επεξεργαστή (cpu load) αυξάνεται. Στην περίπτωση

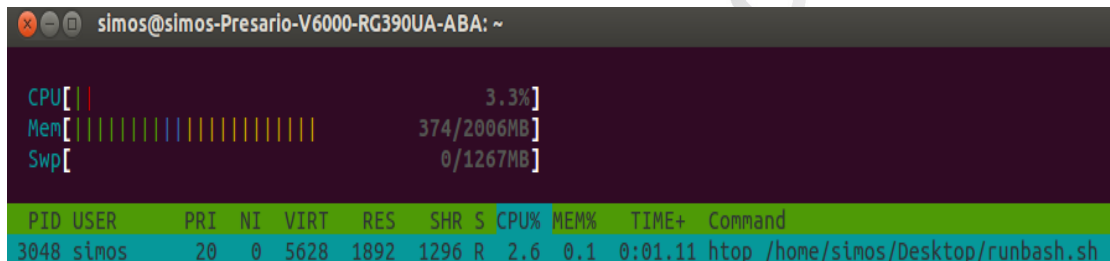
του πρώτου σεναρίου που περιγράφηκε στην παράγραφο 5.3.4, χρησιμοποιήθηκαν 4 USB κάμερες και η απόδοση του υπολογιστή που χρησιμοποιήθηκε κινήθηκε σε οριακά σημεία. Το πρόβλημα αυτό σταμάτησε να υπάρχει στην περίπτωση του δευτέρου σεναρίου όπου χρησιμοποιήθηκαν ασύρματες ip κάμερες για χωροταξικούς λόγους (πολλαπλές κάμερες σε πολλαπλά δωμάτια με μεγάλες αποστάσεις μεταξύ τους).

Όσον αφορά μεμονωμένα την απόδοση του συστήματος κατά την εκτέλεση του λογισμικού Motion σε ένα δίκτυο καμερών, είναι αναμενόμενο να επηρεάζεται ο αλγόριθμος του (motion loop) για την ανίχνευση της κίνησης όσο αυξάνεται ο αριθμός των καμερών. Βέβαια, στο σύστημα που αναπτύχθηκε στα προηγούμενα κεφάλαια λαμβάνεται υπόψη μόνο η πληροφορία της κίνησης, οπότε όλες οι κάμερες είναι ρυθμισμένες για την ελάχιστη ποιότητα εικόνας αφού αυτή δεν μας ενδιαφέρει και σαφώς δεν υπάρχουν άσκοπες καταγραφές στιγμιότυπων που θα είχαν επιπτώσεις στην δέσμευση της χωρητικότητας του σκληρού δίσκου. Αν συνυπολογιστεί το γεγονός ότι δεν υπάρχουν κάθε φορά ταυτόχρονες καταγραφές από περισσότερες από 2 κάμερες, καταλήγουμε στο συμπέρασμα ότι η απόδοση του συστήματος επηρεάζεται ελάχιστα.

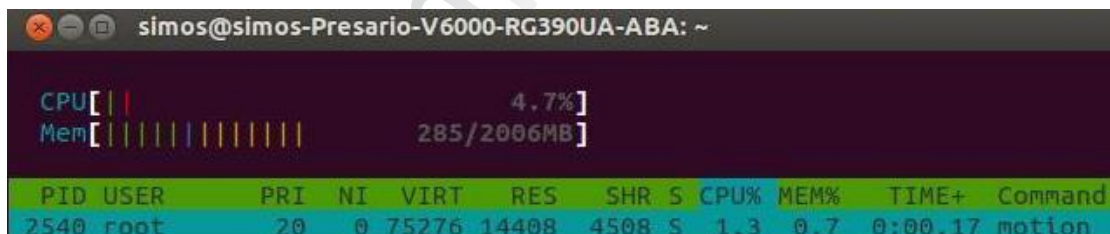
Παρόλα αυτά, το λογισμικό Motion εκτελεί τον αλγόριθμο για κάθε συνδεδεμένη κάμερα ακόμα και αν δεν υπάρχει κίνηση στο πεδίο της, επομένως όσο μεγαλώνει το δίκτυο των καμερών θα προκύπτει κάποια επίπτωση στην απόδοση, η οποία όμως είναι αρκετά μικρή έως αμελητέα. Αναλυτικότερα, σε περίπτωση σύνδεσης n αριθμού καμερών όπως προτυπώθηκε στην παράγραφο 3.1-1 θα πρέπει να δημιουργηθούν αντίστοιχα n "thread config files" και να δηλωθούν στο αρχείο motion.conf. Από εκεί και πέρα, στον βρόγχο motion (ο οποίος συνεργάζεται κατά την εκτέλεσή του με τα παραπάνω αρχεία) θα γίνει ταυτόχρονη εκτέλεση n φορές για όλα τα "thread config files" με αποτέλεσμα να προστίθεται επιβάρυνση στην απόδοση του συστήματος. Συγκεκριμένα, θα υπάρξει δέσμευση κάποιων pipes του επεξεργαστή ώστε να καταχωρήσει τις n συνδεδεμένες συσκευές (videodevice/dev/video1-n), δέσμευση n πορτών για τον http-server που θα αντιστοιχούν στην κάθε κάμερα (webcam_port xxxx), καθώς και δέσμευση χώρου στο σκληρό δίσκο για δημιουργία n ξεχωριστών φακέλων για την καταγραφή από κάθε κάμερα (target_dir /var/www/imageviewer/motion/cam1-n). Επιπρόσθετα, θα δεσμευθούν και κάποιοι ακόμα πόροι από το σύστημα κατά την προετοιμασία n φορές για τα νέα καρέ (frames) από όλες τις κάμερες, κατά την τοποθέτηση στη μνήμη buffer ανάλογες φορές του τρέχοντος καρέ για την καθεμία καθώς και κατά την ανανέωση των παραμέτρων framerate, output, locate, smartmask, noisetune (και πάλι n φορές για όλες τις κάμερες) που λαμβάνει χώρα κάθε δευτερόλεπτο. (παράγραφος 2.7).

Για να τεκμηριωθεί το γεγονός της μικρής επίπτωσης στην απόδοση του συστήματος σε περίπτωση σύνδεσης πολλαπλών καμερών, έγιναν μετρήσεις του χρόνου εκτέλεσης του βρόγχου ανίχνευσης κίνησης (motion loop) ώστε να υπάρχει μια σχετική εικόνα της απόδοσης του (cpu load, memory usage) σε διαφορετικές περιπτώσεις που είναι συνδεδεμένες μία, δύο, τέσσερις, οκτώ, δεκαέξι και τριάντα-δύο κάμερες αντίστοιχα.

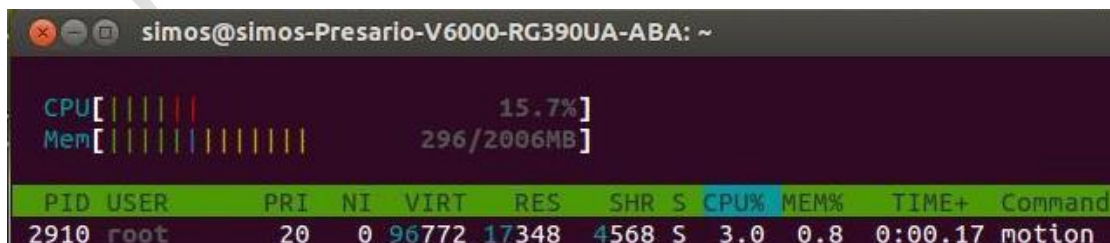
Επίσης, έγιναν μετρήσεις και αναφορικά με την επίπτωση που προκαλεί στο σύστημα η εκτέλεση του κύριου αρχείου `intelligentfeatures.php`, (μέσω του `runbash.sh`) το οποίο έτσι κι αλλιώς θα προσπελάσει και θα συγκρατήσει κάθε φορά από τη βάση δεδομένων μόνο τα χαρακτηριστικά και τις καταγραφές από δύο κάμερες ασχέτως με τον αριθμό αυτών που είναι συνδεδεμένες, χρησιμοποιώντας ελάχιστους πόρους από το σύστημα (2,6 %).



Εικόνα (6.2-1): Cpu Load και Memory Usage κατά την εκτέλεση του script `runbash.sh` που εκκινεί το κύριο αρχείο `intelligentfeatures.php`



Εικόνα (6.2-2): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου `Motion` με μια κάμερα συνδεδεμένη



Εικόνα (6.2-3): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου `Motion` με συνδεδεμένες δύο κάμερες

```

simos@simos-Presario-V6000-RG390UA-ABA: ~
CPU[||||| 17.6%]
Mem[||||| 295/2006MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2701 root 20 0 117M 20516 4568 S 3.6 1.0 0:00.91 motion

```

Εικόνα (6.2-4): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου Motion με συνδεδεμένες τέσσερις κάμερες

```

simos@simos-Presario-V6000-RG390UA-ABA: ~
CPU[|||| 10.1%]
Mem[||||| 301/2006MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2612 root 20 0 152M 25220 4568 S 5.4 1.2 0:01.07 motion

```

Εικόνα (6.2-5): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου Motion με συνδεδεμένες οκτώ κάμερες

```

simos@simos-Presario-V6000-RG390UA-ABA: ~
CPU[|||| 11.8%]
Mem[||||| 461/2006MB]
Swp[ 0/1267MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
5438 root 20 0 253M 36664 2952 S 9.2 1.8 0:02.00 motion

```

Εικόνα (6.2-6): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου Motion με συνδεδεμένες δεκαέξι κάμερες

```

simos@simos-Presario-V6000-RG390UA-ABA: ~
CPU[||||| 20.8%]
Mem[||||| 473/2006MB]
Swp[ 0/1267MB]

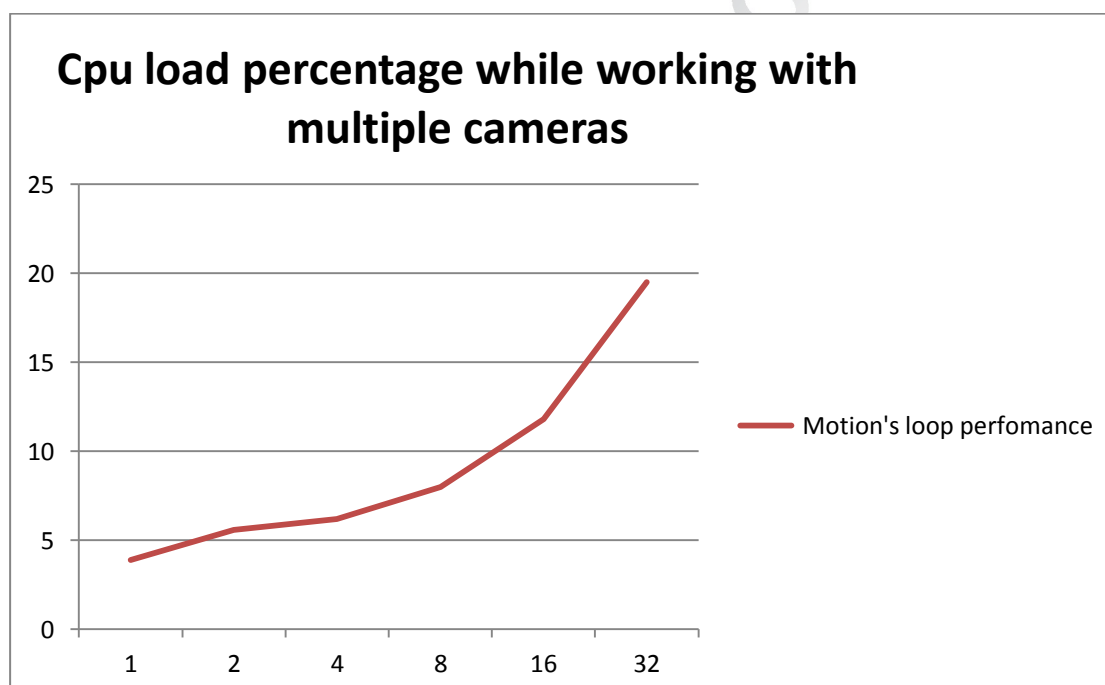
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
5167 root 20 0 423M 55776 2952 S 16.9 2.7 0:03.08 motion

```

Εικόνα (6.2-7): Cpu Load και Memory Usage κατά την εκτέλεση του αλγορίθμου Motion με συνδεδεμένες τριάντα-δυο κάμερες

Είναι προφανές ότι ακόμα και με τριάντα-δυο κάμερες συνδεδεμένες το φορτίο του επεξεργαστή για την εκτέλεση του αλγορίθμου ανίχνευσης κίνησης είναι αρκετά μικρό (16,9 %). Το γεγονός αυτό οδηγεί στο συμπέρασμα ότι το ευφύες σύστημα που αναπτύχθηκε, χρησιμοποιώντας τελικά από το λογισμικό Motion μόνο την αναγγελία της κίνησης, δεν καταναλώνει σημαντικό μέρος των πόρων του υπολογιστή που το εκτελεί καθιστώντας το ικανό να εργαστεί και σε σενάρια με απαίτηση για αρκετά περισσότερες κάμερες από αυτές που χρησιμοποιήθηκαν στις προηγούμενες παραγράφους.

Στο παρακάτω γράφημα (Εικόνα 6.2-8), απεικονίζεται η συγκεντρωτική μελέτη του ποσοστιαίου cpu load που προκαλεί η ταυτόχρονη σύνδεση πολλαπλών καμερών προσθέτοντας κάθε φορά και την εκτέλεση των αλγορίθμων που υλοποιήθηκαν στην τρέχουσα διπλωματική εργασία.



Εικόνα (6.2-8): Συγκεντρωτική μελέτη cpu load κατά την ταυτόχρονη εκτέλεση του αλγορίθμου Motion και των πρόσθετων αλγορίθμων, με συνδεδεμένες μία, δύο, τέσσερις, οκτώ, δεκαέξι και τριάντα-δυο κάμερες

6.3 Μελλοντικές επεκτάσεις-προτάσεις

Το Motion αποτελεί ένα λογισμικό ανίχνευσης κίνησης ευρείας χρήσης και με το ανάλογο configuration μπορεί να εφαρμοστεί και να αξιοποιηθεί σε πολλούς τομείς πέρα από τον συγκεκριμένο που αναφέρεται στην παρούσα διπλωματική εργασία. Βέβαια, ο τρόπος σκέψης για την υλοποίηση σε διαφορετικά σενάρια είναι κοινός και ενδεχομένως πολλά στοιχεία που αναλύονται στο τρέχον project να μη

διαφέρουν ιδιαίτερα και να μπορούν να προσαρμοστούν αντίστοιχα. Στις επόμενες παραγράφους αναφέρονται μερικούς από τους τομείς που μπορεί να χρησιμοποιηθεί το Motion.

Παραμένοντας στον τομέα της υγείας και της διευκόλυνσης ασθενών και προσωπικού, το Motion μπορεί να εφαρμοσθεί κατά τη διάρκεια ιατρικών εξετάσεων-απεικονίσεων. Συγκεκριμένα, κατά την εξέταση σε αξονικό ή μαγνητικό τομογράφο είναι πολύ σημαντικό για τη σωστή απεικόνιση ο ασθενής να παραμένει ακίνητος σε κάποια σημεία της εξέτασης. Το γεγονός αυτό, πολλές φορές δεν είναι εφικτό με αποτέλεσμα να κρίνεται αναγκαία η επανεξέταση η οποία επιβαρύνει τον ασθενή (άσκοπη έκθεση σε ακτινοβολία, σπατάλη πολύτιμου χρόνου κτλ). Έτσι, λοιπόν, αν στο λογισμικό ενός συστήματος τομογραφίας γίνει χρήση του Motion, με την κατάλληλη παραμετροποίηση αν ο ασθενής κινηθεί θα διακόπτεται ακαριαία η εξέταση.

Επιπλέον, είναι ευνόητη η χρησιμότητα του λογισμικού Motion σε συστήματα ασφαλείας. Πρόκειται ίσως για τον πιο διαδεδομένο κλάδο βιντεοεπιτήρησης με εκατοντάδες εφαρμογές και παραλλαγές αλλά παρόλα αυτά η χρήση του Motion θα μπορούσε να προσθέσει τον δικό του τόνο σε περίπτωση που είναι επιθυμητή η παρακολούθηση σε περιβάλλοντα όπου απαιτείται η διασφάλιση της ιδιωτικότητας.

Επιπρόσθετα, το σύστημα θα μπορούσε να εφαρμοστεί με επιτυχία σε συστήματα παρακολούθησης οδικής κυκλοφορίας. Με ανάλογες ρυθμίσεις σχετικά με τον αριθμό των ανιχνεύσεων και το χρόνο μεταξύ διαδοχικών ανιχνεύσεων είναι δυνατόν να προσδιοριστεί η πυκνότητα της κυκλοφορίας σε έναν αυτοκινητόδρομο, η ταχύτητα των κινούμενων οχημάτων, ακόμα και να διαπιστωθεί ένα ατύχημα. Είναι προφανής η χρησιμότητα των ειδοποιήσεων του Motion, καθώς σε ανάλογες περιπτώσεις θα μπορεί να ενημερώνει άμεσα τους αρμόδιους φορείς για όλα τα ζητήματα που αναφέρθηκαν παραπάνω.

Τέλος, για το όλο project θα ήταν δυνατόν να αναπτυχθεί μια εφαρμογή για συσκευές Android όπου θα προωθούνται οι συναγερμοί που εξάγει το σύστημα ώστε ο τελικός χρήστης να διαχειρίζεται τα γεγονότα που έχουν συμβεί καθώς και να προβαίνει στην παραμετροποίηση του.

Εν κατακλείδι, με την ανάλογη παραμετροποίηση το Motion με τις προτάσεις που αναπτύχθηκαν μπορεί να έχει εφαρμογή σε πάρα πολλούς τομείς και να αξιοποιηθεί και να προσαρμοστεί από διαφορετικούς επαγγελματίες ή ακόμα και ερασιτέχνες ανάλογα με το αντικείμενο ασχολίας τους. Σίγουρα, η τρέχουσα διπλωματική εργασία γεννά αρκετές ιδέες για τη χρήση του Motion και σε διαφορετικούς κλάδους, πέρα από αυτόν της υγείας που εξετάστηκε εκτενώς σε αυτήν και οι περισσότερες παραμετροποιήσεις, αν όχι όλες, που έλαβαν χώρα μπορούν να προσαρμοστούν ανάλογα και σε μελλοντικά projects.

BIBΛΙΟΓΡΑΦΙΑ ΚΑΙ ΧΡΗΣΙΜΑ INTERNET SITES

1. *Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering* By George Yang, Springer Publishing Company Incorporated, 2012
2. *Recent Advances in Computer Science and Information Engineering: Volume 5 (Springer)* By Zhihong Qian, Lei Cao, Weilian Su, Tingkai Wang, Huamin Yang, Springer-Verlag Berlin and Heidelberg GmbH & Co. K, Engelska, 2012
3. *Computational Methods in Neural Modeling: 7th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2003* Mao, Menorca, Spain, June 2003 *Proceedings, Part 1*
4. *Robust Motion Detection in Real-Life Scenarios* By Ester Martínez-Martín, Ángel P. del Pobil, Springer Publishing Company Incorporated, 2012
5. Ευφρές σύστημα βιντεοεπιτήρησης με δυνατότητα ανίχνευσης απλών μοτίβων κίνησης και διασφάλισης της ιδιωτικότητας. ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ. Κωνσταντίνος Μπογιατζάκης
6. Εξαγωγή κινούμενου αντικειμένου σε βίντεο. ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ. Σταμούλη Μαριάννα
7. Σχεδίαση και υλοποίηση συστήματος ελέγχου πρόσβασης σε βάσεις προσωπικών δεδομένων με χρήση πλαισίου πολιτικής (policy framework). ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ. Δημήτριος Ραπτοδήμος

8. *Context Aware Privacy in Visual Surveillance* By Simon Moncrieff, Svetha Venkatesh, Geoff West, *Pattern Recognition, ICPR 2008 19th International Conference, Tampa, FL*

9. *A new method of networked video surveillance with privacy area protection* By Jiang Yibo, Yao Xinwei, Wang Wanliang, *Broadband Network & Multimedia Technology, IC-BNMT '09 2nd IEEE International Conference*, pp. 122-126, Beijing

10. *Privacy Enablement in a surveillance system* By Andrew Senior, *Image Processing, ICIP 2008 15th IEEE International Conference*, pp. 1680-1683, San Diego, CA

11. *Privacy Protection Technology in Video Surveillance System* By Hae-Min Moon, Chang Ho Seo, Yongwha Chung, Sung Bum Pan, *Embedded and Multimedia Computing, EM-Com 2009. 4th International Conference*, Jeju

12. *A Study on the Human Identification Technique for Privacy Protection in Intelligent Video Surveillance System* By Min-Gu Kim, Hae-Min Moon, Chulho Won, Yongwha Chung, Sung Bum Pan, *2011 IEEE Asia - Pacific Services Computing Conference*

13. *Evaluation of privacy filters impact on video surveillance intelligibility* By P. Korshunov, C. Araimo, F. De Simone, C. Velardo, J.-L. Dugelay, and T. Ebrahimi, *Quality of Multimedia Experience (QoMEX), Fourth International Workshop on*, pp. 150-151, 2012

14. *Technologies for Granting Balance between Security and Privacy in Video-Surveillance* By Iglesias, R.B. ; Cichowski, J. , Dalka, P. , Ellwart, D. , Pedagadi, S. , Orwell, J., Kroener, I. , Neyland, D., *Intelligence and Security Informatics Conference (EISIC), 2012 European*, pp.278-283, 22-24 Aug.

15. *Embedded Intelligent Video Surveillance and Cooperative Tracking System* By Fei Zhang, Pinyi Ren, Hao Chen and Guobing Li, *2012 7th International ICST*

Conference on Communications and Networking in China (CHINACOM) pp. 632-637, Kun Ming, China

16. *Development of a Low-Cost Fall Intervention System for Hospitalized Dementia Patients* By *Osamah Rawashdeh, Waseem Sa'deh, Muawea Rawashdeh, and Guangzhi Qu, Marisa Ferrari and Barbara Harrison, Robert Hammond and Michael Maddens, IEEE International Conference on Electro/Information Technology, May 6-8, 2012 IUPUI, Indiana, USA.*

17. *Monitoring patients via a secure and mobile healthcare system* By *Yonglin Ren, Richard Werner, Nelem Pazzi, Azzedine Boukerche, IEEE Wireless Communications, pp. 59-65, Feb 2010*

18. *Enabling secure and privacy-aware mobile sensing and e-health applications on everybody's smartphone* By *Steffen Ortmann, Michael Maaser, IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), September 6 2011*

URL's

- <http://www.lavrsen.dk/foswiki/bin/view/Motion/MotionGuide3x1x20>
- http://www.mit.edu/activities/anime/arch/i386_rhel4/share/doc/motion-3.2.8/motion_guide.html
- <http://roussell.net/blog/using-motion-on-ubuntu-10-04/>
- <http://www.chriswpage.com/2009/05/setup-an-advanced-webcam-security-system-with-ubuntu-8-04-and-motion/>
- <http://linux.die.net/man/1/motion>
- <http://emit.demon.co.uk/motion/>
- <http://techspalace.blogspot.gr/2012/02/send-sms-using-gammu-command-line-tool.html>
- <http://ebooks.edu.gr/modules/ebook/show.php/DSB103/173/1206,4409/>
- <http://www.thegeekstuff.com/2009/06/15-practical-crontab-examples/>