

Sommario

0 Introduzione.....	3
0.1 Cos'è IntruderVISION	3
0.2 What is IntruderVISION?	3
1 Specifications	4
1.1 Technical	4
1.2 Functionalities	4
1.3 Minimum tested hardware requirements	4
2 Architettura del sistema.....	5
3 Interfaccia utente.....	6
3.1 Schermata del Menu principale	6
3.2 Schermata di Calibrazione	7
3.3 Schermata delle Impostazioni	8
3.5 Schermata di Monitoraggio	9
4 Architettura software ed algoritmi utilizzati.....	10
4.1 Architettura.....	10
4.2 Stato del Menu iniziale	11
4.3 Stato delle Impostazioni	11
4.4 Stato di Calibrazione	12
4.5 Stato di Monitoraggio	14
4.5.1 Rilevazione del movimento.....	16
5 Implementazione software	17
5.1 Ambiente di sviluppo	17
5.2 Librerie esterne	17
5.3 Listato.....	17
6 Conclusioni	18
6.1 Funzionalità aggiuntive e migliorie possibili	18
7 Appendice	19
7.1 Motion detection	19
7.1.1 Basic models.....	19
7.1.2 Parametric models	19
7.1.3 Non – parametric models	19
7.1.4 Other methods.....	19
7.1.5 Chosen model	19
7.2 Foglio di calibrazione	20
7.3 Fonti e riferimenti	21

0 Introduzione

0.1 Cos'è IntruderVISION

IntruderVISION è un programma per la sicurezza domestica che permette di utilizzare qualsiasi computer a cui sia collegata una webcam come una security camera.

Il software sfrutta infatti il flusso di immagini provenienti da una qualsiasi webcam collegata al computer per rilevare movimenti ed intrusioni, per poi segnalarli all'utente tramite e-mail.

Oltre a segnalare l'intrusione, il software permette di salvare dei filmati relativi alle intrusioni, in modo tale che sia possibile rivedere la scena e ricavare più informazioni possibili riguardo l'intrusione.

Inoltre, se il sistema è stato prima adeguatamente calibrato, IntruderVISION è in grado di calcolare le dimensioni in cm^2 di qualsiasi oggetto in movimento che sia stato rilevato e che sia più grande di un quadrato di 2 cm di lato.

0.2 What is IntruderVISION?

IntruderVISION is a software aimed at domestic security which allows to use any personal computer with an attached webcam to serve as a security camera.

In fact, it makes use of the stream of images coming from any webcam connected to the PC to detect movements and intrusions, thus then warning the user about them via e-mail.

Other than that, the software also records a video of the intrusion, so that it is possible to watch again the scene and extract more information about the intrusion.

Furthermore, if the system have been previously calibrated, IntruderVISION is able to calculate the dimensions in cm^2 of any moving object that is detected and that is bigger than 2 x 2 cm.

1 Specifications

1.1 Technical

Note: The technical specifications are to be considered relative to a situation where the webcam is placed at a 90° angle to the controlled area, at a distance of about 50 cm from it.

- Acquisition resolution: 640x480 pixel
- Minimum luminosity of the surfaces: 11 lx
- Minimum dimensions of the moving object: 2 x 2 cm
- Speed limits of the moving objects
 - Minimum: 0.5 cm/s
 - Maximum: 1.5 m/s

1.2 Functionalities

- Movement detection in the controlled area
- Documentation of the intrusion through a recording

The following feature is usable and guaranteed only if the webcam is placed in a 90° angle to the controlled surface, at a distance from 50 cm to 1 m.

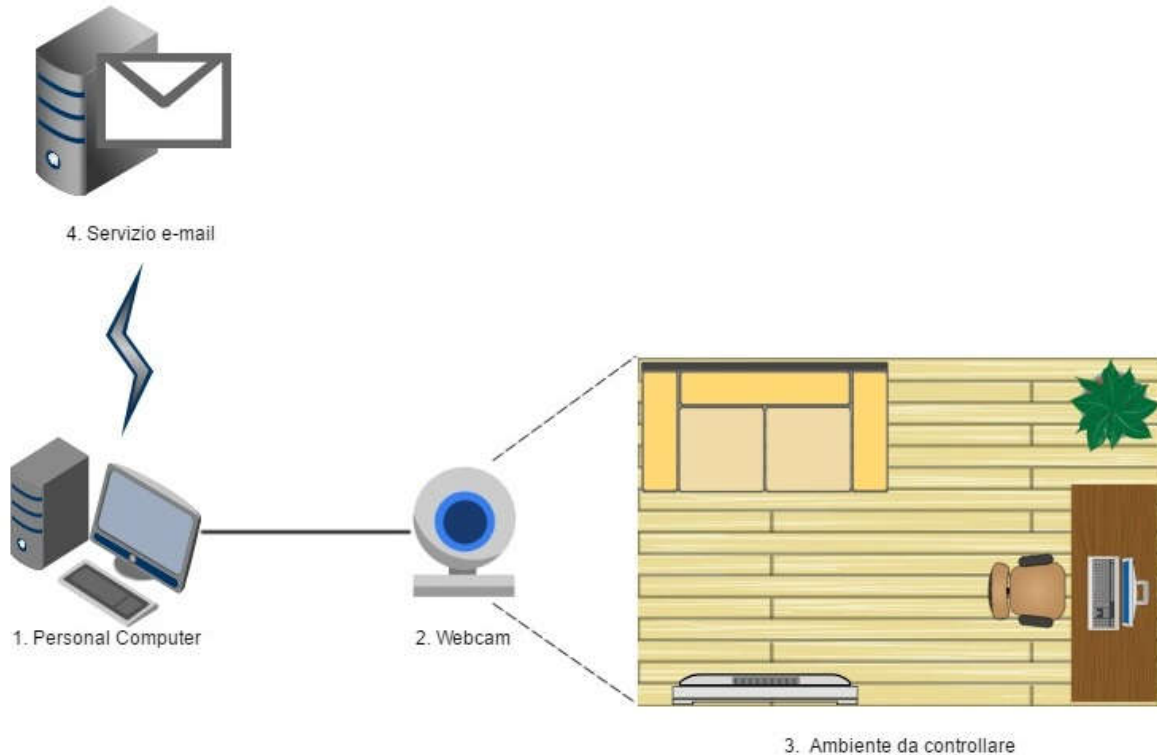
- Calculation of the dimensions of the moving object

1.3 Minimum tested hardware requirements

- Writing speed Hard Disk: 100 MB/s
- CPU: Intel i7 3632QM Quad core @ 2.2GHz o AMD FX3650 Six core @ 3.9GHz
- OS: Windows 8.1 or higher
- OpenCV (compiled with Visual Studio 2015) and OpenSSL libraries installed

2 Architettura del sistema

L'architettura del sistema viene descritta dallo schema sottostante.



1. **Personal computer:** si occupa di acquisire le immagini provenienti dalla webcam e le processa, individuando eventuali movimenti e segnalandoli tramite e-mail.
2. **Webcam:** viene utilizzata per analizzare l'ambiente da controllare. La sua posizione ottimale è direttamente sopra all'area da controllare (perpendicolarmente), in modo da avere una migliore copertura di tutta la zona (e supportare il calcolo delle dimensioni degli oggetti in movimento).
3. **Ambiente da controllare:** l'area che si vuole controllare. Si ipotizza che abbia un'illuminazione sufficiente all'individuazione dei movimenti (ogni superficie deve avere una illuminazione di almeno 15 lux).
4. **E-mail:** quando viene individuato un movimento, il software provvede ad inviare una mail di segnalazione all'utente.

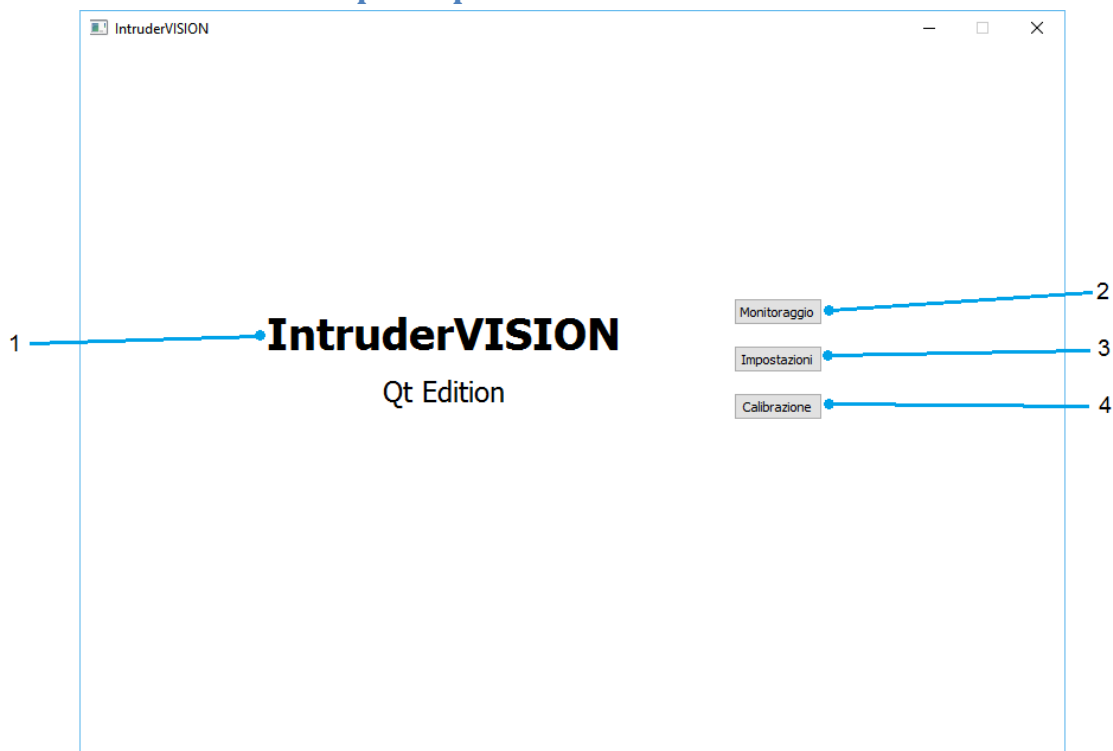
3 Interfaccia utente

Per l'interfaccia si è deciso di utilizzare lo stile di default degli applicativi Windows, in modo tale che l'utente sia in grado di riconoscere elementi già noti, e sia pertanto in grado di navigare l'interfaccia nel minore tempo possibile.

Vengono di seguito descritte le schermate principali del software, ovvero:

- Schermata del **Menu principale**, da cui è possibile selezionare se avviare il monitoraggio, accedere alle impostazioni o calibrare il sistema.
- Schermata di **Calibrazione**, da cui è possibile calibrare il sistema per individuare il rapporto tra dimensioni in pixel ed in centimetri.
- Schermata delle **Impostazioni**, che permette di accedere ai controlli relativi al funzionamento delle notifiche, del salvataggio dei filmati e della webcam.
- Schermata di **Monitoraggio**, che consente di monitorare la zona inquadrata della webcam e di essere sempre a conoscenza di intrusioni in corso ed intrusioni passate.

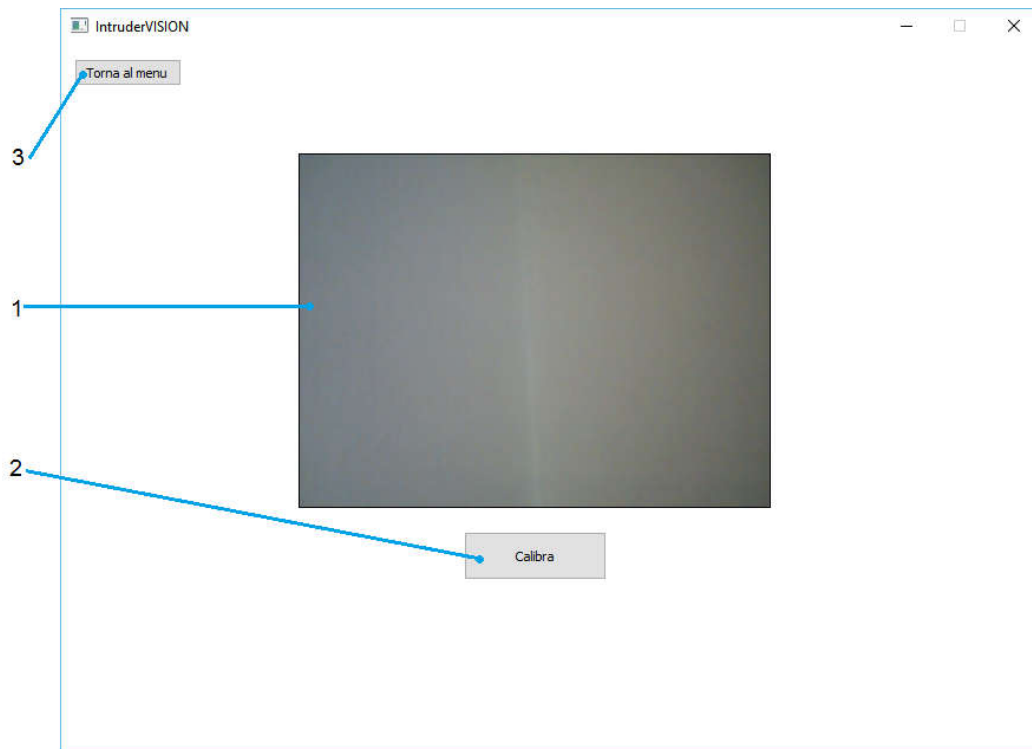
3.1 Schermata del Menu principale



Il menu principale è il più minimale possibile, ed è composto da solo quattro elementi:

1. Il **nome del software**, posizionato al centro della schermata in modo che sia la prima cosa sui cui l'utente concentra la propria attenzione.
2. Il pulsante **Monitoraggio**, attraverso il quale la schermata cambia in quella di monitoraggio e rilevazione dei movimenti.
3. Il pulsante **Impostazioni**, attraverso il quale si accede ai controlli di configurazione del monitoraggio.
4. Il pulsante **Calibrazione**, che permette di accedere alla schermata di calibrazione del sistema.

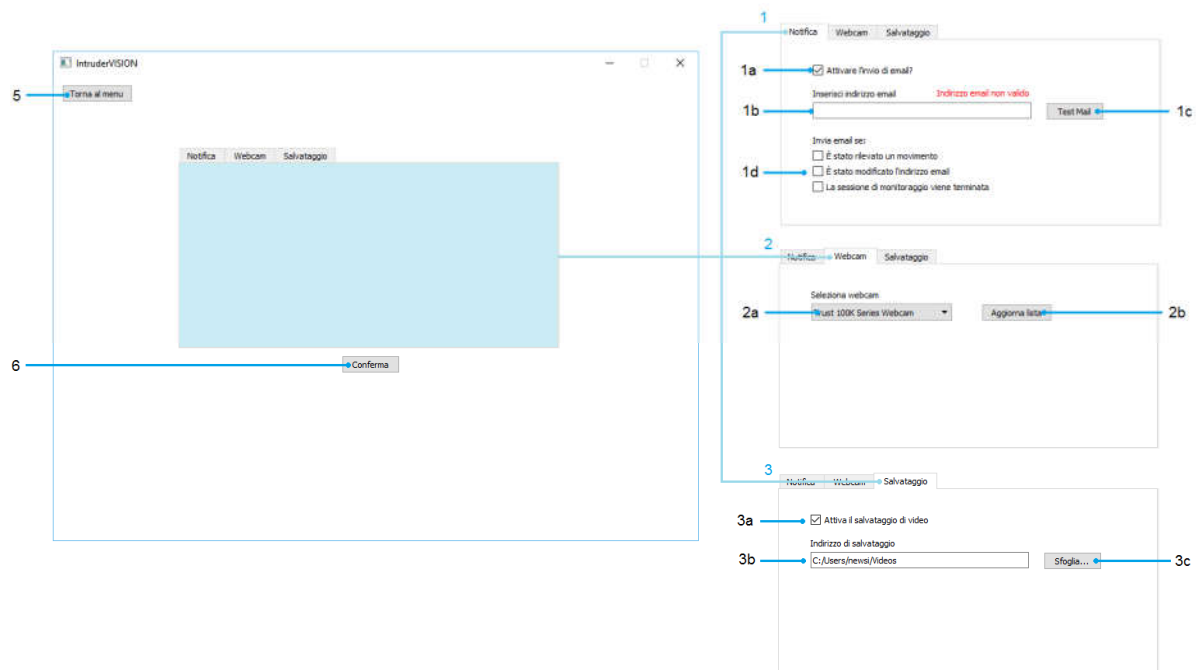
3.2 Schermata di Calibrazione



La schermata di calibrazione è composta da:

1. **Preview webcam**, utilizzata per mostrare all'utente in real time quello che la webcam sta inquadrando, in modo tale da poter posizionare correttamente il foglio di calibrazione.
2. Pulsante **Calibra**, attraverso il quale si dà conferma di aver posizionato correttamente il foglio di calibrazione.
3. Pulsante **Torna al menu**, utilizzato per tornare al menu principale.

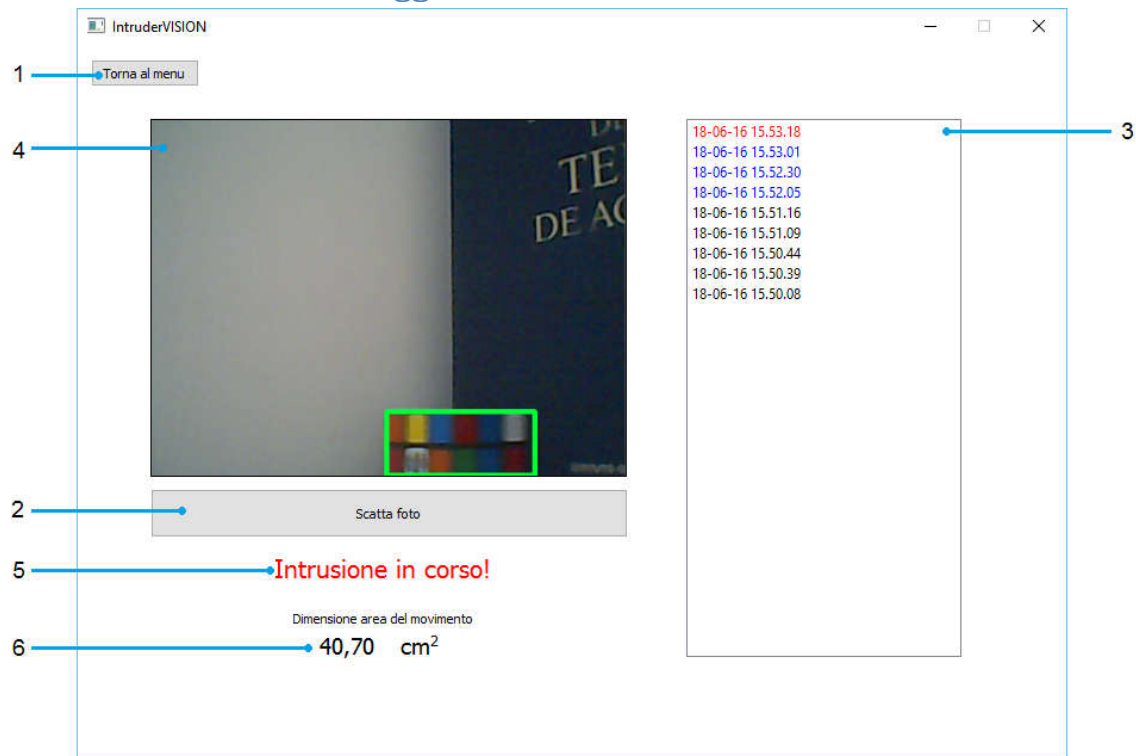
3.3 Schermata delle Impostazioni



La schermata delle impostazioni è divisa in:

1. **Tab delle notifiche**, che contiene
 - a. **Checkbox di attivazione delle e-mail**, con cui si attivano e disattivano le funzioni di invio delle e-mail. Quando disattivate, tutti gli altri controlli della schermata diventano invisibili.
 - b. **Input di testo** per fare inserire all'utente l'indirizzo a cui si vogliono ricevere le e-mail.
 - c. Pulsante **Test Mail**, utilizzabile per testare il funzionamento del sistema ed il corretto inserimento dell'indirizzo e-mail.
 - d. **Checkbox per la selezione dei casi di invio** delle e-mail. È possibile selezionare la rilevazione di un movimento, la modifica dell'indirizzo e-mail e la terminazione di una sessione di monitoraggio.
2. **Tab della webcam**, che contiene
 - a. **Menu a tendina** per la selezione della webcam che si desidera utilizzare.
 - b. Pulsante **Aggiorna lista**, utilizzabile per aggiornare la lista delle webcam selezionabili.
3. **Tab del salvataggio**
 - a. **Checkbox di attivazione del salvataggio**, con cui si attivano e disattivano le funzioni di salvataggio dei filmati delle intrusioni. Quando disattivate, tutti gli altri controlli della schermata diventano invisibili.
 - b. **Indicatore testuale** per mostrare all'utente la cartella attualmente impostata.
 - c. Pulsante **Sfoglia...**, quando premuto apre una finestra che permette all'utente di selezionare una cartella dove salvare i filmati.
4. Pulsante **Conferma**, quando premuto salva le impostazioni e torna al menu principale.
5. Pulsante **Torna al menu**, se premuto annulla le modifiche apportate alle impostazioni e torna al menu principale.

3.5 Schermata di Monitoraggio



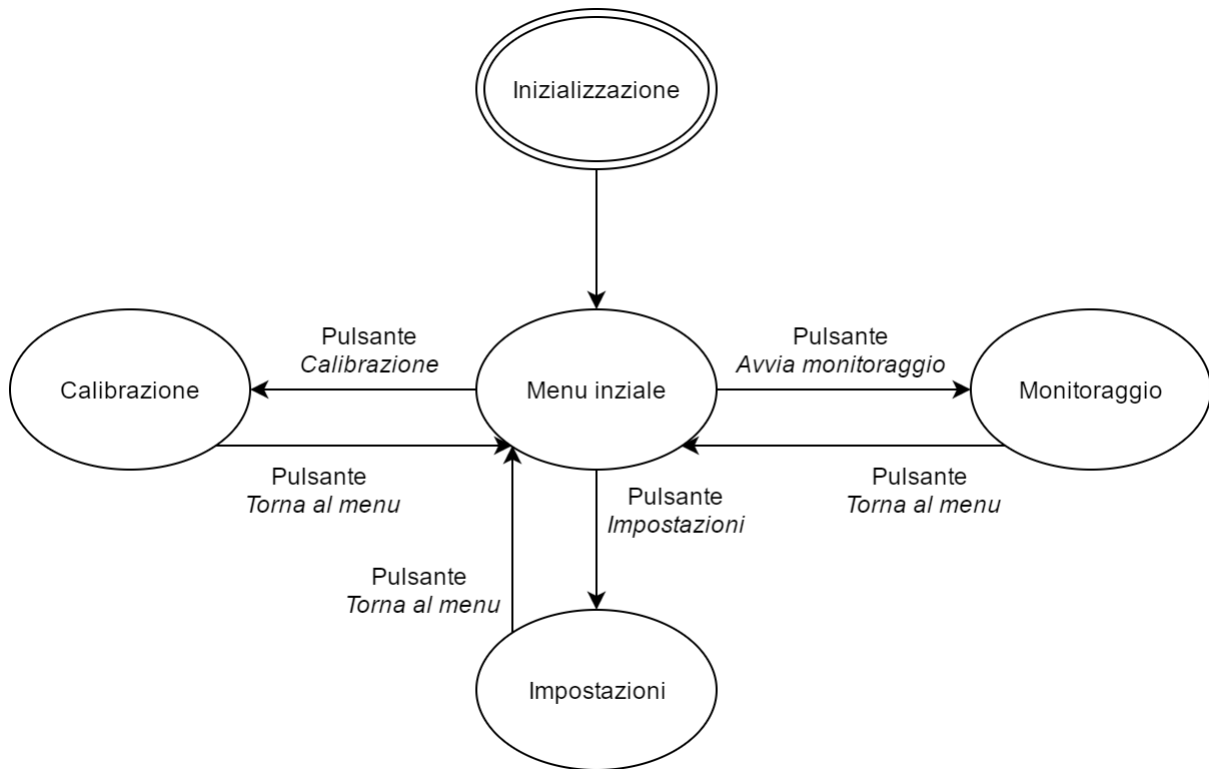
La schermata di monitoraggio è composta dai seguenti elementi:

1. Pulsante **Torna al menu**, se premuto permette di tornare al menu principale.
2. Pulsante **Scatta una foto**, se premuto scatta una foto e la salva nella cartella di default *Immagini* di sistema.
3. **Lista delle intrusioni**, viene aggiornata ad ogni nuova intrusione. Si distinguono tre casi:
 - a. L'**elemento è nero**, vuol dire che non è disponibile un filmato che documenti quell'intrusione.
 - b. L'**elemento è rosso**, vuol dire che il filmato sta ancora venendo salvato e vi si può quindi ancora accedere per guardarlo.
 - c. L'**elemento è blu**, vuol dire che il filmato è disponibile ed è sufficiente cliccare due volte sull'elemento per visualizzarlo.
4. **Preview webcam**, consente all'utente di controllare personalmente l'area inquadrata dalla webcam. In caso di intrusioni rilevate, la zona in movimento viene riquadrata in verde.
5. **Segnalatore di intrusione**, un indicatore testuale rosso che segnala quando viene rilevata una intrusione.
6. **Indicatore dimensione oggetto in movimento**, che mostra l'area in cm^2 del riquadro relativo al movimento. Questo valore è disponibile solo se il software è stato precedentemente calibrato.

4 Architettura software ed algoritmi utilizzati

4.1 Architettura

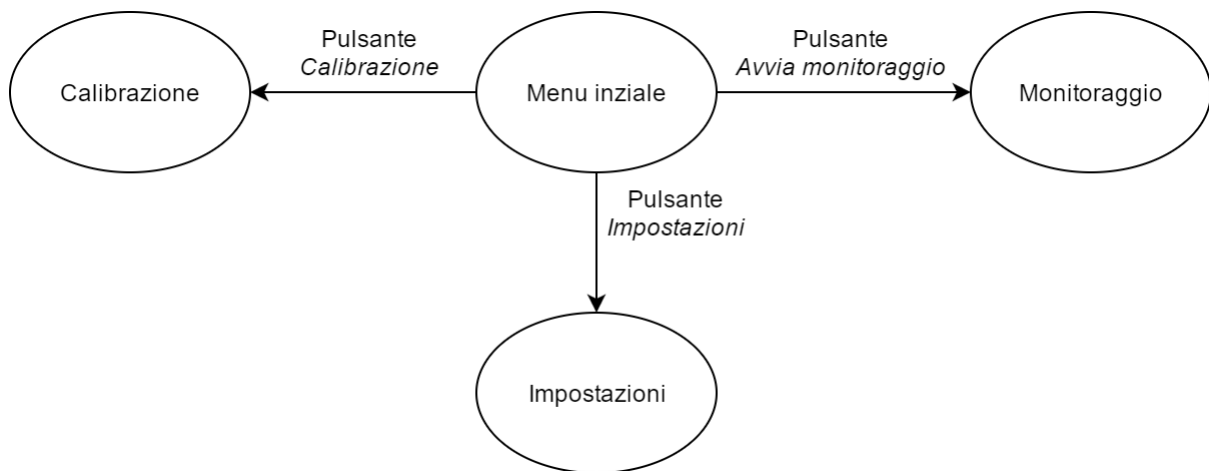
Il sistema è descritto da un automa con il seguente diagramma degli stati:



- **Menu iniziale:** Schermata dalla quale è possibile selezionare lo stato successivo, a scelta tra monitoraggio, calibrazione ed impostazioni.
- **Impostazioni:** Schermata in cui l'utente può selezionare ed impostare le diverse impostazioni del software.
- **Calibrazione:** Schermata in cui, se necessario, è possibile calibrare tramite un preciso *pattern di calibrazione* il rapporto tra dimensioni in pixel ed in centimetri di un oggetto.
- **Monitoraggio:** Schermata in cui è possibile osservare quanto sta venendo rilevato dalla fotocamera, e da cui è possibile rivedere i filmati relativi ad intrusioni terminate.

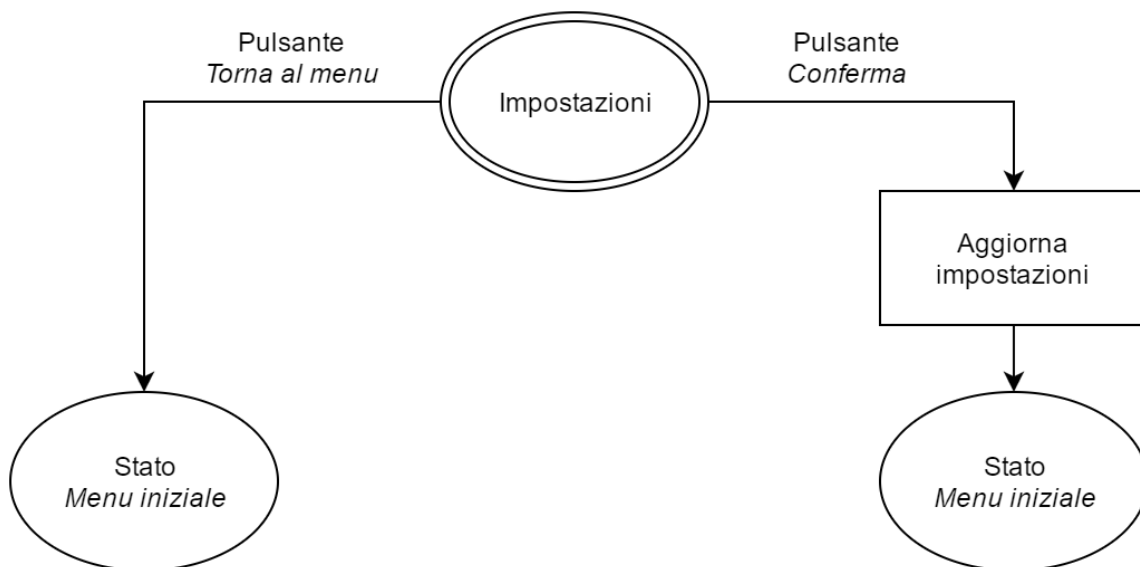
4.2 Stato del Menu iniziale

Il **menu iniziale** permette all'utente di selezionare lo stato successivo attraverso dei pulsanti



4.3 Stato delle Impostazioni

Nello **stato delle impostazioni** l'utente può accedere ai controlli per la gestione delle notifiche via e-mail, per la selezione della webcam da utilizzare e per l'attivazione e gestione del salvataggio dei filmati di documentazione delle intrusioni.



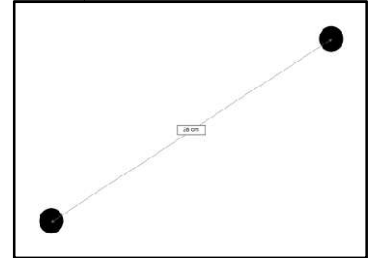
Al termine della modifica delle impostazioni, le si può salvare ed applicare premendo "Conferma", mentre tornando al menu tutte le modifiche apportate vengono annullate.

4.4 Stato di Calibrazione

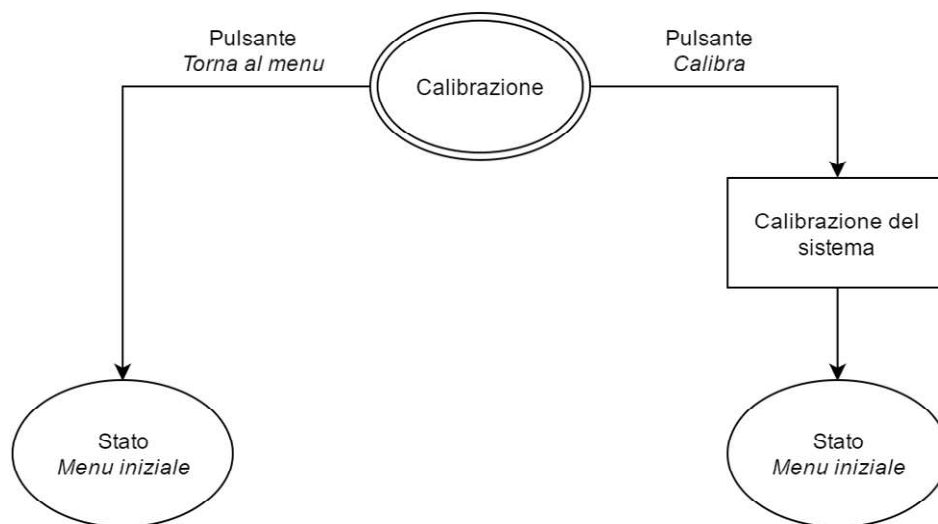
Lo **stato di calibrazione** si occupa di calcolare il rapporto tra una dimensione in pixel e la stessa dimensione in centimetri. Questo risultato è raggiunto utilizzando un foglio di calibrazione (*vedi appendice*) da porre nella zona visibile della webcam, per poi premere il pulsante "Calibra".

Il foglio di calibrazione (*immagine a fianco*, fare riferimento a pag. 19) consiste in due cerchi posti ad una distanza fissa e conosciuta (28 cm).

Questo permette al software, dopo aver individuato i due cerchi e calcolato la distanza in pixel, di ottenere il rapporto effettivo tra pixel e centimetri.

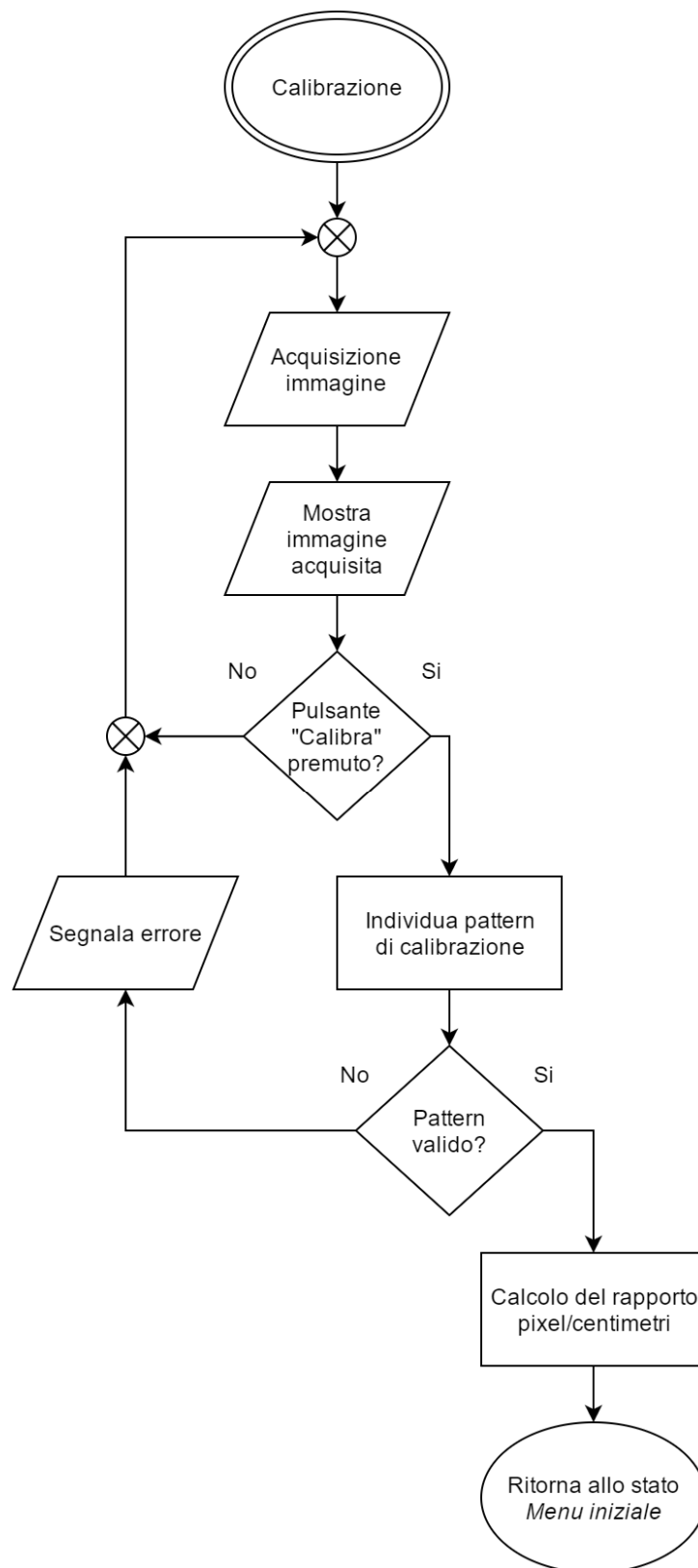


Di seguito il funzionamento generale dello stato:



Viene quindi descritto l'**algoritmo di calibrazione**:

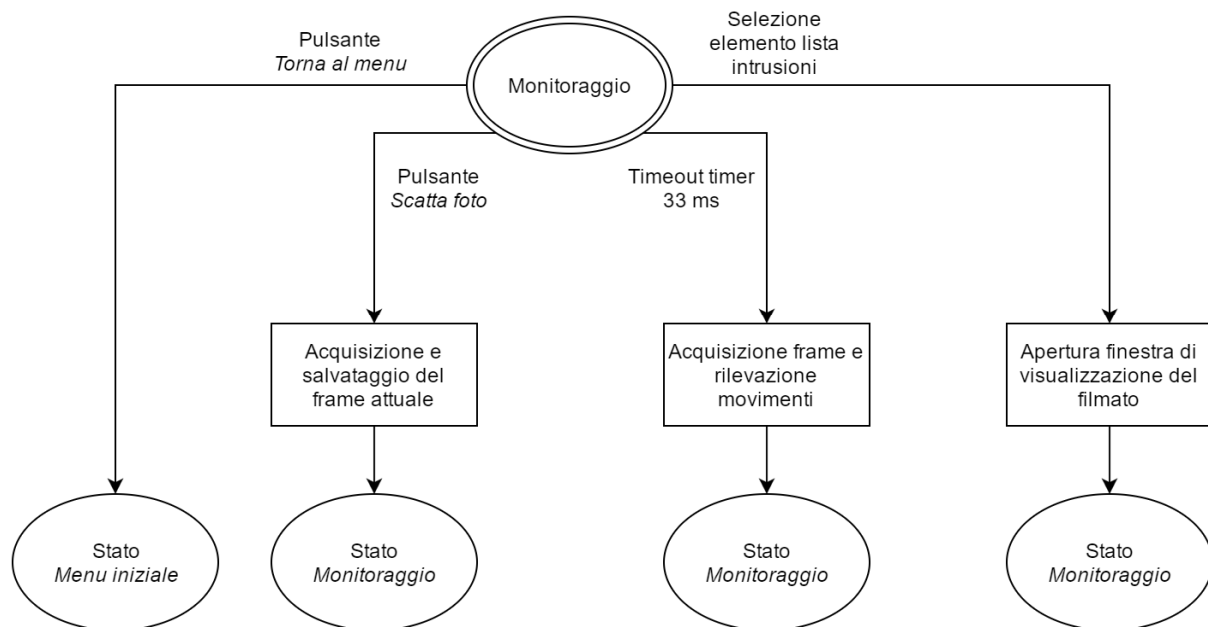
1. Viene acquistata l'immagine (frame) dalla webcam;
2. L'immagine ottenuta viene mostrata all'utente (in modo che possa posizionare correttamente il foglio di calibrazione);
3. Si verifica se il pulsante per calibrare è stato premuto:
 - a. Se **non è stato premuto**, si continua ad acquisire e controllare fino a che l'utente non preme il pulsante;
 - b. Se **è stato premuto**, si individua il pattern di calibrazione (i due cerchi) dall'immagine acquisita;
4. Si verifica se il pattern individuato è valido (es. ci sono 3 cerchi, o nessuno):
 - a. Se **non è valido**, si segnala l'errore e si continua ad acquisire;
 - b. Se **è valido**, si calcola la distanza tra i due cerchi in pixel e la si divide per la distanza conosciuta in centimetri;
5. Si ritorna allo stato *Menu iniziale*.

Diagramma a blocchi dell'algoritmo:

4.5 Stato di Monitoraggio

Nello **stato di monitoraggio** le immagini provenienti dalla webcam sono utilizzate per rilevare la presenza di movimenti attraverso il metodo della *differenza con lo sfondo* (vedi capitolo 7.1 *Motion detection*).

Il funzionamento complessivo dello stato è descritto dal seguente schema:

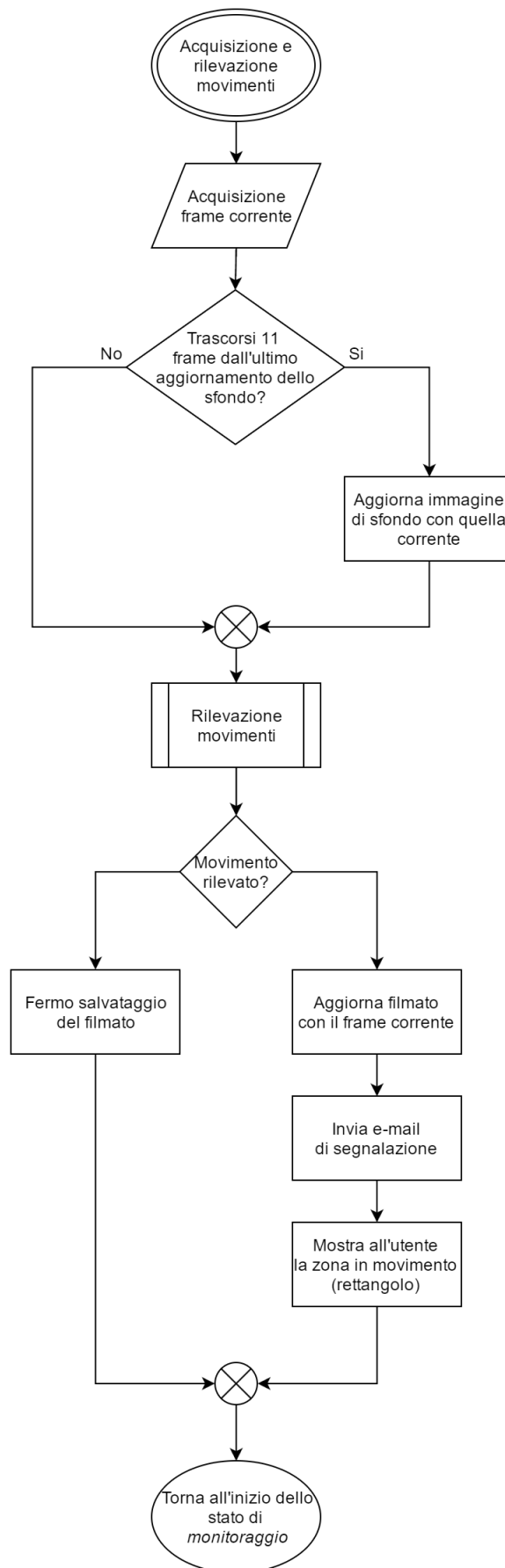


Le immagini sono quindi acquisite e processate una volta ogni 33 ms (ottenendo così un massimo di 30 FPS in acquisizione). In questa fase si procede anche ad eventualmente salvare dei filmati e ad inviare delle e-mail di notifica.

Se il salvataggio dei filmati è attivo, è possibile avviare la loro riproduzione selezionando il relativo elemento dalla lista delle intrusioni (contraddistinto da ora e data).

Viene di seguito descritto l'**algoritmo** che descrive l'operazione di **acquisizione e rilevazione**:

1. Acquisizione del frame corrente dalla webcam;
2. Aggiornamento della lista degli ultimi 10 frame acquisiti;
3. Verifica che il numero di frame trascorsi dall'ultimo aggiornamento dello sfondo sia divisibile per 11:
 - a. Se **si**, significa che sono passati 11 frame da quel momento, e si procede aggiornando l'immagine di sfondo a quella appena acquisita;
 - b. Se **no**, si procede normalmente;
4. Verifico la presenza di un movimento confrontando l'immagine appena acquisita e quella di sfondo:
 - a. Se **non c'è movimento**, fermo l'eventuale salvataggio del filmato;
 - b. Se **c'è movimento**, salva l'immagine attuale nel filmato, invia una mail di segnalazione e mostra all'utente l'area in movimento.

Diagramma a blocchi dell'algoritmo di acquisizione e rilevazione:

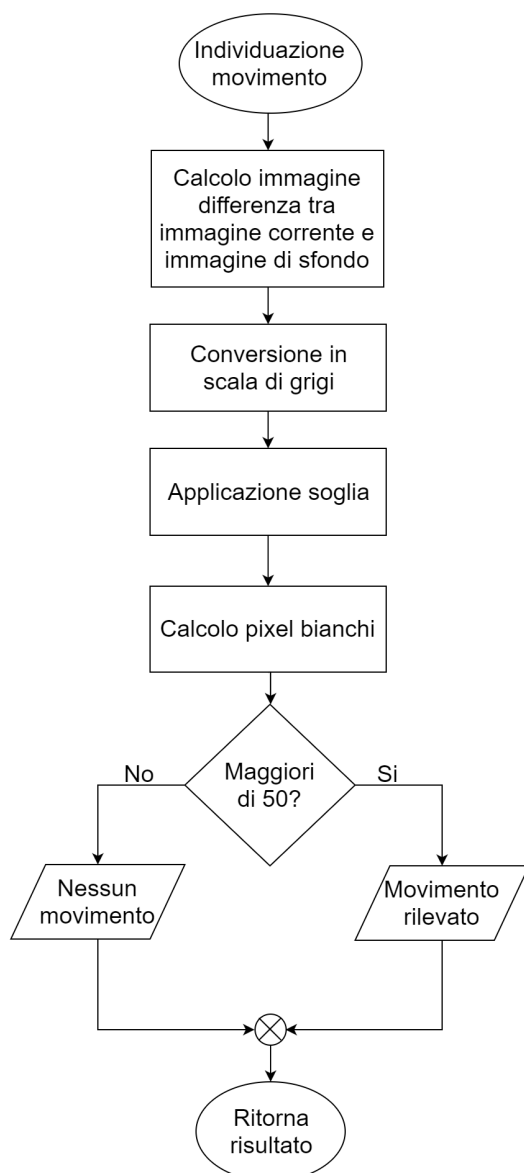
4.5.1 Rilevazione del movimento

Più nello specifico, l'**algoritmo di rilevazione del movimento** è il seguente:

1. Calcolo dell'immagine differenza tra il frame corrente e quello di sfondo;
2. Conversione in bianco e nero dell'immagine differenza ottenuta;
3. Applicazione di una soglia per ottenere un'immagine con pixel solo bianchi (ovvero i pixel in cui le due immagini sono diverse) e pixel neri (in cui le due immagini erano uguali);
4. Calcolo dei pixel bianchi;
5. Si verifica se i pixel bianchi sono più di 50:
 - a. Se sono più di 50, è presente un movimento nella zona controllata;
 - b. Se sono meno di 50, non è presente alcun movimento effettivo;
6. Il risultato ottenuto viene ritornato alla funzione di acquisizione e rilevazione.

Maggiori informazioni sui metodi di rilevazione dei movimenti sono disponibili nell'appendice, al capitolo *Motion detection*.

Diagramma a blocchi dell'algoritmo di rilevazione dei movimenti:



Esempio:

Immagine di sfondo:



Immagine acquisita:



Immagine differenza con soglia applicata:

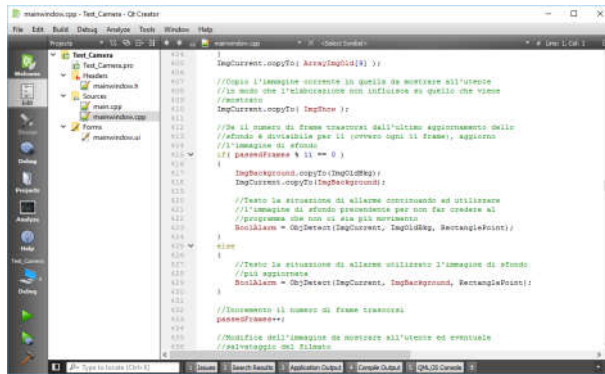


5 Implementazione software

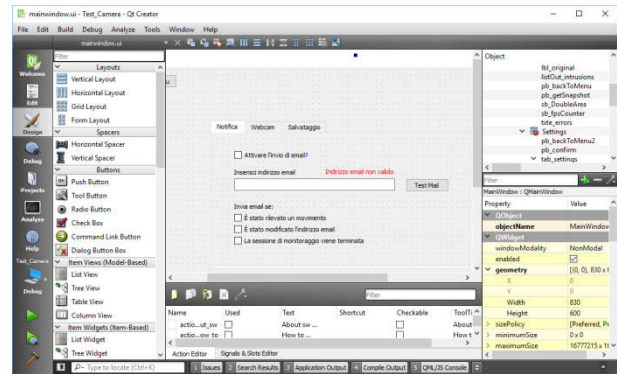
5.1 Ambiente di sviluppo

Lo sviluppo del software è avvenuto in C++ attraverso le librerie dell'ambiente Qt Open Source.

Questo ha permesso di produrre un'applicazione windows senza la necessità di acquistare licenze, ottenendo una buona ottimizzazione delle prestazioni ed un buon supporto per lo sviluppo multiplatforma.



Editor di codice



Editor interfaccia utente

5.2 Librerie esterne

Oltre a quelle fornite in Qt, sono state utilizzate anche le seguenti librerie Open Source:

- **OpenCV**, che contiene funzioni atte alla gestione ed elaborazione delle immagini, attraverso la quale sono stati implementati tutti gli algoritmi relativi ai movimenti.
- **OpenSSL**, che ha permesso l'implementazione della funzione di invio delle e-mail.
- **SMTP Client for Qt**, anch'essa necessaria per l'implementazione dell'invio delle e-mail.

5.3 Listato

Il listato è consultabile nel plico allegato, suddiviso nei tre file che compongono il codice:

- **mainwindow.h**: al suo interno vengono dichiarati tutti i metodi e gli argomenti della classe *mainwindow*.
- **mainwindow.cpp**: in questo file vengono implementati tutti i metodi della classe *mainwindow*, attraverso cui viene gestito l'intero software.
- **main.cpp**: in questo file viene chiamata la classe *mainwindow*.

6 Conclusioni

Il software ottenuto è in grado di svolgere tutte le funzioni che erano state previste.

Ad ogni modo, sono state individuate delle funzionalità aggiuntive che, avendo a disposizione una maggiore quantità di tempo, permetterebbero di ottenere una migliore esperienza per l'utente.

6.1 Funzionalità aggiuntive e migliorie possibili

- Utilizzo di protocolli di sicurezza e codifica dei dati sensibili (es. indirizzo e-mail)
- Programmazione temporale dell'inizio del monitoraggio
- Supporto per sistemi embedded (es. Raspberry Pi)
- Comunicazione in locale con altri dispositivi (es. sistema di allarme)
- Accesso al programma tramite browser in rete locale o in remoto

7 Appendice

7.1 Motion detection

There are different ways to detect a movement by analyzing a stream of images (a video), but there are three main models that are used:

- Basic models
- Parametric models
- Non – parametric models

7.1.1 Basic models

In these models the motion is detected by comparing the current acquired frame to the previous one and checking for the pixel that changed from one image to another.

In some variants, the comparison is not with the previous one, but with a *background image* set at the beginning of the session.

Pros	Cons
Fast to process	Cannot find stationary objects or slow movements
Easy to implement	

7.1.2 Parametric models

Instead of studying two images, this method calculates the mean value of the pixels and the standard deviation. Then every pixel's color probability is calculated, and if it is below a certain threshold, it is considered a part of the movement area.

Pros	Cons
Higher robustness to noise	Longer processing time
Less sensitive to little background movements	

7.1.3 Non – parametric models

Firstly, a model of the background image is created, based on the previous frames. The current frame's pixels are then compared to their probability function, and if the probability of being part of the background are lower than a certain threshold, it is considered a part of the moving object.

Pros	Cons
More sensitive than the basic models	Needs a lot of processing power and memory
Adapts easily to background changes	

7.1.4 Other methods

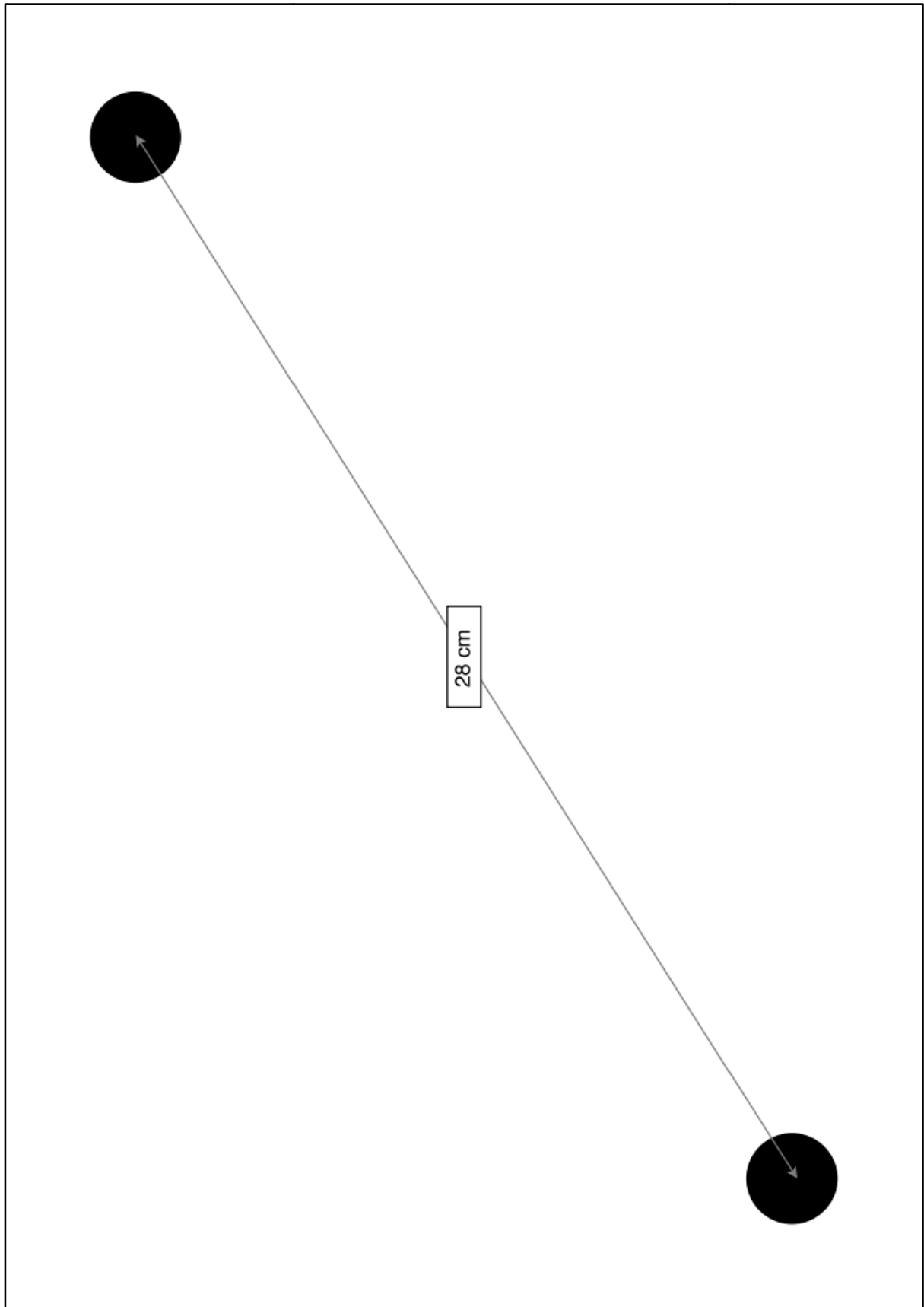
There are also other more complex models, such as the *data driven model*, *matrix decomposition* or *machine learning*. These models may be more precise than the other ones, but need lots of processing power and knowledge on more advanced algorithms.

7.1.5 Chosen model

Since the project is target to a domestic use, it needed to be able to run on most of the computers, even the ones with not much processing power.

For this reason, the model chosen to detect motion was the *background difference model*.

7.2 Foglio di calibrazione



7.3 Fonti e riferimenti

- Utilizzo di OpenCV con Qt:
https://github.com/MicrocontrollersAndMore/OpenCV_3_Windows_10_Installation_Tutorial
- Modelli per la rilevazione di movimenti:
P.-M. JODOIN, S. PIÉRARD, Y. WANG, and M. VAN DROOGENBROECK. Background Modeling and Foreground Detection for Video Surveillance. In T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, editors, "Background Modeling and Foreground Detection for Video Surveillance", chapter 24, CRC Press, July 2014.
- Qt Open Source:
<https://www.qt.io/>
- OpenCV:
<http://opencv.org/>
- OpenSSL:
<https://www.openssl.org/>
- SMTP Client for Qt:
<https://github.com/bluetiger9/SmtpClient-for-Qt>
- Diagramma a pagina 3 disegnato con:
<https://www.gliffy.com/>