Persistent Authentication in Smart Environments using COTS Hardware

Jorge Martinez

Kongens Lyngby 2017

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

This thesis studies a system that tracks principals from an authenticating point in the entrance trough his path as he moves in the building, this in order to provide authorization of services in the areas in which this user has the right privileges. Such authentication scheme allows the authorization of location-based services (for example automatic doors for restricted areas), in environments where low cost cameras are in use. The proposed system implements Persistent Authentication combining Blob detection techniques and different remote biometric factors using multiple Commercially Off-The-Shelf (COTS) cameras.

Background subtraction is performed on the video provided by the cameras, obtaining a shape around moving subjects in the environment, this shape is enhanced by different image processing techniques including Morphological transformations, providing a more complete model of the principals in the image. A series of biometrics is performed on the found model of principals, retrieving the identity made available during the principal authentication, finally the information resulting from multiple analyzed cameras is combined into one coordinate system giving as a result the position of each identity.

The implementation is used to study additional biometric factors not addressed by previous authors of Persistent Authentication and also to shine some light in the challenges presented by the usage of multiple cameras.

Summary (Danish)

Denne afhandling undersøger et system, der sporer borgernes vej igennem en bygning fra et autentificeringspunkt i et indgangsparti. Dette gøres med henblik på at yde bevilling af tjenester i de områder, hvor denne bruger har de rette beføjelser. Sådan en godkendelsesordning tillader autorisation af lokationsbaserede tjenester (For eksempel automatiske døre til begrænsede områder), i omgivelser hvor der bliver brugt billige kameraer. Det foreslåede system implementerer Persistent Authentication, der kombinerer Blob-detektionsteknikker og forskellige fjernbiometriske faktorer ved brug af flere COTS kameraer.

Kameraerne udfører background subtraction på videoen, herfra opnås en form omkring bevægelige objekter i området. Denne form forbedres ved forskellige billedbehandlingsteknikker, herunder morfologiske transformationer, hvilket giver en mere komplet model af personerne i billedet. En serie af biometri udføres på den fundne model af mennesker, dette er i stand til at identificerer personen ud fra Id'et, som er stillet til rådighed i den primære autentificering. Endelig kombineres oplysningerne fra flere analyserede kameraer i et koordinatsystem, der som resultat giver hver persons position.

Implementeringen bruges til at undersøge yderligere biometriske faktorer, som ikke blev nævnt af de tidligere forfattere af Persistent Authentication, og til at diskutere de udfordringer der opstår ved brug af flere kameraer.

Preface

This thesis was prepared at the Department of Applied Mathematics and Computer Science during the course of five months, in partial fulfillment of the requirements for acquiring the degree Master of Science in Engineering, Security and Mobile Computing.

Lyngby, 30-June-2017

Jorge Martinez

Acknowledgements

I would like to thank my supervisor, Christian Damsgaard Jensen for all the comments and guidance during the course of this thesis, and for the original idea.

Thanks to Agnieszka Golinska for her help with so many testing sessions, and additionally to Michelle Lind, Nikolaos Tatsis, and Tariq Hassen for all their help as subjects of my experiments and with final editing.

Finally thanks to my family whose support and encouragement made all of this possible.

Contents

Chapter 1

Introduction

The spread of pervasive computing is bigger every year, right now there are small connected computers in everyday objects such as screens, cameras, door locks, light bulbs and thermostats, this enabled, years ago, the implementation of smart homes, and is bringing also smart offices, integrating multiple of connected devices to improve comfort and productivity. Such smart environments can benefit from the knowledge of the presence users and also from increased security allowing their activation only to authorized personnel recognizing users' identity without the need of interaction for every device.

1.1 Motivation

A frequent objective in security is to reduce the effort that the user has to spend interacting with the security system, while maintaining the level of security that traditional security systems have. The proposed solution behind persistent authentication is to have a strong authentication system at strategic points such as the entrance of the building, in which principals would be identified with high confidence, and from that point on, the system will use sensors available in the smart environment to keep those authenticated principals' identities while they are inside the building.

Previous work, as discussed in section 2.2, have approached persistent authentication both using a Time of Flight camera and a color camera; the ideas of this thesis are: to expand previous work on using multiple cameras for covering a larger area, explore the challenges that this setup introduce, explore different factors of authentication to be used for maintaining the identity of principals, and develop a system that implements this findings and allow future work on this area such as expansion of the authentication system and integration with a location-based access control system.

In this instance we will be using COTS security IP cameras, and existing and reliable computer vision libraries and techniques to perceive the user position and maintain the knowledge of his identity while he moves in the building.

1.2 Goals

The work in this thesis has the following objectives:

- Identify the challenges that come from introducing multiple cameras.
- Explore factors of authentication and tracking methods needed to keep the identity of principals.
- Design and develop a system that allows exploring these items and that reflect the found concepts.

The developed system will be driven by the need to secure smart environments, provide user's position to services and facilitate the application and study of the involved techniques.

For the system to implement a useful persistent authentication it has to maintain recognition of principals without loosing track, which in this area is know as the ability of "persistence".

An important property related to persistence is the robustness against principals' normal behavior and interaction with each other, this includes considerations such as identifying a user which is occluded from some cameras, or in a situation where the perceived information is partially ambiguous such as principals walking close to each other.

The most important idea behind having a continuous authentication system in a building, is to be able to use the positions of the principals to verify if such person is authorized to use a certain object in the vicinity of such position, this object could be a door leading to a certain restricted area, or an electronic device such as a computer. Because of this, the position identified by the system should be adequate for such applications.

This also raises the importance of covering a big area with cameras, because of this the system needs to enable the usage of multiple cameras and have the ability to escalate in its number.

In addition, the system need to allow at least two points of expansion, the Initial Authentication mechanism (which should be independent of the rest of the application) and the biometrics should be easy to include. Firstly, decoupling from the authentication method allows the security to be increased by rising the confidence in the initial identity of the principal. And secondly, allowing the addition of biometrics facilitates the exploration of different alternatives both in the present thesis and in future work.

This property of the system also indicates a boundary, as an initial authenticator is out of the scope of Persistent Authentication and there will not be an exhaustive investigation of the best biometric factors to be used. In addition to this, the users might be concerned by their privacy, because they are being monitored constantly, in Persistent Authentication, no storage of the user position is required, so no analysis of the user behavior is possible, analysis of attacks on the implemented system to collect and store such information will not be considered in this thesis.

This is a summary of the previously described system properties:

- Good level of Persistence
- Robustness to principals interaction
- Allows to identify principals' position for location-based services
- It should enable the usage of multiple cameras
- Decoupling from the initial authentication method
- Facilitates the exploration of additional biometric factors

1.3 Summary

This thesis is composed of the following chapters: This chapter, the introduction, explains the project, its goals and motivation. Chapter 2, contains the theoretical framework and and explains the previous work in which this thesis is based. Chapter 3 connects the found literature to explain the overall proposed scheme, In chapter 4 we deal with the design of the system, the experimentation environment and how the previously described theory was applied to the final version of the system. Chapter 5 contains a detailed explanation of the implemented system, Chapter 6 explains the method designed to test the system and measure its capabilities. In chapter 7 we showcase the obtained results and finalize with the conclusions.

Chapter 2

Theory

2.1 Indoor Positioning Systems

Mainetti, et al.[17] reviewed several methods for people tracking found in the literature and classified them on 2 dimensions: enabling technology and propose of its development, from this study we can observe that, for people tracking, we can divide them in two groups: methods that require for people to carry a device, and those that do not. Table 2.1 contains this technologies.

Table 2.1: Technologies Enabling Indoor Positioning Systems

Device Required	Device not required
Vision, with building model	Vision, without reference
Vision, with pattern recognition	Natural Infra Red (IR)
Vision, with deployed targets	Artificial IR
IR with artificial light source	
Wi-Fi	
RFID	
Bluetooth	

Other authors have used technologies that require a device for authentication, Corner and Noble presented [8] a device that authenticates the user to his laptop when the user is close to it, using a short range wireless signal to establish communication and then providing proof of identity. In this case the short range nature of the communication is used to establish that the user is close to the laptop, and the user carrying the device is assumed to be the true owner of the device.

The previous scheme present the problem that the user himself is not being the one authenticated, but the device, and this allows a malicious agent to impersonate a trusted user by stealing the device. This problem is also present with technologies involving carrying a device mentioned in table 2.1

From the technologies enabling IPS, natural and artificial Infra Red have the problem that their commercially available sensors are expensive, on the other hand there are fairly inexpensive cameras and they are already deployed in buildings where security is a concern.

2.2 Persistent Authentication In Smart Environments (PAISE)

2.2.1 Continuous Authentication

In 1995 [20] Dr Simon J. Shepherd described the concept of continuous authentication for a system that kept identifying the user after the initial log in, based on the pre-registered pattern of typing characteristics. The user would log in normally to the PC using his user name and password, and from that point, the system would verify his writing pattern, and if the pattern did not fit with the pre-registered model the network administrator would be notified of a possible breach.

From that first implementation of Continuous Authentication the following scheme can be abstracted: There is a preliminary phase, in which the user registers his credentials along with his biometric features. Then when he wants to use the desired service, there are two phases, The first one consists of an authentication using the user credentials, and the second phase consists of the user being authenticated by his biometric features periodically until the user logs out from the system. This scheme is illustrated on figure 2.1

One of the important concepts of this idea is that it does not try to replace the

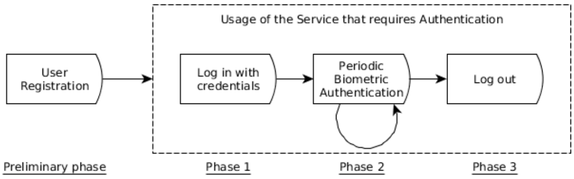

Figure 2.1: Continuous Authentication phases

authentication mechanism, instead, the system continuously validates that the user is still who he claimed to be during the authentication phase, this is to solve the problem of a malicious user taking over the computer after the user authenticated and left it unlocked, it also strengthens the security of the login as the pre-registered factors used for continuous authentication could serve as a second factor for authentication.

2.2.2 Previous work in PAISE

2.2.2.1 Origins of the idea

In 2008 Kirschmeyer and Hansen [15] take the concept of Continuous Authentication and constrain it to be more specific in order to apply it to building security. They call this Persistent Authentication, and it differs from continuous authentication in that the user is not authenticated from pre-registered hard biometrics during phase 2 (see figure 2.1), instead it uses techniques to verify that the user using the service (inside the room in this case) is still the same one that authenticated in phase 1.

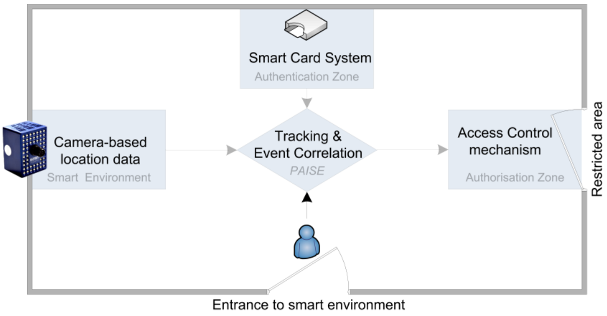

Figure 2.2: General Overview of PAISE [15]

Figure 2.2 shows the components included in Kirschmeyer and Hansen's work, In their experiments they set up a Time-Of-Flight (TOF) camera in one of the corners of the ceiling in a room, a person would enter the room, and the system would then start tracking such person based on the input from the TOF camera, the person would approach the authentication zone, and authenticate using the smart-card system, from this point on, this person would be recognized by the system as the owner of such card and would get clearance according to his access control rights, the authenticated person would then proceed to the authorization zone in order to access the restricted area behind a door, the system would check the permissions that the user has and would open the door if the user has the required clearance.

In this model, every person is tracked, even if there is no user associated with him, this is done by applying a Background Subtraction (BS) method (see section 2.3.1) and a clustering algorithm that groups pixels according to similarity between the position of the pixels and their depth information, so pixels close to each other would get grouped in the same cluster if their depth is very similar.

In this system, the phase 2 of figure 2.1 is replaced by the tracking of authenticated blobs using the method previously described and the log out is done when the blob leaves line of sight with the camera, additionally people are tracked all the time, even if they are not authenticated, which could enable its usage for alarms; and finally there is an additional verification of user rights for the usage of several services based on the position of the user.

The results from this work show that the system is able to track an authenticated user and grant him access to services according to specified security policies, however multiple users can introduce some problems.

One of the weaknesses of this system, presented by the authors, is that having two people close to each other and at roughly the same distance to the camera can make the system to swap the identities of the two users, this happens because for the tracking algorithm, there is no way to differentiate these blobs.

2.2.2.2 Resilient Infrastructure And Building Security

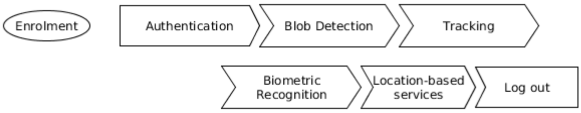

Figure 2.3: Persistent Authentication for Location-based Services [13]

Figure 2.3 shows Ingwar's approach called Persistent Authentication for Location-based Services, presented in his PHD thesis [13]. He replaced the TOF Camera for a color camera which allows the usage of remote biometrics from the camera image, and also proposes the fusion of such biometrics by using the False Positive Rate (FPR)s and False Negative Rate (FNR)s to calculate the confidence that a user has been correctly authenticated by those factors.

In the blob detection component, Ingswar's system makes use of BS to detect blobs that are moving in the scene, as in this case there is no depth information coming from the color cameras, the blob detection is then performed by grouping the pixels in the foreground that are connected to each other.

After blobs have been detected in the current frame, the systems combines Tracking information and biometric factors to recognize the previously authenticated users and assign their identity to those blobs.

Once identification of the blobs have been done, the system is able to provide the position of each user to the authorization component who grants or denies access to location-based services.

Ingwar additionally introduces a confidence variable in the user state, which models the changing confidence of measures that identify the user, such changes happen because of different events such as a biometric feature being or not available, and errors in measurements.

This confidence variable is changed according to Ingswar's Persistent Authentication Algorithm which increases the confidence every time a biometric feature is measured or also when the user authenticates, and lowers it when there is a noisy event during blob detection or in the tracking component. Finally, when a location-based service requires authorization, this confidence is also verified to maintain a level of minimum allowed confidence.

2.3 Blob Detection

2.3.1 Background Subtraction (BS)

BS consist of techniques to identify moving elements in a video, identify them as part of the foreground, and the rest as background. This is a frequent technique used in people detection.

Table 2.2: BS Algorithms Compared: symbols in order: n_s: Number of sub sampled frames, m:Number of Gaussian distributions, n: buffer size for last background frames, m: number of modes approximated for the Probability Density Function (PDF),

Method	Speed	Memory	Accuracy
Running Gaussian average	$O(1)$	$O(1)$	Limited/Medium
Temporal median filter	$O(n_s)$	$O(n_s)$	Limited/Medium
Mixture of Gaussians	$O(m)$	$O(m)$	High
Kernel Density Estimation (KDE)	$O(n)$	$O(n)$	High
Sequential KD approximation	$O(m+1)$	$O(m)$	Medium/High
Cooccurence of image variations	$O(8n/N^2)$	$O(nK/N^2)$	Medium
Eigenbackgrounds	$O(M)$	$O(n)$	Medium

In the most popular algorithms, a series of frames are analyzed to establish a background model and then if the analyzed pixel differs from the model by a lot (by some defined measurement), the pixel is considered to be foreground. After this the background model is updated to include for changes in light.

Piccardi [19] wrote a review of the main methods for BS, in the paper, he presents each method, an analyses the speed, memory and accuracy, table 2.2 shows his findings. In the following sections I describe the most relevant ones.

2.3.1.1 Running Gaussian Average

Wren et al [24] proposed a method for fitting a Gaussian PDF for each pixel, this in order to describe the probability of a pixel to be part of the background using a accumulated mean function to update the mean of the Gaussian, and a similar function for the variance.

$$\mu_t = \alpha I_t + (1-\alpha)\mu_{t-1} \tag{2.1}$$

Equation 2.1 is the equation used for updating the mean of the Gaussian of each pixel, α is a parameter that describes how fast the background is supposed to change and I_t is the current intensity of the pixel.

To actually establish if a pixel belongs to the foreground The current difference from the mean, is compared to a weighted value of the variance, the inequality

2.2 shows how to verify this.

$$|I_t - \mu_t| > k\sigma_t \tag{2.2}$$

2.3.1.2 Mixture of Gaussians

This model is inspired by backgrounds with regularly moving objects, such as leaves in a tree, it works by assigning to each pixel a mixture of Gaussians (see equation 2.3) describing the probability of observing a RGB color in that pixel, with each of the K Gaussians representing one possible object in he background, and $\Sigma_{i,t}$ the covariance matrix of pixel i at frame t.

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(x_t - \mu_{i,t}, \Sigma_{i,t}) \tag{2.3}$$

In order to discriminate a pixel between background or foreground the Gaussians are sorted by amplitude ω and standard deviation, then the first B Gaussians to satisfy inequality 2.4 are representing the background. T is a threshold parameter.

$$\sum_{i=1}^{B} \omega_i > T \tag{2.4}$$

Having this, with each frame and pixel, to establish to which Gaussian belongs a pixel color x_i, each Gaussians is evaluated in order, selecting the first one to satisfy inequality 2.5, then the selected Gaussian's parameters are updated using cumulative averages. If there is Gaussian is selected, the last one is replaced with a new one centered in x_i.

$$\frac{(x_i - \mu_{i,t})}{\sigma_{i,y}} > 2.5 \tag{2.5}$$

2.3.1.3 Kernel Density Estimation (KDE)

The idea behind this algorithm is to model the background using a histogram of the last n values of the pixel identified as background, in order to make a continuos model of such histogram a the PDF is modeled as a sum of gaussians with center in the values, as shown in equation 2.6

$$P(x_i) = \frac{1}{n}\sum_{i=1}^{n} \eta(x_t - x_i, \Sigma_t) \tag{2.6}$$

For this model, to distinguish if the pixel color x_t belongs to the foreground it is verified if the inequality $P(x_t) < T$ is satisfied for a threshold T.

One advantage of this model is that it does not require parameters that change depending on the scene, although implementations might allow configuring n and T, but such parameters are very intuitive.

2.3.2 Histograms of Oriented Gradients (HOG) Descriptors

A feature descriptor represents an image, typically as a vector, in terms of some descriptive qualities, in the case of Histograms of Oriented Gradients (HOG) the vector describes histograms of the change in intensity from each pixel to its neighbours.

Dalal and Triggs [9] proposed HOG a new feature descriptor and suggest its application for pedestrian detection, by using the descriptors to find a shape describing the image and classify such shape into pedestrian or non-pedestrian using a linear Support Vector Machine (SVM).

In order to obtain the HOG descriptor of the image to be analyzed, first, the gradient of intensity change has to be calculated, this can be done by applying two 1D centered point derivatives $[-1, 0, -1]$ one vertical and other horizontal, this will give the change of intensity with two components (in the y and x axis) for each pixel in the image; this two values are then transformed to polar coordinates, which gives the magnitude and direction of the intensity gradient, angles between 180°- 360° are re-mapped to 0°- 180°, ignoring the 'sign' of the vector.

Then the image is analyzed pixel by pixel in small blocks (of 8x8 pixels in the paper) condensating the magnitude and direction of each pixel in a orientation histogram with 9 bins for angles between 0-180, this is done by finding the two bins that enclose the pixel direction value and adding the magnitude of the pixel proportionally to both bins. To make the histograms independent of lighting conditions, a normalized version of them is calculated, this is done by iterating over groups of a few blocks (2x2 group of blocks for example), and the groups overlap with each other, in each group the histograms of blocks are concatenated and normalized, and stored for that group before moving to the next group. After every group is analyzed, the resulting normalized histograms are concatenated, resulting in the HOG descriptor.

After finding the HOG descriptor of two images, those could be compared to find out if they are similar and in this way detect if there is a pedestrian in the analyzed image, based on the HOG descriptor of a pedestrian. But the previously described process assumes that the pedestrian in the analyzed image is filling the frame in the same proportion as the pedestrian in the model, to solve this, the image is analyzed in a sliding window manner iterating in two levels, in the first level the image is downscaled by a parametrizable factor, in the second level the the image is cropped to meet the window size (the size of the reference image representing a pedestrian) and in each iteration the cropping area is shifted by a parametrizable value.

In the proposed algorithm, a single image is not used as an object reference, to find out if an image contains a certain object (pedestrians in this case) a SVM is used to discriminate in each iteration if the window's content belongs to a previously trained category. This training is done by presenting the SVM with a classification of pedestrian/non-pedestrian HOG descriptors of reference images, so the SVM develops a hyperplane that separates such images. Because of this, the SVM is able to classify new images (each window) into containing or not a pedestrian.

The described algorithm returns a list of rectangles surrounding the windows in which the pedestrian was found. This can be used as a position and size of the person in the image and as a discriminator in areas of the image containing pedestrians or not. The authors recommend for the reference images to have some padding between the person and the border of the image, because of this, the found bounding boxes may have the same padding. All of this characteristics determine the precision of the result of this method.

2.4 People Recognition

This section describes techniques found on the literature that can be used to identify the person in an image, this techniques can be applied after the areas of the image containing people have been identified and there is a reference of the person that can be linked to the found person in the image.

2.4.1 Blob Tracking

Object tracking is usually studied as an alternative to object tracking by identification, in order to speed up the search of the object in the image, the algorithms analyze the movement of the object and then make a small feature search around the predicted position.

For this section the interest in tracking is to have a measure of similarity between found blobs and previous observed blobs, so other forms of tracking where also studied.

2.4.1.1 Kalman Filter

A Kalman filter is an algorithm for prediction and correction of the state for a linear dynamic system with a known behavior, his algorithms models the system as having a transition matrix A from the previous state to the new state, a matrix B as the matrix modeling the control over the system, the previous real state x_{k-1}, the control input u_k, and the process noise w_k which is assumed to have a normal distribution ($w_k \sim \mathcal{N}(0, Q_k)$). Equation 2.7 models then the current real state of the system x_k.

$$x_k = Ax_{k-1} + Bu_k + w_k \tag{2.7}$$

the system is described by a matrix H transforming from the real measurable properties of the system (x_k) to the modeled system state, and some measurement noise assumed to have a normal distribution ($v_k \sim \mathcal{N}(0, R_k)$). The recorded state ($z_k$) is then described by equation 2.8

$$z_k = Hx_k + v_k \tag{2.8}$$

Prediction is done by equations 2.9 and 2.10 that predicts the covariance error, where $\hat{x}_{k-1}$ and P_{k-1} are the recorded state from time $k-1$:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{2.9}$$

$$P_k^- = AP_{k-1}A^T + Q_k \tag{2.10}$$

In each measurement of the system, the recorded state is updated with equations 2.11, 2.12 and 2.13, Where

$$K_k = P_k^- H^T (HP_k^- H^T + R)^- 1 \tag{2.11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.12}$$

$$P_k = (1 - K_k H)P_k^- \tag{2.13}$$

2.4.1.2 Kalman Filter for Multiple object tracking

Li et al. [16] propose a method for using a Kalman filter to track multiple objects in a video. In their model, the position of the object is measured as well as the width and height, an it assumes that from one frame to the next, the speed can be considered constant, in this sense, the state of each object of the system can be described by the variable x_k in 2.14

$$x_k = [x_{0,k}, y_{0,k}, l_k, h_k, v_{x,k}, v_{y,k}, v_{l,k}, v_{h,k}]^T \tag{2.14}$$

This state describes with $x_{0,k}$ and $y_{0,k}$ the coordinate of the object on the x and y axis, with l_k and h_k the half of the width and height respectively, and with $v_{x,k}$, $v_{y,k}$, $v_{l,k}$ and $v_{h,k}$ their respective rates of change.

The measurable properties are then represented by z_k in 2.15.

$$z_k = [x_{0,k}, y_{0,k}, l_k, h_k]^T \tag{2.15}$$

The transition matrix A is 2.16

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

Which from equation 2.7 and 2.15 it can be understood that A is linking the rates of change with their respective the position and dimension values and it is assuming constant speed, at the time Δt between frames.

The measurement transformation matrix H is proposed as 2.17

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.17}$$

Matrix 2.17, vector 2.15 and equation 2.8 show that the only values included in the measurement are the position and the dimensions of the object, and they are direct mappings.

To track multiple objects two functions are defined that allow to compare which object is better described by a Kalman tracker, function 2.18 describes the difference between the current position and the previous tracked position, and function 2.19 describes the difference between the previously tracked area and the current measured area.

$$D(i,j) = \frac{\sqrt{(x_k^i - x_{k+1}^j)^2 + (y_k^i - y_{k+1}^j)^2}}{Max_n(\sqrt{(x_k^i - x_{k+1}^n)^2 + (y_k^i - y_{k+1}^n)^2})} \tag{2.18}$$

$$A(i,j) = \frac{|S_k^i - S_{k+1}^j|}{Max_n(|S_k^i - S_{k+1}^n|)} \tag{2.19}$$

In both equations k denotes the previous frame, $k+1$ the current frame, i is the index of an object in the previous frame, j an object in the current frame, and Max_n a is the maximum value iterating trough all the objects in the current frame (each object identified by n in each iteration). From this functions, the author derives a score (equation 2.20) used to measure how closely an object j is described by a Kalman tracker of an object i. having α and β as weighs of the score with $\alpha + \beta = 1$.

$$V(i,j) = \alpha D(i,j) + \beta A(i,j) \tag{2.20}$$

Using these definitions, the tracking algorithm works as follows:

1. In the first frame that a moving object appears a new Kalman tracker is assigned to that object and initialized, if it is not the first frame, but the object doesn't fit any previous tracking windows, then a new Kalman tracker is assigned to that object and initialized.

2. Measure the properties of moving objects in the frame, use the score 2.20 to determine the best matches from previous objects in the frame to current ones (the lower the better).

3. If one of the objects is occluding another treat this as a new object (this is called a merge operation), if a previously merged tracker is separated again in 2 objects, treat those objects as the previously merged ones, if a tracker that was not the product of a merge is split, then create new trackers for them.

4. If an object leaves the frame stop tracking that object.

2.4.2 Remote Biometrics

In the usual described types of factors, something you know, have or are, biometrics fall into something you are, so they are measures of characteristics of a person, such as their faces or voice, this characteristics can could be used by another human to describe him, as the previously listed ones or could be more hidden such as fingerprints and retina or iris patterns. However this information is always public and could be collected by a human without the user knowledge. This is why usually biometrics are used together with another types of factors, such as a password or an id card.

When designing a system that uses biometrics, measurement of features usually requires user interaction, which would break the propose of persistent authentication of keep authentication of the user with as little interactions as possible, so the biometrics of interest are the ones that could be measured at a distance of the user, remote biometrics, such as face or several soft biometrics as iris and skin color.

Dantcheva et al. [10] collects several used soft biometrics and classifies them with properties like Permanence of the feature, Distinctiveness and subjectivity of its definition. Here I name some of them that are useful in this context.

Hair color According to the review by Dantcheva et al. [10] hair color can be measured using Gaussian mixture density models, which accounts for the broad range of colors.

Skin color Skin color has been previously successfully detected from color images using color segmentation [7] among other techniques.

Clothes color Dantcheva et al. [10] proposed this soft biometric making emphasis in its usefulness for the scenario of video surveillance, the proposed method describe subjects using a histogram of 11 specific colors that are clustered in the images through fuzzy KNN classifier, in this way the person is described according to the histograms of colors found in the upper part and lower part of the body.

Height This feature could be measured either by using several cameras to obtain a 3D representation of the body, or by comparing the person to an object of known height as proposed by BenAbdelkader and Davis [10].

Body dimensions BenAbdelkader and Davis [5] describes as well a method for measuring shoulder breath and height using images provided by surveillance cameras, it employs model fitting of a 3D ellipse that takes into account the velocity of the person to find out its orientation.

Gait The measuring of the walking pattern, as purposed by Wang et al. [23] requires finding the silhouette of the person in the image, then the distances from the centroid to all the points in the silhouette are measured and normalized, this is transformed in a signal that is compared to the template using a eigenspace (more on this on section 2.4.2.4)

Weight Estimation Velardo and Dugelay [21] proposed that there is a linear dependency between several measurements from the body and the weight, such dependency can be used to take remote measurements from images and estimate the weight of the person, using such value as a biometric feature.

2.4.2.1 Histogram

As proposed by Dantcheva et al. [10], the color of the clothes can be used as a feature to include when identifying people in videos, they use a histogram of recognizable colors as described in several cultures because using a lower amount of bin of colors have been found to improve the accuracy of object detection by color.

A color histogram is proposed by Niinuma et al. [18] as one of the soft biometrics used in their continuous authentication system for PC, they use both face and clothes color histograms in the RGB color space, they do a 3D color histogram assigning each channel of the color space as an axis, making a 3D matrix of 16 x 16 x 16 and counting for each cell the amount of pixels in such range.

When the user logs in, the system calculates the Histograms of the user and stores it as the template to use for comparison, then periodically an image is capture with a web camera and the biometrics are chequed, when checking the color histograms, the Bhattacharyya coefficient is used to compare the taken image histograms with the ones stored as templates.

2.4.2.2 Skin Color

Chai and Ngan [7] propose a method for detecting faces in video conferences, they present an algorithm that first does color segmentation using a color ranges in the color space YCrCb, these ranges where obtained empirically from images of people of all races. This color range is for Cr [133;173] and for Cb [77;127]. Their findings for different races are shown on figure 2.4.

After this color segmentation, some objects in the background are still included in the resulting mask, so they also employ other techniques to filter out pixels found with this range that do not belong to a face, namely: Density regularization, Background subtraction, and geometric analysis of the found shapes. A different approach is proposed by Gomez and Morales [12], they created a machine learning algorithm called Restricted Covering Algorithm, which receives a set of operators, constants and variables, and evaluates its combinations over a set of training examples, increacing the reliability of the found rules on each iteration. After running the algorithm over a set of 32 million labeled pixels in the RGB color space, and with the operators $[+, -, /, ^2]$ and operands $[r, g, b, 1/3]$ they obtained the rule shown in figure 2.5 which obtained a performance with recall=93.7%, precision=91,7% and success rate= 92.6%.

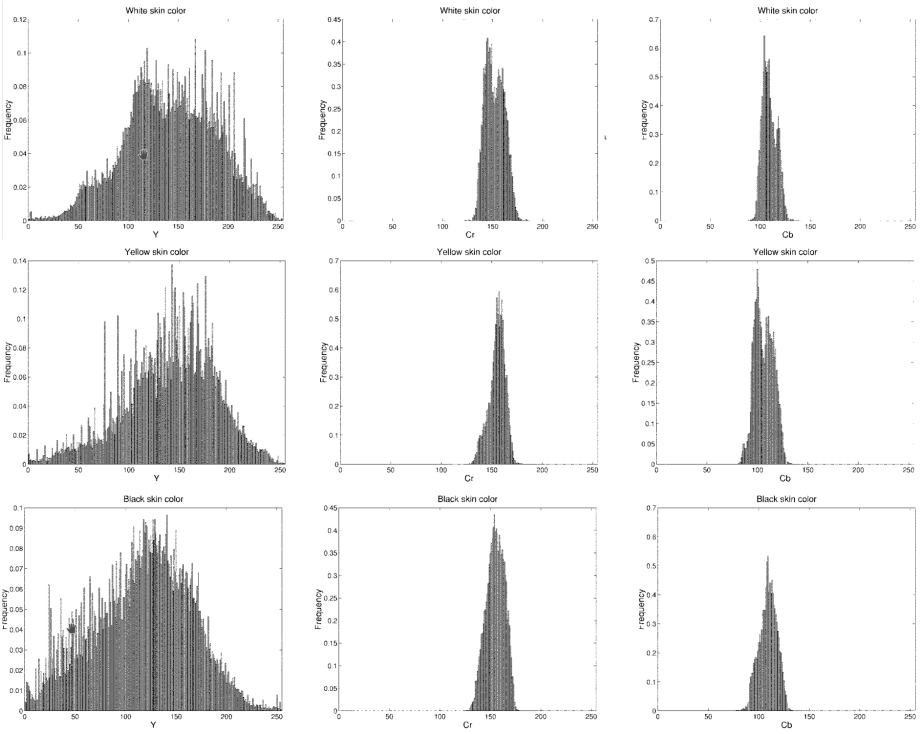

Figure 2.4: Skin ranges for different races [7]

2.4.2.3 Face Recognition

In part II of Handbook of face recognition [14], The author makes a review of the Face recognition techniques, finding that the first step is always a technique for finding every face in the image, ideally this would be done independently from the orientation, position, expression, scale or rotation of the faces.

For detection of faces, Li and Wu found that according to the literature the best method for face detection is using Haar-like proposed by Viola and Jones [22], the usage of Haar-like features is very efficient and it works independently of the scale of the face, additional features can be used to improve results on non frontal faces.

Viola and Jones' method works on the basis of 4 concepts, Haar features, Integral Image, AdaBoost and Cascading.

$$\begin{array}{lll} \frac{r}{g} & > 1.185 & \text{and} \\ \frac{r*b}{(r+g+b)^2} & > 0.107 & \text{and} \\ \frac{r*g}{(r+g+b)^2} & > 0.112 & \end{array}$$

Figure 2.5: Rule for classifying a RGB color as skin [12]

Haar Features Figure 2.6 shows the proposed types of features by Viola and Jones for face recognition, these features model characteristics found on faces on grayscale images, for example a 2 rectangle feature can be used to describe the difference on intensity between the eyes and the cheeks, with the eyes darker and the cheeks lighter. A feature describe a characteristic by adding the sum of the intensity values of the pixels in the area marked by the white rectangle and subtracting the sum of intensity values of the pixels in the area marker by the black rectangle. A feature can be of any of the types shown on Figure 2.6 and of any size and the value it returns form evaluating it on a portiion of the image reflects how good it describes the characteristic (the higher the better). In order to find these characteristics the features have to be evaluated at every portion of the image.

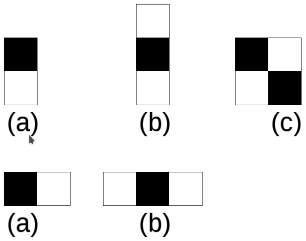

Figure 2.6: Types of Haar like features, proposed by Viola and Jones: (a) 2 rectangles, (b) 3 Rectangles, (c) 4 rectangles

Integral Image Haar features require to compute several times the sum of intensity of rectangular regions, and this leads to a lot of redundant computation, in order to solve this problem, Viola and Jones propose the concept of an integral image, in this, every pixel (x_i, x_i) gets assigned the value of the sum of intensity of every pixel that is at a position (x, y) with $x < x_i$ and $y < y_i$ (pixels at the top left area). This can be computed recursively using the formula 2.21 This allows to calculate the sum of intensity of any rectangular region in the image, by using operations on the 4 pixels of the corners of the region in the integral image as shown in figure 2.7

$$
\begin{aligned}
s(x,y) &= s(x,y-1) + i(x,y) \\
ii(x,y) &= ii(x-1,y) + s(x,y)
\end{aligned}
\tag{2.21}
$$

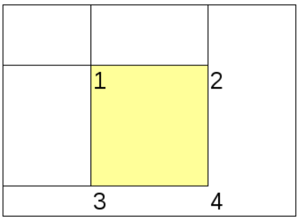

Figure 2.7: The sum of intensities the shaded area is equal to the integral image pixel values $II_4 - II_2 - II_3 + II_1$

AdaBoost As mentioned before, the Haar features can be of any of the 3 types of any size, and in any position of the detector, and a Haar-like features face detector has a resolution of 24x24 pixels, this means that there are over 160.000 features to evaluate. To reduce the amount of features to evaluate and improve the precision, AdaBoost is employed to calculate a strong classifier as a linear combination of some of the previously described features (week features), for this only the features that are able to recognize more than 50% of the training set are selected as weak features. And using this week features a series of iterations are performed to improve the weights of each weak feature in the linear combination, the weights of the weak features are calculated as a function of the importance of the examples in the training set, and the importance of each example is modified to make more relevant the ones that where wrongly classified in the previous iterations.

Cascading The final characteristic in Haar-like feature detector, Cascading, is used to chain the precision of several detectors, in a way that fast but not so reliable strong classifiers are computed first to discard rapidly windows of the image that are clearly not a face, and then increasingly reliable strong classifiers are computed so that at the end of the chain the detector has the expected precision. The chain is made of several strong detectors, connected as shown in figure 2.8 so that at the end of each stage, it is decided whether a window is definitely not a face or maybe a face, if it is not a face, the algorithm ends, and returns such result, if it is maybe a face, the next strong classifier is evaluated, if this continues until every stage has been evaluated positive then the window is classified as a face.

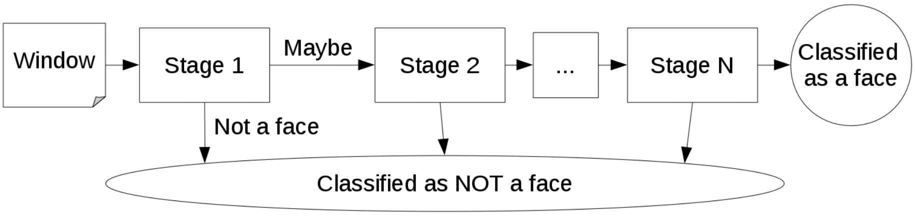

Figure 2.8: Cascading algorithm for the Haar-like features detector

2.4.2.4 Face identification

Once the face have been detected, an algorithm for face recognition can be used such as Eigenfaces, this techniques involve machine learning using samples of the faces to be recognized.

Eigenfaces works by using Principal Component Analysis (PCA) to model the labeled samples of faces as a subspace created by the principal components (eigenfaces) of images created by substracting the average face to each sample. Using this model each face can be reconstructed from a linear combination of the eigenfaces. Then when analyzing a new image of a face, it is projected on the subspace and the resulting coefficients are compared to the coefficients of known faces using euclidean distance and the closest one is selected as the identified face.

2.4.2.5 Feature Matching

Feature matching is the process of finding the correspondence between two images, this is done by finding representative points (features) in an image, calculating a descriptor of such feature, as a model of a small sub-image around the point, and then finding a single point in the second image that is identified by the descriptor.

One recent descriptor is KAZE [3], which is a scale and rotation invariant descriptor proposed by Alcantarilla et al..

To make the descriptor scale invariant, a scale image space is calculated using a diffusor function that softens the image but leave object's edges intact, this is done to extract descriptor for several scales.

Good features are found by using a hessian detector, normalized by the scale in a 3x3 pixel patch, its dominant orientation is then detected rotating a circular window and calculating the first order derivatives weighted by a gaussian.

Then using first order derivatives L in different subregions over a small patch around the point, the descriptor vector is calculated as $d_v = (\sum L_x, \sum L_y, \sum |L_x|, \sum |L_y|)$ and this is weighted with a Gaussian σ_2 over a 4x4 patch.

2.5 Camera Model

The camera model is a transformation from a point in a real world coordinates system (in 3 dimensions) to a coordinate system in the captured image (2 dimensions). This model simplifies the real camera by supposing that it works like a pinhole camera, i.e. a camera that captures light coming in from the outside, trough a small point hole, and projecting such light into a photosensitive material.

This model in a simple form is described by the equation 2.22 where Q_i is a point in real world coordinates, q_i in image coordinates, R is a rotation matrix, t a translation vector, (having $[A;t]$ as a matrix concatenating such elements) and A is a transformation containing the camera parameters.

$$q_i = A[R;t]Q_i \tag{2.22}$$

The rotation and translation moves the coordinate system to match in origin with the image coordinate system, while the camera parameters (containing focal length and center point of the photo-sensor) match the image compression effect produced by the focal length, figure 2.9 illustrates such parameters.

Additional parameters can be introduced to the model to account for other effects product of different conditions, one of this is the radial distortion, produced specially by fisheye lenses.

The effect of such distortion is approximated by a Taylor expansion with n coefficients denoted by k_n.

[1] Based on commons.wikimedia.org/wiki/File:Camera_Oscura.jpg Carlo.benini-Own work, CC BY-SA 3.0

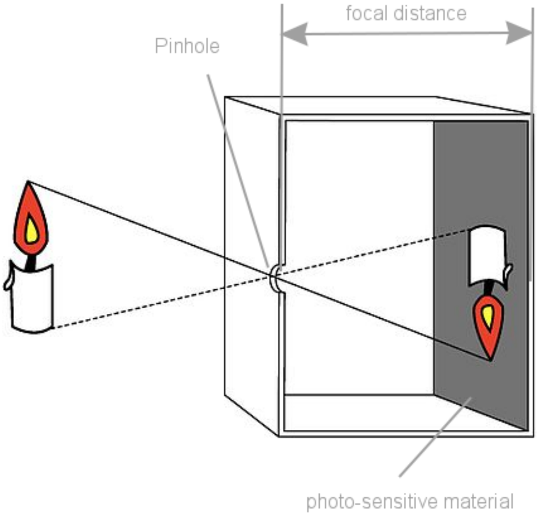

Figure 2.9: Pinhole camera model [1]

2.5.1 Camera Calibration

Camera calibration is the process of estimating the camera internal parameters, using known points in the real world coordinate system to map them in the picture coordinates minimizing iteratively the produced error.

2.5.2 Homography

An important concept when dealing with multiple cameras is the homography between planes in two different coordinate systems. An homography is a transformation between the coordinate systems of such planes (from plane 2 to plan 1 in this case).

$$q_1 = Hq_2 \tag{2.23}$$

In this equation the points are in homogeneous coordinates, where the first 2 positions are the x and y coordinates and the third position is the scale of such coordinates. Having that an homography is a full rank square matrix, the inverse is the opposite homography, so H^{-1} transforms from a plane 1 to a plane 2.

CHAPTER 3

Analysis

A persistence authentication system is a bridge between two components of a smart environment: an Authentication system, and the devices and services in the building, offering security and enabling location-aware services.

In this thesis we are concerned with the application of such system in a multi-camera multi-scene [1] environment using COTS cameras.

3.1 Offered Services

Persistent authentication systems not only serves for protection, but it also provides other systems with users' position and identity in order to provide personalized services or additional security mechanisms.

[1] This refers to a scene with several cameras, each of which are located in separate scenes

3.2 Selected Techniques

This section shows the analysis made over the studied techniques in chapter 2 along with the reasoning for the selected techniques.

One of the objectives in this work is to use proven implementations of the most complex algorithms, because of this, the selected algorithms to use for this work are limited to the ones implemented in OpenCV 3.2. This library was selected for being the most used library for computer vision.

3.2.1 Blob detection specifics

In order to detect people in the building, some human characteristics have to be detected in the image, this section presents two relevant techniques found in the literature.

3.2.1.1 Background Subtraction

Previously in section 2.3.1 a comparison found in the literature was presented, it shows that Mixture of Gaussians and Kernel Density Estimation (KDE) are techniques both efficient and with good performance. When compared [11] KDE shows better performance in high contrast images and has the advantage that the kernel parameters are automatically adjusted, and it offers shadow detection.

In OpenCV an implementation of Mixture of Gaussian is implemented as suggested by Zivkovic and Van Der Heijden [26], in their work they offer a technique to continuously update the parameters for Mixture of Gaussians and enable detection of shadows. In the implementation for OpenCV, the model allows to set the parameters of amount of frames in the history that affect the background model and the threshold of the Mahalanobis distance (in the squared quantity of σ) between the Gaussian component and the observed pixel to determine if it is part of the background or not. Thanks to this implementation, you get the performance and speed of mixture of Gaussians without having to set some of the parameters, and the left parameters are more intuitive to tweak than the ones in the original algorithm, with also shadow detection.

3.2.1.2 Histograms of Oriented Gradients (HOG)

This technique is appropriate for Indoor Positioning Systems because it is able to find the human shape independently of other factors such as changes in illumination and the movement of the person, it would also detect people completely and not partially as it can happen with other techniques relying on movement detection.

One characteristic that can make the implementation of this method costly is the need of a large training data to include the particularities of the implementation, some already trained models could are provided on OpenCV, and some training data could be found in the literature, although it is usually limited in the angle and focal length of the camera. For example the INRIA dataset [2] is made of images taken from an eye level perspective and with a normal focal length.

3.2.1.3 Comparison

Table 3.1 summarizes the comparison between this two techniques for people detection. Tree characteristics are compared, detection, output and some limitations. One advantage of HOG over Mixture of Gaussians is that it can be trained specifically to detect humans, while ignoring other elements present on the picture. This detection comes in the form of a bounding box for HOG and in a silhouette to Mixture of Gaussians, this last one is more useful for measurement of some of the discussed biometrics (such as histogram of colors or height). In the case of hog, additional processing would be necessary to achieve the same result.

Mixture of Gaussians detection performance is affected when the person is still as still people start to be included in the background when they stop, also if the person is moving very slowly the edges of the person are detected for being different to the background, but the difference between a part inside such edge and another one in the vicinity could be imperceptible for the algorithm, waking holes in the result, also if the intensity and color of some parts of the person (with shadows for example) are similar to the background, the algorithm can fail to recognize them.

On the other hang HOG is limited in terms of the position of the person, for example in the work of Zhang et al. [25] a HOG detector had to be trained for each position of the person (for example standing or crouching) and if the camera can detect the person at different angles, this differences have the same effect and would require training in several angles in addition to the positions.

Table 3.1: Comparison between HOG and Mixture of Gaussians as mechanisms for people detection.

	Mixture of Gaussians	HOG
Detection	•Detects any movement	• Detects human shape
Output	•Silhouette of the person	• Bounding box around the person with some padding space
Limitations	• Cannot detect still people • Can fail to detect parts of a person	• Cannot detect people in not trained positions

3.2.2 Recognition specifics

Once the people in the building has been detected, a method to correlate them with the authenticated identities has to be applied. In the following we discuss the techniques studied in chapter 2.

3.2.2.1 Tracking

In the work of Ingwar [13] a tracking algorithm (Kanade-Lukas-Tomasi (KLT)) is used to follow the person when there is no doubt of their identity, while this tracking is in operation, no identification is made to verify the identity of the tracked person. This helps to improve the performance of the persistent authentication system. For this work the tracking algorithm is used as an identification method, in order to describe the current behavior of the person in terms such as its position, direction, speed, and apparent size.

Other more recent trackers such as Kernelized Correlation Filters (KCF) can be used for the propose of following an identified principal, nevertheless both KLT and KCF share the property that they are based on the input of a single camera image, an thus no connection can be made between cameras.

For Identification, a simpler Kalman Filter is more pertinent, (this was used in Ingwar's work to reestablish tracking after an occlusion) such technique can predict the next state of a measured object, in the specified terms, and it is independent of the measurement technique. This prediction can be compared to the observed state of the principal in order to generate a score of their similarity.

3.2.2.2 Remote Biometrics

To the best of owr knowledge there is not a widely used library for biometric recognition, but several algorithms employed in some biometric recognition are implemented in OpenCV and some biometrics are not so complex to implement, for this part of the system, the availability of a library that implements certain biometric techniques is not going to be a strong filter in the decision of using such biometric factors.

Factors based on the appearance such as skin and clothes color, measurements of the body or face, and characteristics such as beard or glasses are very useful in this context as they are easy to compute and the information is available in each camera frame.

Histograms: For biometrics involving histogram of colors, a technique for reducing the influence of lighting is necessary because in the path from one point to another in a building a principal might cross different illumination conditions. A common technique for minimizing lighting intensity influence is to convert the image to color spaces with an intensity component, such as Hue-Saturation-Value (HSV) or YCbCr and ignoring the intensity component during the analysis of color. Another consideration to have for histograms is the selection of colors, this comes in form of the amount of bins used, each one determines a different color to take into account, one option in the HSV space is to use 12 values for the HUE axis (with a similar treatment for saturation) with the uniquely named colors separated by 30° as shown in figure 3.1, the importance of such separation is that there should be some distance between the average of each class, so a different number of bins could be used, having that the higher the number of classes would mean that a more detailed differentiation would be done, with the risk of having similar colors classified as distant.

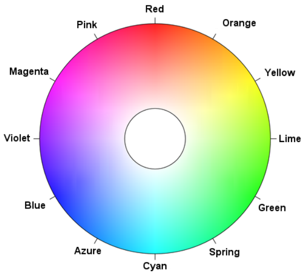

Figure 3.1: Color classes for histograms in HSV

Feature detectors: Feature detectors are commonly used for object recognition, in this instance they are useful as they can capture relevant characteristics during login, and then when the principal is detected in another frame, the captured features can be searched in the detected principal, a high number of matching features would indicate a high probability that it is the same principal. The most famous feature descriptor is Scale-Invariant Feature Transform (SIFT) as it usually presents superior performance and speed when compared to other feature descriptors, a downside of SIFT is that it is patented and its use requires to buy the rights to use, some alternatives will be considered such as AKAZE[4], such feature descriptors are found in OpenCV.

3.3 Specific Considerations

3.3.1 SWOT Analysis in PAISE

In previous designs of Persistent Authentication In Smart Environments (PAISE) [15] the authors proposed the analysis presented in table 3.2. And details and their relevance to this work are presented bellow.

Usability is offered by the ability of authorizing the user without his interaction, this characteristic is key to continuous authentication as it is one of the biggest advantages compared to traditional security mechanisms in buildings.

Intelligence makes reference to possibility of authorizing the usage of any object in an smart environment, and for this continuous authentication scheme, insensibility is one of the goals, so this strength is also relevant to this work. This is also considered an opportunity because implementing such system enable new types of services in a smart environment that take advantage of the location of its users.

Security Kirschmeyer and Hansen's work is based on blob tracking, this work will also include biometrics to strengthen the continuous authentication, but biometric factors have false positives and this could lead to allowing an attacker to impersonate a rightful hones user.

Table 3.2: SWOT analysis in [15]

Strengths	Weaknesses
• Usability • Intelligence	• Security • Privacy
Opportunities	Threats
• Intelligence	• Cost • Privacy

Privacy As the system tracks in real time the position of the user, and requires installing cameras that cover the whole building, such system could be a threat to users' privacy, both by legitimate but malicious users or external attackers. It is also considered a threat because fear of compromised privacy could restrict the implementation of the system.

Cost In [15] the cost was a risk because of the unknown future of TOF cameras prices, in this work, the cost is not a risk because of the usage of COTS cameras.

3.4 Attacker Model

This work is concerned with two types of attackers, they could be either external attackers or internal malicious users, both of them would be motivated to gain access to areas, services and objects they would not have permission to use, these attackers would try to gain access to the such resources by physical means, this means trying to walk in pretending to belong to the building.

External attackers are people not registered as users of the smart environment that want to make use of it, they could attempt to gain access to restricted resources by braking in, following someone with access permissions or stealing a legitimate user's credentials.

Internal Malicious Users have access to the smart environment, and to some resources, and would attempt to access restricted resources by impersonating other system users.

Chapter 4

Design

4.1 System structure

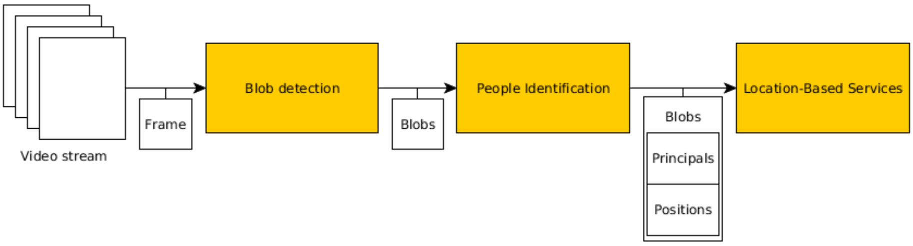

Figure 4.1: General structure of the system

The system will implement continuous authentication by sending each frame of every camera to a blob detection component that will look for people in the images, such output is going to be handled then by a principal identification component that will use biometrics and previously captured information about the principal to assign the appropriate labels to each person and calculate his position in real world coordinates, which can then be used for providing location based services. This structure is depicted in figure 4.1.

For the implementation of the blob detection component, two possible methods for Blob detection where presented in section 3.2.1.1, mixture of Gaussians and HOG, from this two methods, we select Mixture of Gaussians because its output (a silhouette) is more precise and therefore useful for the identification component, and additionally because of the chosen camera and its orientation, the training for HOG would require a great amount of images, for several poses, in several angles from the camera, because a person is perceived as a very different shape when it is far away from the camera compared to when it is straight below the camera, and also several people would need to be involved.

The identification component will use biometrics and measurements of the principal's behavior in order to assign a previously authenticated identity to the blobs in the current frame, and then a model describing the orientation of the person plus the camera model will be used to convert the position of the principal in the frame to a position in real world coordinates.

4.1.1 Main Components

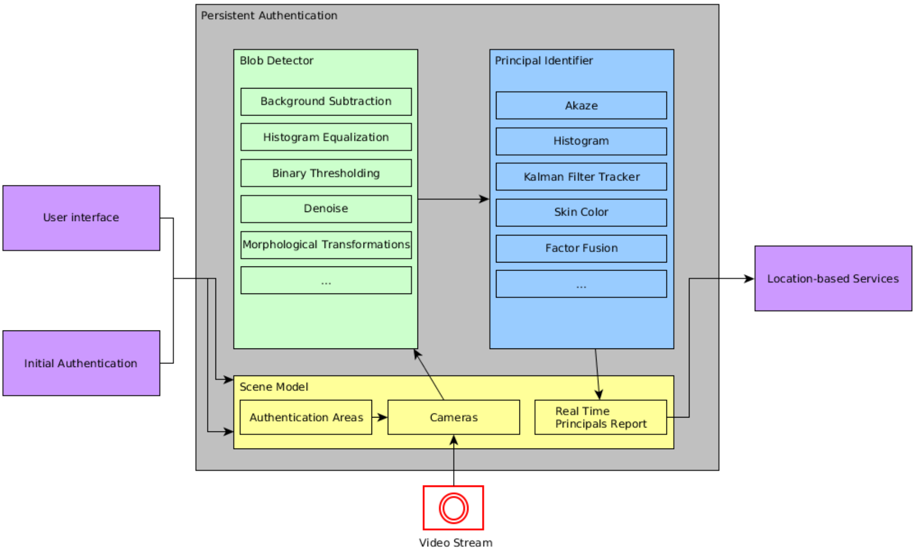

Figure 4.2: Conceptual components in the system

Figure 4.2 illustrates the detailed structure with the main components of the system, here a third general component is introduced (Scene model) and the boundaries of the system are established.

The Persistent authentication system is implemented as a component of the final solution, leaving outside of its scope the user interface, the location-based services and the modules in charge of the initial authentication. This external components will interact with Persistent authentication using the interfaces exposed by the Scene Model. With this component, during installation of the system, the User Interface can configure the Cameras and Authentication Areas located in the building. Later, when the system is fully deployed the Initial Authentication Modules can notify the system when a user has been authenticated, providing his identity and authentication zone; with this information and the configured parameters, the Scene Model knows which Camera is currently recording the principal. Finally a Location-base service can access the latest available positions of every principal present in the building.

The Scene Model component contains the following components:

Authentication Areas Stores the coordinates of the authenticated areas and the cameras that can perceive them, allowing to identify which camera stream needs to be identified when a user authenticates.

Cameras Stores the configuration of each camera and coordinates the analysis of the camera stream.

Real Time Principals Report Allows to consult the position of the principals according to the last analysis, this is provided in parallel to the rest of the operation, so consulting this report does not slow down the analysis of the camera stream.

When a Camera component is analyzing the most recent captured frame, first it will apply a series of techniques to detect moving blobs in the image, such techniques will receive and return either the frame or a mask (a black image containing in white the areas where blobs are), in this way the frame will have some pre-processing to improve the blob detection, the actual background subtraction algorithm will be run, and then the returned mask will be improved by the successive techniques, bellow is the description of such techniques in the order of application. All of them can be implemented with functions found in OpenCV.

Histogram Equalization This is a normalization technique that aims to reduce the influence of lighting changes in background subtraction algorithms, this works by increasing the contrast of the image and normalizing the brightness values of each pixel.

Background Subtraction In this component, the selected implementation from OpenCV for mixture of Gaussians selected in section 3.2.1.1 is used, including the detection of shadows, in order to be able to eliminate them.

Binary Thresholding This component eliminates the shadows from the mask returned from the mixture of Gaussians algorithm.

Denoise As implemented in [13], this component applies denoise filtering on the mask. Eliminating in this way small spots previously identified as blobs.

Morphological Transformations Finally the remaining blobs are dilated so that small gaps can be filled, resulting in a complete blob instead of several small fractions.

Finally the white blobs in the mask are analyzed to calculate its bounding box and the smallest blobs are eliminated according to a parameter (note that this parameter changes when the camera is at different height distance from the principals) leaving as a result an array of bounding boxes and a mask.

Additionally other techniques can be implemented in this module, following the interface of receiving an image for pre-processing or receiving a mask for post processing.

The result from the Blob Detection is analyzed by the Principal Identifier, here, during authentication, the template of the principal is recorded for each identifying feature, and then, in the following frames, previously authenticated identities are assigned to the corespondent blob, To verify the identity of the blob, several remote biometric factors are used and their resulting scores are averaged, and the final score is assigned to the blob with the highest score of similarity, given such score is higher than certain threshold.

The following are the selected biometrics and identification factors. For all of them the input is the bounding box, the mask, and the current frame for each blob, we are going to refer to the result of cropping the current frame around the bounding box using the mask as the 'blob frame'.

Akaze This component implements feature matching using Akaze [4] descriptors, a set of descriptors is detected in the blob frame during authentication and stored as template, and then in the following frames another set of descriptors are detected, and matched to the template, resulting in a similarity score of the amount of matched descriptors over the amount of descriptors found in the template.

Histogram During authentication a histogram of the blob frame is calculated and during the identification of principals that histogram is compared to the current blob frame's histogram, using the Bhattacharyya distance[18].

Kalman Filter Tracker A Kalman Filter models the principal movement in the real world coordinates using as the state the position and speed, and as measurement only the position. During authentication, a new tracker is created for the principal, then, in the following frames, the position of the blob is compared with the prediction of the principal's position, normalized with a distance considered as the double of the maximum allowed error distance (which means that a score of 0.5 is the maximum allowed score to say that the Kalman tracker approximately models that principal's position.

Skin Color A skin region is searched using the ranges defined by Chai and Ngan (See section 2.4.2.2) in the upper area of the blob, and then the average color detected as skin is compared with the template taken during authentication using euclidean distance, the score is normalized dividing the distance over the maximum distance possible in the skin color range.

Chapter 5

Implementation

5.1 Class structure

Figure 5.1 shows the packages conforming the implementation of the prototype, in addition to the main continuous authentication system, an initial authenticator was implemented. The system was implemented in C++11 Using OpenCV 3.2 which was compiled with support for contrib features, CUDA 8 and QT.

The package GUI implements a user interface that configures the parameters for the Experimental Environment described in section 6.1, and it shows the results in real time.

The package Scene Model (shown in figure 5.2) corresponds to the component of the same name described in section 4.1.1, This is the entry point to the system for the graphical interface, allowing the configuration of the Environment.

Camera is one of the Main classes in the system, as is the one modeling the general behavior of the computer vision analysis (apply background subtraction, identify principals) if some other behavior where to be added to the analysis, for example changing background subtraction for HOG in certain cases, then this would be a starting point in such implementation. One important characteristic to mention is that the capturing of the frames from the video stream and the

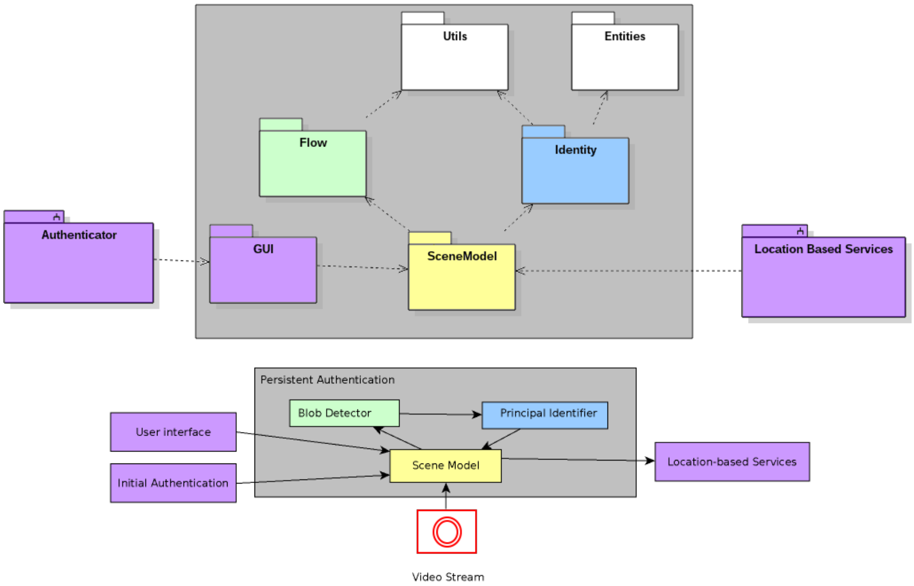

Figure 5.1: Packages in the implemented prototype matched with the designed component

actual processing is done in two different threads, this helps to maintain the analysis in real time, even if the last frame took a long time to process, the current frame would be the last one captured.

Environment is the interface of communication with external components, this class is used for maintaining the configuration of the building (zones and cameras), trigger the registration of templates and shutting down the system.

The packages Flow and Identity Implement the components Blob Detector and Principal Identifier respectively, the main classes in the package flow are displayed in figure 5.3, here the mentioned behavior of this techniques for blob detection are modeled by the abstract class Transformation, the received Mat is a OpenCV Matrix and can contain a frame or a mask in this case.

In the final version, only Histogram Equalization was excluded, this technique is suggested to reduce lighting influences in background subtraction, but it was observed that it made some dark areas even darker, which would make blob detection worse at those areas.

Package Identity is shown in figure 5.4, the Classes that implement the interface

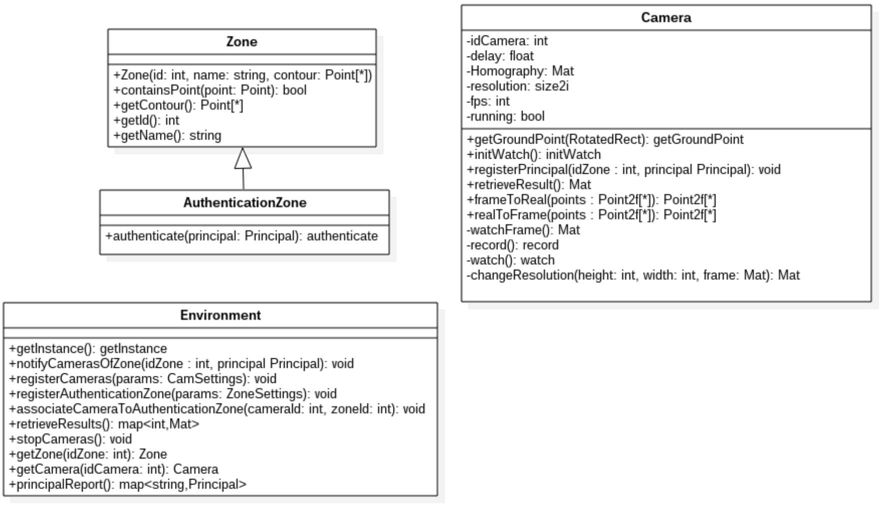
Zone
+Zone(id: int, name: string, contour: Point[*])
+containsPoint(point: Point): bool
+getContour(): Point[*]
+getId(): int
+getName(): string
AuthenticationZone
+authenticate(principal: Principal): authenticate
Camera
-idCamera: int
-delay: float
-Homography: Mat
-resolution: size2i
-fps: int
-running: bool
+getGroundPoint(RotatedRect): getGroundPoint
+initWatch(): initWatch
+registerPrincipal(idZone : int, principal Principal): void
+retrieveResult(): Mat
+frameToReal(points : Point2f[*]): Point2f[*]
+realToFrame(points : Point2f[*]): Point2f[*]
-watchFrame(): Mat
-record(): record
-watch(): watch
-changeResolution(height: int, width: int, frame: Mat): Mat
Environment
+getInstance(): getInstance
+notifyCamerasOfZone(idZone : int, principal Principal): void
+registerCameras(params: CamSettings): void
+registerAuthenticationZone(params: ZoneSettings): void
+associateCameraToAuthenticationZone(cameraId: int, zoneId: int): void
+retrieveResults(): map<int,Mat>
+stopCameras(): void
+getZone(idZone: int): Zone
+getCamera(idCamera: int): Camera
+principalReport(): map<string,Principal>

Figure 5.2: Classes in the SceneModel package

modeled by the abstract class Credential implement the principal recognition techniques, they are instantiated and then initialized with the method init (init can be different for each technique), this will record the template from the current frame. Then in the following frames the checkSimilarity method will return the similarity score and finally the confirm method is called with the result that contained the highest score (that is for the techniques that require such behavior, as the Kalman filter). In this module, IdentityAnalyser is the class iterating trough the Blobs and fusing the scores.

During development, we found that histogram of colors and Kalman Tracker gave good results regarding persistence and scalability; Color of skin was too sensitive to the light changes and the difference in score between most people is not high enough to improve identification; And both the speed and accuracy of matching Akaze features where too low to give any good result.

5.2 Calibration

During the development 6 parameters where found that have to be changed according to light conditions of the place and to the altitude of the camera. The parameters for the adaptive mixture of Gaussian, number of frames to base the background on and distance from background to consider a pixel as foreground,

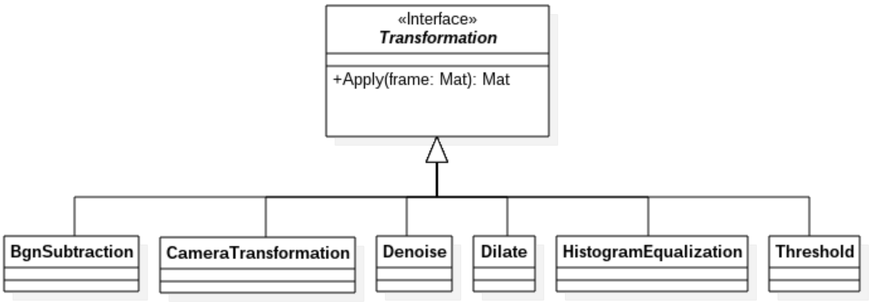

Figure 5.3: Classes in the flow package

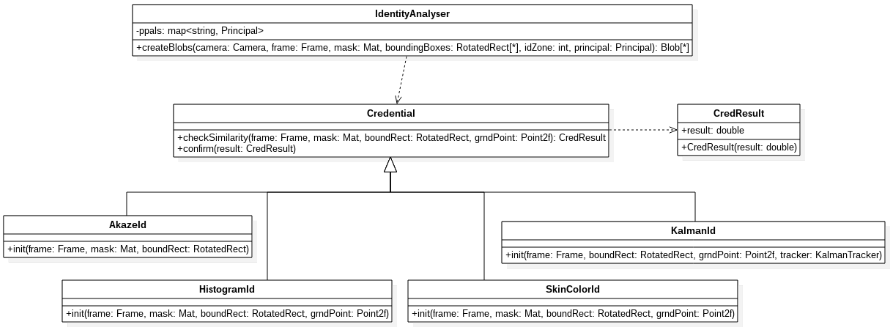
IdentityAnalyser
-ppals: map<string, Principal>
+createBlobs(camera: Camera, frame: Frame, mask: Mat, boundingBoxes: RotatedRect[*], idZone: int, principal: Principal): Blob[*]
Credential
+checkSimilarity(frame: Frame, mask: Mat, boundRect: RotatedRect, grndPoint: Point2f): CredResult
+confirm(result: CredResult)
CredResult
+result: double
+CredResult(result: double)
AkazeId
+init(frame: Frame, mask: Mat, boundRect: RotatedRect)
KalmanId
+init(frame: Frame, boundRect: RotatedRect, grndPoint: Point2f, tracker: KalmanTracker)
HistogramId
+init(frame: Frame, mask: Mat, boundRect: RotatedRect, grndPoint: Point2f)
SkinColorId
+init(frame: Frame, mask: Mat, boundRect: RotatedRect, grndPoint: Point2f)

Figure 5.4: Classes in the identity package

are susceptible to how even is the lighting conditions, and how fast do they change. The light conditions can also cause the human related blobs to have a different geometry, to adjust to such changes, the size of the dilate and erode filter have to be changed. And specially, when the camera altitude is different than 3m, the minimum area size of the blobs to be detected as human has to change, the way that area is calculated is amount of $< pixels\ in\ the\ frame >$ / $< area\ parameter >$. And changing that parameter makes you change the other parameters, as they filter small blobs to exclude from the analysis. All these parameters can be configured in the file settings/calibration.xml.

5.3 How to expand

The system was designed with 2 main points of extension. If the accuracy of the continuous authentication wants to be increased, more identifying factors can be added by making a class that implements the Credential interface and including it in the IdentityAnalyzer, including additional features can decrease the frames per second, so this has to be taken into account. Another way to increase the security level is to improve the Initial Authentication methods, multiple methods can be added or the current one could be replaced by making a class that (when the authentication is successful) invokes the method notifyCamerasOfZone of Environment, in this way the authentication of the blob found on the zone sent as parameter, is triggered and the identity sent as parameter is assigned to it.

Another point of extension is the User Interface, a different user interface could be implemented integrating the authentication methods of the specific building (using the class Environment) and also making the configuration process (see section 5.4) more user friendly. The Method in charge for authentication (notifyCamerasOfZone), receives a string identifying the principal and the configured id of the authentication zone.

5.4 Configuration process

All the configuration for the experiments setup is hard coded in the file player.cpp, here, the classes CamSettings and ZoneSettings are filled to describe the building conditions, the cameras' settings and the Cameras that are seeing the authentication zones, the following lists describes such parameters.

5.4.1 Camera Settings

As multiple cameras might want to be included to cover a large area, this configuration enables to include each camera with their own parameters.

id A unique number used to map cameras and zones.

sourcePath Path of the video, it can be a file or a stream url.

settingsFilePath Path of the file containing the camera model settings, this file can be generated with the utility program "Calibrate", application was adapted from the camera calibration code provided with the documentation [1].

homographyFilePath This file describes the homography used in the Camera class, this file can be generated with the utility program "homography", it asks for four points in the screen, and using the file cornersID.xml it calculates a homography to transform the camera coordinates to real world coordinates.

isFisheye This parameters stablishes if the camera model is using the fisheye technique described in the documentation [1], it should be true if the camera has a fisheye lens.

focalLenght This has to be manually configured to obtain a proper image, figure 5.5 shows a common example of how an image looks like if this parameter has to be increased.

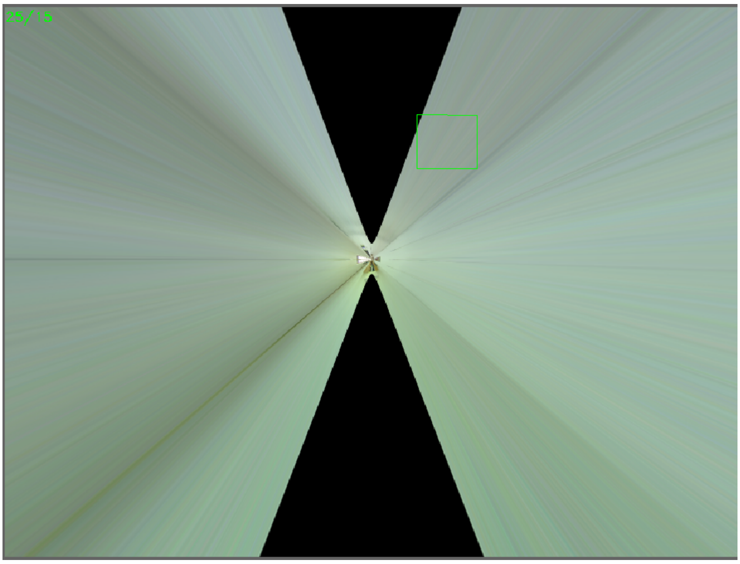

Figure 5.5: Example of a wrong Focal length, used in the camera matrix used after transforming the image with the camera calibration process.

delay In case of using de-synchronized video input between cameras, this parameter can be used to adjust the difference with respect to a camera configured with zero delay.

5.4.2 Authentication Zone settings

This configuration represents the places in the building where authentication can be done, and the configured id is the number that the authentication mechanism has to provide when authenticating a principal.

id the identifier of the zone, used to link it to a camera.

contoursFilePath A file with a series of points, describing the area, this file has to be manually configured, using real world coordinates.

Chapter 6

Evaluation

The system was installed in a residential house, and tested using several scenarios that challenge and explore its limits. this scenarios change in terms of interaction between principals and amount of people involved. The scenarios are described in section 6.2 and the results are shown and explained in section 6.3

6.1 Experiments Setup

Video cameras are usually already deployed in buildings where security is a concern, but these cameras usually don't offer high quality video, this makes, including COTS cameras very relevant for wide spread application. For the experiments we used two LevelOne FCS-3091 Cameras, they are have a 2MP color sensor with a software removable IR-filter and a fisheye lens.

For the analysis of the video feed, we used a computer with a 8 core intel core i7 cpu (2.5 Ghz), 12 GB RAM, and a NVIDIA GeForce 850M video card. The computer was connected to the cameras using a wired network with a 100Mbps switch. And the software used wide spread libraries with the implementations of the used computer vision algorithms (such as OpenCV)

Camera placement is a multi-scene and multi-camera environment. The cameras where installed in a space consisting of 3 rooms, with one camera in two of the rooms and a intersection in the cameras field of view in the other room, the cameras where installed on the ceiling at 3m height, this can be observed in detail in figure 6.1.

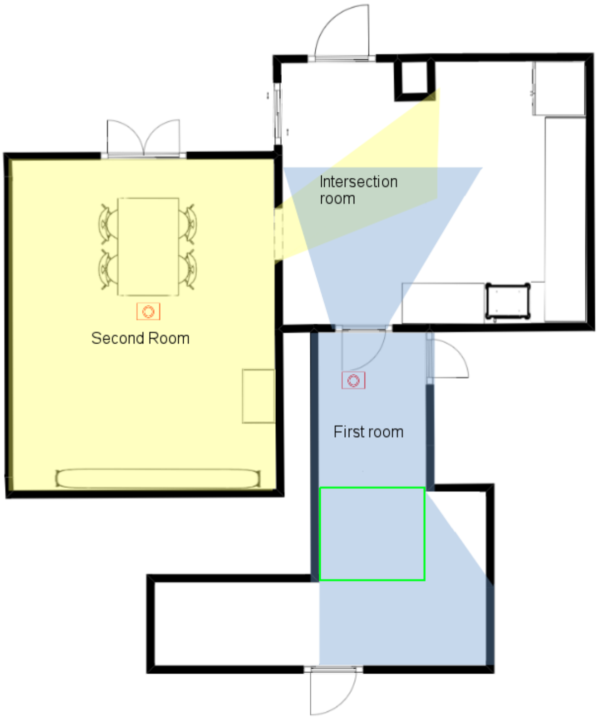

Figure 6.1: Cameras' field of view

A simple experiment involving only one principal would work as follows:

1. A person would get inside the first room (by the lower part of the image)
2. He will step in the authentication zone (delimited by the green rectangle)
3. Authenticate using an initial authentication mechanism
4. The person will walk towards the intersection room to the intersection of the fields of view of the cameras
5. He will continue to the second room, leaving the field of view of the first camera and being only perceived by the second camera.

6.2 Scenarios

The Implemented system was evaluated with a set of test scenarios divided in 3 groups, the first was designed to test the persistence of the system, the second to test the performance (in terms of speed) and the third group evaluates the impact of adding more cameras.

6.2.1 Type 1: Persistence

Four types of tests were performed for testing the persistence, types from 1 to 3 consist of two people interacting with each other, two tests are performed in each type, for the first test, the interaction occurs in the second room, and in the second test the interaction occurs in the interception room, for the four type one test challenging weaknesses in both biometrics is performed:

1 Crossing In this test the subjects cross each other in opposite directions. This test is the least problematic of the challenges, as both the color and the movement of the principals are different.

2 Handshake The subjects approach each other and shake hands for 3 seconds, then continue in opposite directions. This is expected to generate just one blob that stays static for the duration of the handshake.

3 Hug This test is similar to the handshake, but changing it for a hug to generate occlusions on top of having just one blob.

4 Similar clothes The subjects are wearing clothes with the same colors in the same proportion, in the test they cross paths in the second room.

6.2.2 Type 2: Performance

In this scenarios, multiple tests were performed for each type of test, increasing the number of subjects between tests, and the average frames per second was recorded as a result. Each type of test varies in the complexity of the interaction between subjects; in the first type the subjects are walking one behind the other with a distance of 2 meters approximately, in the second, the subjects cross each other in the second room, and in the third type they cross each other in the intersection.

6.2.3 Type 3: Horizontal Scalability

The final test simulated 3 to 6 cameras by repeating 2 to 5 times the input from the second camera. A video from test of type 2 "Simple path with 2 subjects" was used, measuring the average frames per second and the amount of successfully tracked subjects.

6.3 Results

6.3.1 Type 1 Persistence results

The results for this test type are presented as success or failure followed by explanation of the result in table 6.1, then general findings are presented.

6.3.2 General findings

A particularity of using cameras placed above the subjects is that the geometry of a person is perceived different depending on the angle between the person and the camera, or in other words, depending on the position of the person in the frame. In the tests it was observed that the camera frequently lost track of subjects when they where directly bellow of the camera, sometimes because the blob was too small to be considered, or sometimes because the difference in color was too high (as most of the body was not visible)

The intersection between the fields of view of the camera was a very challenging zone for tracking, both because of illumination differences with the first and second room, and because the principals had to change the direction of their path, this made this zone specially sensitive to low frame rates, producing losing of track for the combination of several weaknesses at the same time, and making it difficult to recover, as the tracking would expect the subjects to move in a straight line.

The system is better at dealing with full occlusions than with partial occlusions, this is because of the way the ground position is calculated, when there is an occlusion that covers more than half of the lower part of the body, the calculated position is very different than the prediction. Dealing with this is specially hard with a camera on the ceiling as the geometry of a person changes depending on the position in the frame.

Table 6.1: Results of the test scenarios for type 1

Test	Result	Explanation
Crossing 2nd room	Success	Both principals where tracked correctly, the challenge in tracking was overcame by the histogram measure.
Crossing intersection	Success	Same as before
Hanshake 2nd room	Success	After the handshake Subject A gains both identities for a few frames, then only keeps his right identity, and subject B gains her identity back.
Hanshake intersection	Failure	After the handshake Subject A gains both identities for a few frames, then only keeps his right identity, and subject B is not able to recover from the lost identity.
Hug 2nd room	Success	Both principals where tracked correctly, the histogram helped the tracking to decide which blob to choose
Hug Intersection	Failure	Same behavior of the hand shake, this is due to combined challenges in tracking for the detection of one blob and in histogram for difference in illumination from the 2 cameras
Similar clothes crossing	Failure	Principal A got assigned the two identities after the blob separation, when running this scenario with just the histogram is observed that the tracker cannot differentiate the principals, and during the blob separation the tracker lost track of principal B. When running the scenario with only the Kalman filter one of the principals lost track after the blob separation, while the other maintained his correct identity. This is because subject A's position is closer to the prediction of the Kalman Tracker for both trackers.

The biggest found problem, is that all the techniques are very sensitive to light changes, even using procedures to reduce the influence of lighting. In several of the tests (for the same subject) there where moments in which the histogram score was too high (indicating difference with the template) and this happened in the more illuminated spots. Using the color space YCrCb helped to reduce the effect in most of the areas, but bright spots would still have this problem.

This effect of the bright spots was reduced by using the Kalman Tracker, as the consistency of the movement reduced the overall score, but depending on the amount of light coming in, this errors could become more frequent and accumulate to other weaknesses (such as problems for the Kalman tracker or low frame rates in the intersection), producing lost of tracking even without interaction between blobs.

An opposite case, darker areas made the blob detection fail to detect parts of the body such as the legs or some times the face, this happened because at low illumination conditions, the color of the missing parts was similar to the color of the background.

Another problem related to illumination found in several test scenarios, was the difference in illumination between the first and the second room. Having that the authentication zone was in the first room, there was a constant difference between the histogram template and the histograms in the second room, this was not usually a problem because the difference was not too big, but it reduced the ability to recover from losing the tracking in the second room.

6.3.3 Type 2 Performance results

The results are presented in figure 6.2, increasing the number of subjects in each test measuring the average of frames per second after every subject was authenticated.

There was no difference in average frames per second between the 3 types of interaction, the only factor influencing performance was the number of subjects in the test, and this was only when they where authenticated. Running the same scenarios without authenticating the subjects had no difference in speed, indicating that the blob detection mechanism is not affected by the amount of subjects.

The results show a fast drop in speed, reaching 2 fps for 5 subjects[1], although

[1] More than 5 persons in the selected scenario would not allow normal fluid movement

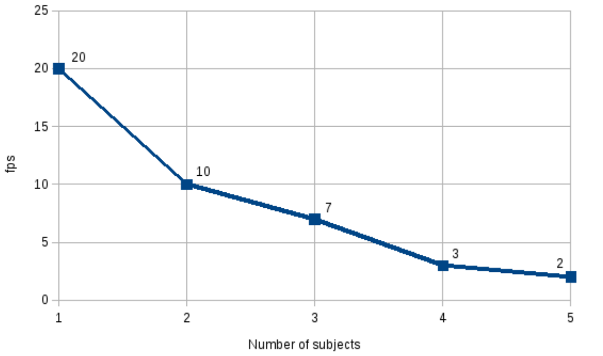

Figure 6.2: Performance results for increasing number of subjects

2 fps is still a reasonable speed for tracking, because assuming an average speed of 146.2 cm/s that would mean a change of 73.1cm between frames [6], by solving the synchronization problem the system could maintain tracking in these scenarios.

6.3.4 Type 3 Horizontal scalability results

Nine tests where executed in this test type, increasing the number of cloned second camera feed, and using the same videos every time. As a result in terms of persistence of tracking, the implemented system has no problem tracking between 1 and 3 cameras, when 4 cameras where introduced, the tracking was lost for only one of the subjects, and from 5 cameras the tracking was lost for both. This happened at different places when the subject was in the second room, in every case the tracking loss happened because the cameras where desynchronized. Even manually adjusting the delay times, some difference in timing was introduced, this could be explained by delays opening each video and differences in execution of the threads.

Differences in speed are presented in figure 6.3, showing the impact in speed induced by the amount of cameras. The figure shows a fast drop between 2 and 4 cameras, followed by a slow drop increasing the number of cameras. This is explained by the failure in tracking after 4 cameras, having 2 authenticated subjects has an impact in speed while increasing the cameras, although when the subjects are not identified, their Kalman filters are not updated leaving with less computation to be done. This means that without good synchronization

which would affect the tracking mechanism.

is difficult to analyze the performance impact, this result can be observed as a lower bound to the performance effect indicating that the result is satisfactory for less than 9 cameras.

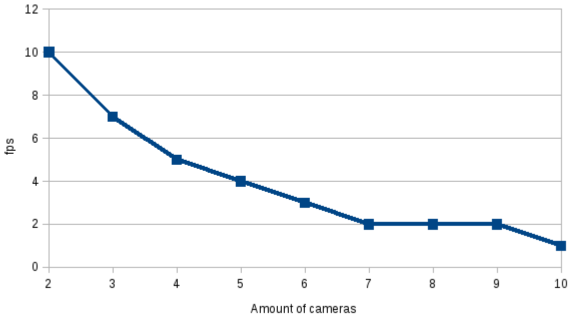

Figure 6.3: Performance results for increasing number of cameras

Chapter 7

Conclusion

This thesis main contributions where firstly, the study of new techniques for continuous authentication, namely HOG descriptors for principals detection, and skin color and Akaze features for principals identification. Secondly the identification of challenges (not discussed in previous work) for implementing Persistent Authentication, and finally the implementation of the system, made available for future studies. In the following we present the results of the implementation followed by the challenges involving multiple cameras.

7.1 System implementation

In this thesis, a Persistent Authentication system was implemented using 2 COTS cameras allowing the position of principals be connected to their identities, providing a tool for location-aware services and for exploration of biometric factors relevant to this field.

Four remote biometrics where explored: histogram of colors, direction of movement, Akaze features and skin color, finding that Akaze features and skin color where not good enough, Neither alone nor when combined with the other factors. So only histogram of colors and a Kalman filter (for movement direction) where used.

The system is able to follow a principal as he goes trough the building and crossing other principals, as long as they are wearing different clothes. When the two used biometrics where challenged the system was not able to differentiate between the principals, which may be mitigated by the introduction of an additional biometric factor.

The system maintained tracking of principals after their interaction as long as this was not added to more challenges. Several types of interaction where tested, crossing, handshakes and hugs, failing when other problems where present.

The implemented system included an example of Authentication using facial recognition, this Implementation could be replaced as described in section 5.3. The templates captured at authentication do not depend on the authentication mechanism, making the system agnostic to the implementation scheme.

7.2 Challenges involving multiple cameras

This thesis studied an a multi-camera and multi-scene environment, and this presents specific problems not existent with only one camera or scene. This section summarizes the identified problems.

The biggest problem affecting the implemented system was light influence, and some of this problems are introduce by this type of environment. When one of the scenes has a more intense light than the other, the perceived colors can vary very much, even when compared using color spaces that separate intensity in a different channel. This introduced a lot of noise in the biometrics that rely on color.

Additionally, in scenes with different intensity of light with two different cameras, the intersection will be perceived very dark in one camera and very bright in the other (due to automatic exposure in the camera) leading to errors in identification and blob detection.

However intensity is not the only quality of light that affects this type of environment, the color of light also generates differences of measured colors, different colors of illumination not only come from different types of light sources but also from the reflection in different surfaces, for example in the tested scenes one of the rooms had a wooden ceiling and wall while the others where mostly white. This problem was exaggerated when the camera adjusted automatically the white balance, but such feature can be turned off.

A different problem is the desynchronization of the video feeds, as there is nothing present in the cameras that ensure such synchronization, there can be a difference in timing between cameras. This can be addressed by generated a synchronized event in every camera, such as the firing of a flash, and using that event to add a delay in one of the videos. Such process was done manually for the test scenarios, but automating such process is possible.

When the system escalates in the number of cameras other factors add up to the desynchronization of the videos, a synchronization of the threads could solve this problem, but would reduce the speed of the system and could introduce a delay in the whole system, ending up with a desynchronization with the real world (and therefore with the principal interaction). Another solution could be implementing a more distributed approach of this design.

When tracking the position in the intersection of the fields of view of two cameras, real word coordinates have to be used to be able to connect the movement perceived in one camera to the next. Because tracking-by-detection techniques are based in image position, their application is impossible for the intersection. This was observed in the work by Ingwar [13].

Appendix A

Source of the implementation

The system source is available publicly under the Apache License 2.0 in the repository https://bitbucket.org/joramartinezc/persistentauthentication

The system was developed Using Eclipse and its usage for its compilation is recommended, a readme file is located in the root folder with more installation details.

The main project is found under the folder PersistentAuth, the Initial authenticator is contained in the project FacialRecognition in the file authenticator.cpp, The project Authenticator remote works as a trigger button for starting the initial authenticator. The project homography can be used to generate the homography for each camera and their scene.

Acronyms

BS Background Subtraction. 8–10, 46

COTS Commercially Off-The-Shelf. 2, 24, 31, 46

FNR False Negative Rate. 2, 3, 8, 46

FPR False Positive Rate. 2, 3, 8, 46

HOG Histograms of Oriented Gradients. 11, 12, 26, 27, 33, 38, 46

HSV Hue-Saturation-Value. 28, 46

IR Infra Red. 5, 31, 46

KCF Kernelized Correlation Filters. 46

KDE Kernel Density Estimation. 10, 25, 46

KLT Kanade-Lukas-Tomasi. 27, 46

PAISE Persistent Authentication In Smart Environments. 23, 46

PDF Probability Density Function. 10, 46

SIFT Scale-Invariant Feature Transform. 29, 46

SVM Support Vector Machine. 11, 12, 46

TOF Time-Of-Flight. 8, 46

Bibliography

[1] Opencv documentation, camera calibration. http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html. Accessed: 2017-06-20.

[2] Inria dataset. http://pascal.inrialpes.fr/data/human/. Accessed: 2017-06-20.

[3] Pablo Alcantarilla, Adrien Bartoli, and Andrew Davison. Kaze features. *Computer Vision–ECCV 2012*, pages 214–227, 2012.

[4] Pablo F Alcantarilla, Jesus Nuevo, and Adrien Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[5] Chiraz BenAbdelkader and Larry Davis. Estimation of anthropomeasures from a single calibrated camera. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 499–504. IEEE, 2006.

[6] Richard W Bohannon. Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants. *Age and ageing*, 26 (1):15–19, 1997.

[7] Douglas Chai and King N Ngan. Face segmentation using skin-color map in videophone applications. *IEEE Transactions on circuits and systems for video technology*, 9(4):551–564, 1999.

[8] Mark D. Corner and Brian D. Noble. Zero-interaction authentication. In *In International Conference on Mobile Computing and Networking*, pages 1–11, 2002.

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] Antitza Dantcheva, Carmelo Velardo, Angela D'angelo, and Jean-Luc Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, 51(2):739–777, 2011.

[11] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. *Computer Vision—ECCV 2000*, pages 751–767, 2000.

[12] Giovani Gomez and Eduardo Morales. Automatic feature construction and a simple rule induction algorithm for skin detection. In *Proc. of the ICML workshop on Machine Learning in Computer Vision*, volume 31, 2002.

[13] Mads I Ingwar. *Resilient Infrastructure And Building Security*. PhD thesis, Technical University of Denmark, DTU, 2014.

[14] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2nd edition, 2011. ISBN 9780470666418.

[15] Martin Kirschmeyer and Mads Syska Hansen. Persistent authentication in smart environments. Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2008. Available online, http://etd.dtu.dk/thesis/211455.

[16] Xin Li, Kejun Wang, Wei Wang, and Yang Li. A multiple object tracking method using kalman filter. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 1862–1866. IEEE, 2010.

[17] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. A survey on indoor positioning systems. In *Software, Telecommunications and Computer Networks (SoftCOM), 2014 22nd International Conference on*, pages 111–120. IEEE, 2014.

[18] Koichiro Niinuma, Unsang Park, and Anil K Jain. Soft biometric traits for continuous user authentication. *IEEE Transactions on information forensics and security*, 5(4):771–780, 2010.

[19] Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[20] SJ Shepherd. Continuous authentication by analysis of keyboard typing characteristics. 1995.

[21] Carmelo Velardo and Jean-Luc Dugelay. Weight estimation from visual body appearance. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–6. IEEE, 2010.

[22] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[23] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1505–1518, 2003.

[24] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785, 1997.

[25] Li Zhang, Bo Wu, and Ram Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[26] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.