

# An Integrated Vision-based Architecture for Home Security System

John See, *Student Member, IEEE*, and Sze-Wei Lee, *Member, IEEE*

**Abstract** — Automated security systems are a useful addition to today's home where safety is an important issue. Vision-based security systems have the advantage of being easy to set up, inexpensive and non-obtrusive. This paper proposes an integrated dual-level vision-based home security system, which consists of two subsystems – a face recognition module and a motion detection module. The primary face recognition module functions as a user authentication device. On an event of a failure in the primary system, the secondary motion detection module acts as a reliable backup to detect human-related motions within certain locations inside the home. Novel algorithms have been proposed for both subsystems. Several experiments and field tests conducted have shown good performance and feasible implementation in both subsystems<sup>1</sup>.

**Index Terms** — Home security system, face recognition, motion detection, integrated architecture.

## I. INTRODUCTION

In today's age of digital technology and intelligent systems, home automation has become one of the fastest developing application-based technologies in the world. The idea of comfortable living in home has since changed for the past decade as digital, vision and wireless technologies are integrated into it.

Intelligent homes, in simple terms, can be described as homes that are fully automated in terms of carrying out a predetermined task, providing feedback to the users, and responding accordingly to situations. In other words, it simply allows many aspects of the home system such as temperature and lighting control, network and communications, entertainment system, emergency response and security monitoring systems to be automated and controlled, both near or at a distance.

Automated security systems play an important role of providing an extra layer of security through user authentication to prevent break-ins at entry points and also to track illegal intrusions or unsolicited activities within the vicinity of the home (indoors and outdoors). There has been much research done in the design of various types of automated security

systems. Recently, Choi et al. [1] proposed a new algorithm for an acoustic intruder detection system for home security. Their algorithm estimates the variation of features of the room acoustic transfer function to detect intruders. Luo et al. [2] developed a multiple remote interface security system (MRISS) that is integrated with an intelligent security robot (ISR), security supervise computer, GSM module, RF interface and appliances control module. Hagiwara et al. [3] implemented a community security system using individually maintained home computers that are connected via the Internet. Their experimental project has been effectively implemented in the town of Kiryu, Japan.

Vision-based security systems have many advantages to consumer applications. Firstly, and most importantly, vision-based security systems are unobtrusive and user-friendly. User authentication and intruder tracking can both be performed from a distance without any human intervention. This is an important advantage as opposed to sensor-based systems that rely on contact or movement sensors or contact-based systems such as fingerprint and palmprint scan or keypad activation that require substantial amount of contact with an input device. Secondly, setup is easy and inexpensive as they only require simple low-cost vision devices of reasonable resolutions such as consumer cameras, web cameras and embedded cameras in mobile devices, computers or servers, and other peripheral devices. Recently, Zuo and de With [4] proposed a near real-time embedded face recognition system for consumer applications called *HomeFace*. The system is embedded into a smart home environment for user identification.

Many security systems are based on only a single system. In an event of system failure or intrusion of the user authentication, there is no backup system to monitor the home continually. This shortcoming can be dealt with using multiple security systems (or multi-layered security systems). However, multi-system implementations will definitely be more demanding in terms of computational cost and organization. This requires careful integration and sharing of resources. Thus, a feasible system should be effective, practical and reasonable in cost.

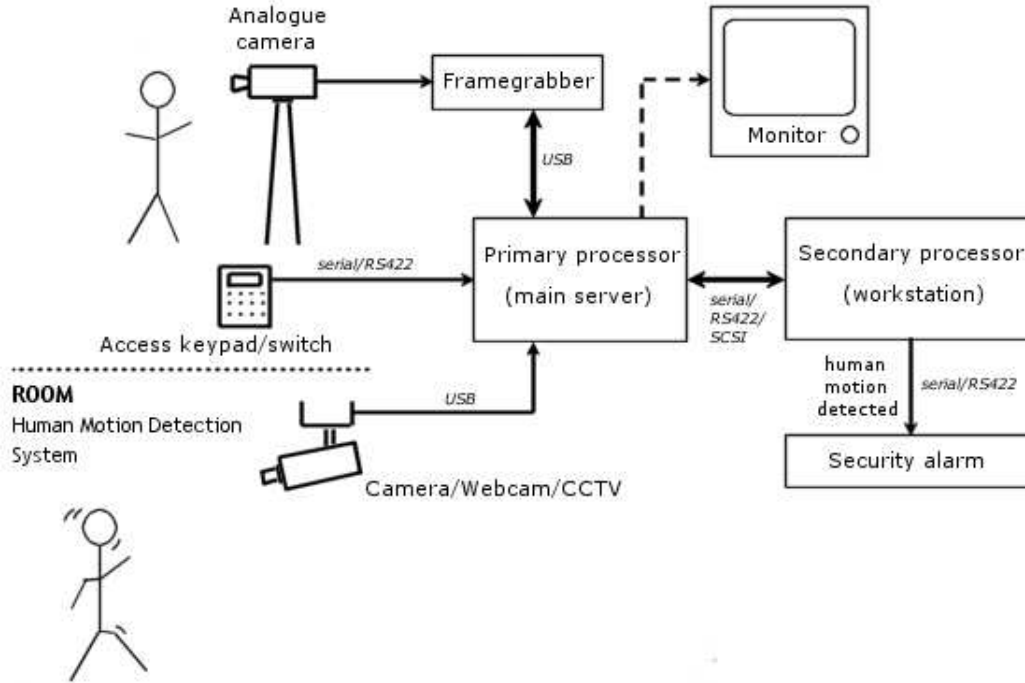
In this paper, we proposed an integrated dual-level vision-based home security system, consisting of two subsystems – a face recognition module and a motion detection module. Both subsystems work independently but are incorporated into a single automated system for practical implementation. The *primary* face recognition module works as a user authentication device whereas the *secondary* motion detection module scans for human-related movements within certain

<sup>1</sup> This work was supported in part by the IRPA Grant, "The Development of an Automated Home Security System" under IRPA 04-99-01-0068-EA064.

J. See is with the Faculty of Information Technology, Multimedia University, 47100 Cyberjaya, Selangor, Malaysia (email: johnsee@mmu.edu.my).

S.-W. Lee is with the Faculty of Engineering, Multimedia University, 47100 Cyberjaya, Selangor, Malaysia (e-mail: swlee@mmu.edu.my).

**DOOR/ENTRANCE Face Recognition System**



**Fig. 1. Block diagram of proposed integrated home security system architecture**

enclosed areas inside the home while the occupants are away. On an event of a failure in the primary system or intrusions through other means (windows, roof, etc.), the secondary system acts as a reliable backup. In both systems, we have incorporated novel schemes that are robust and efficient. Promising results were reported in both accuracy rates and computation time.

The organization of this paper is as follows. In section II, the integrated architecture of the system is further elaborated. In section III, the proposed algorithms of both subsystems are described concisely, and experimental results are discussed in section IV. Finally, section V will give the conclusion and future directions.

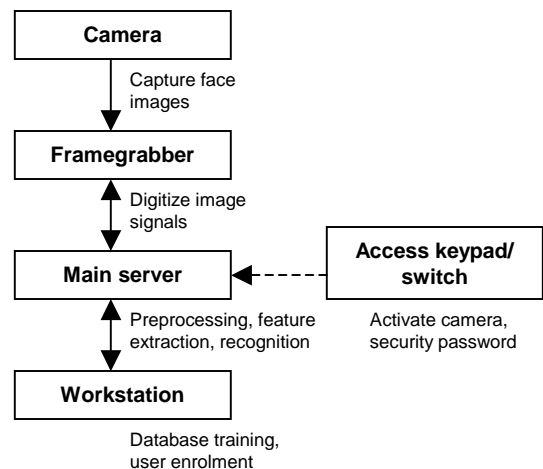
## II. INTEGRATION ARCHITECTURE

The proposed integration architecture incorporates two subsystems – face recognition system and motion detection system, into a single automated architecture for practical implementation in intelligent home environments. Fig. 1 above shows a block diagram of the proposed system architecture and its setup and connectivities. Both modules work independently and parallelly but share computational resources.

### A. Face Recognition Module

This module is the *primary* security system that functions at the user authentication level, which aims at granting entry into the home to authorized people. Likes most face recognition implementations, this entry-level system

consists of a simple analogue camera, which is mounted at the door or entrance to the home. The camera may operate in both tracking and non-tracking fashion to allow flexibility to users. Thus, it is recommended that an access keypad or switch be used to activate the camera to capture the image of the person’s face on an event of necessity. The camera and framegrabber are connected to the main server where the preprocessing, feature extraction and recognition tasks are performed. Though this task is not computationally expensive, the secondary dedicated processor (workstation) can be used to aid the training and enrolment of new users. Fig. 2 shows the process flow of the face recognition module.



**Fig. 2. Face recognition module process flow**

## B. Motion Detection Module

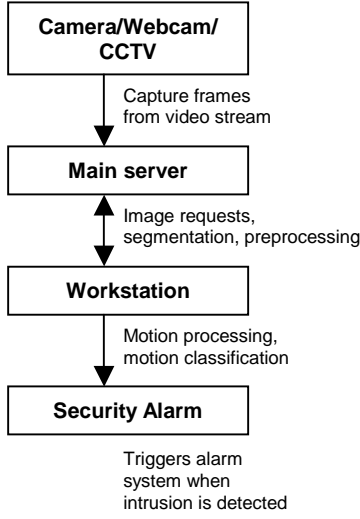


Fig. 3. Motion detection module process flow

This module acts as the *secondary* security system, and it is intended as a backup measure to track down a moving intruder inside a monitored area in the home. Fig. 3 shows the process flow of the motion detection module. This system consists of a continuous-tracking camera (such as web camera or CCTV), preferably running at a high frame rate (20-30 *fps*) with a reasonable low resolution (240x300 pixels or smaller). The camera is connected to the main server via USB or any serial connection. The secondary processor (workstation) takes charge of the computationally heavy portion of the detection process to lighten the computation load of the main server.

The main server obtains the acquired frames from the camera and carries out preprocessing and segmentation on the images. If the detection is positive, then the images are sent to the workstation for further intensive processing while the main server resumes the acquisition of the next frame in sequence. In an event where the detected motion is classified as human motion, the security alarm system will be triggered to indicate an intrusion in the home.

## III. PROPOSED SUBSYSTEM ARCHITECTURE

In our work, we have implemented novel schemes for both subsystems that are robust and effective, yet low in computational cost.

### A. Face Recognition System

In consumer applications, face recognition has attracted much commercialization interest from many companies worldwide [5], [6]. The advantages of vision-based authentication as previously mentioned, have given face recognition an upper hand over contact-based biometrics such as fingerprint and iris recognition. As a result, there has been enormous interest in face recognition work in the recent years [7]. However, face recognition for real-life, real-time applications remains a challenging problem today.

In a consumer environment, a face recognition system should be 1) effective in coping with various operating environments (illumination, ambience) and facial constraints (pose, expression, age), 2) efficient in processing, and 3) inexpensive in hardware/software cost.

In this paper, we proposed a face recognition system that incorporates a novel personalized feature fusion method with a multi-tier classification scheme. This *conditional cascaded classification* is able to improve success rate while maintaining efficiency and computational cost at an acceptable level. As shown in Fig. 4, the algorithm is subdivided into three main tasks as in most face recognition schemes – the preprocessing, feature extraction and recognition (or classification).

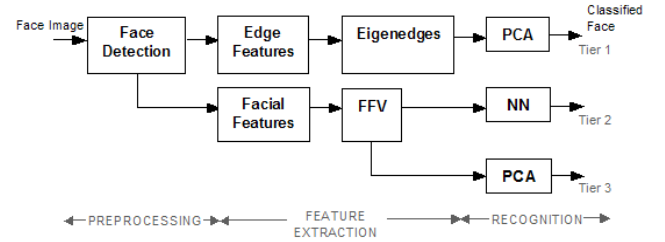


Fig. 4. Face recognition algorithm architecture

### 1) Face Detection

Firstly, the face is detected from the captured image by performing preprocessing steps such as image normalization, eye localization, and face detection. We employ a simple template matching method used by Brunelli and Poggio [8], by means of a normalized cross-correlation coefficient, which can be defined as

$$C_N(y) = \frac{\langle I_T T \rangle - \langle I_T \rangle \langle T \rangle}{\sigma(I_T) \sigma(T)} \quad (1)$$

where  $I_T$  is the patch of image  $I$  which must be matched to template  $T$ ,  $\langle \rangle$  is the average operator,  $I_T T$  represents the pixel-by-pixel product, and  $\sigma$  is the standard deviation over the area being matched.

Psychological studies often show that the eye region contains much information according to cue saliency assessments. In an assessment on front-view face fragmentation, the eye region was the overwhelming choice above the mouth and nose regions. We used an averaged eye window as the template, and further rescaling by a magnification factor of 1.2 to 0.8 (on steps of 0.1) was performed to obtain a set of five templates. Two additional matching criterion were used to enhance the robustness of location selection – 1) *maximum value of the normalized correlation values*, which considers the most correlated position among all template scales; and 2) *upper and lower bound region sum of intensities*, which considers the ratio of the sum of intensity values of the region above the selected location to the region below it. This is used to eliminate the possibility of erroneous detection of the eyebrows instead of the eyes.

From the detected eye regions, the eye centers are then easily determined using simple iterative thresholding. The eye centers are used to estimate the *inner face* window anthropometrically. The inner face is defined as the face region that excludes the hair, ears and chin features. To accomplish that, a mouth region point  $(x_{mr}, y_{mr})$  is estimated anthropometrically by approximating the length from the midpoint between both eye centers  $(x_{mid}, y_{mid})$  to  $(x_{mr}, y_{mr})$  as

$$l_2 = 1.2(l_1) \quad (2)$$

where  $l_1$  is the distance between both eye centers. All three points –  $(x_{mr}, y_{mr})$  and both eye center points, are used to interpolate the inner face window. Through this process, the inner face windows are properly normalized based on the eye locations.

## 2) Feature Extraction

The second stage deals with feature extraction where facial information cues are extracted from the localized face region. On the selection of features, Zhao et al. [7] showed that both holistic and local features are vital for feature extraction in face recognition, and it was concluded that both global description and dominant features of a face have different importance and contributions. In our proposed three-tier scheme, we used both edge features (for tier 1) and facial features (for tiers 2 and 3) to maximize the importance of both holistic and local features.

In tier 1, we used a fast parameterized fuzzy-based edge detector [9] to derive the face edges. This fuzzy edge detector uses both local and global nonlinear properties, which enables edges to be classified according to their significance (important and trivial) and strength (strong and weak). Firstly, a histogram-based fuzzy membership function is used to represent pixels in the fuzzy domain. Then, a modified nonlinear contrast intensification function,

$$\mu'(m, n) = NINT[\mu_{mn}] = \frac{1}{1 + \exp[-t(\mu_{mn} - x_c)]} \quad (3)$$

was applied to adjust every  $(m, n)$ th pixel. Here,  $t$  is the intensification operator and  $x_c$  is the crossover point of the membership function. Finally, the fuzzy parameterized Gaussian edge detector,

$$\eta(m, n) = e^{-\sum_i \sum_j \left( \frac{|\mu^{(m+i, n+j)} - \mu^{(m, n)}|}{f_g} \right)^\alpha} \quad (4)$$

is applied to filter out the edge features.  $\mu'(m, n)$  is the membership value of central pixel of the mask at location,  $i, j \in [-(w-1)/2, (w-1)/2]$ , and  $w \times w$  is the size of the edge detector mask. Parameters  $\alpha$  and  $f_g$  are both tunable fuzzifiers representing edge detail and edge strength

respectively. In our experiments, we used preset values of  $\alpha = 3, f_g = 50$  and  $w = 3$ .

The extracted face edges are then used to form *eigenedges*, an eigenface representation [10] of the edges. This representation is also commonly known as Principal Component Analysis (PCA). All edge images are normalized to  $70 \times 57$  pixels before undergoing PCA representation.

For tiers 2 and 3, more analytical features consisting of a set of fiducial points and their geometric measures were used. The first step towards the extraction of these facial features involved estimating the region-of-interest (ROI) for both nose and mouth regions. Two measures were used to determine both ROIs – 1) Prominence of integral projections, and 2) location confidence rating (LCR).

Horizontal integral projections can be extremely effective in determining the position of features, provided that the ROI for the feature region is well-located. Here, the prominence of integral projections defines the importance of local maximas and local minimas of the horizontal integral projection of the estimated ROIs. The prominence of the nose region is taken as

$$P_{nose}(u) = \frac{H_{peak}(u)}{\max\{H_{peak}(u)\}} \quad (5)$$

whereas the prominence of the mouth region is taken as

$$P_{mouth}(v) = \frac{\min\{H_{valley}(v)\}}{H_{valley}(v)} \quad (6)$$

for  $u$  local maximas and  $v$  local minimas, where  $H_{peak}$  denotes local maximas,  $H_{valley}$  denotes local minimas, and  $\{P_{peak}(u), P_{valley}(v)\} \in [0, 1]$ .

Location Confidence Rating (LCR) evaluates the certainty of a feature region being in an estimated location (from local maximas and local minimas), and it can be simply computed by the following equations:

$$LCR_{nose}(u) = P_{nose}(u) * RD_{nose}(u) \quad (7)$$

$$LCR_{mouth}(v) = P_{mouth}(v) * RD_{mouth}(v) \quad (8)$$

where  $RD_{nose}$  and  $RD_{mouth}$  are the relative distance from the estimated feature position for nose and mouth regions respectively.

After locating the respective region of interests (ROIs), both nostril and mouth corner positions are easily determined using iterative thresholding method, similar to that used to determine eye center positions. In addition to the facial geometric points, the edge density is also a useful feature to measure the concentration of edge pixels in the edge image. It can be denoted as



Fig. 5. (left to right): Sample test image; Detected inner face region; Extracted facial feature points and geometric measures; Normalized and cropped face; Face edges extracted from parameterized fuzzy edge detector.

$$\kappa = \frac{\sum \{\eta(m,n) | \eta(m,n) \neq 1\}}{MN} \quad (9)$$

where the image size is  $M \times N$  pixels and  $m \in [1, M]$ ,  $n \in [1, N]$ .

Finally, a Facial Feature Vector (FFV), consisting of four extracted facial feature distances and the edge density measure is formulated as:

$$FFV = [d_n \quad d_{en} \quad d_m \quad d_{nm} \quad \kappa] \quad (10)$$

where  $d_n$  is the distance between the nostrils,  $d_m$  is the distance between the mouth corners,  $d_{en}$  is the distance from the midpoint of the eyes to the midpoint of the nostrils, and  $d_{nm}$  is the distance from the midpoint of the nostrils to the midpoint of the mouth corners.

### 3) Classification

In the third stage, classification is performed using PCA (in tiers 1 and 3) and  $k$ -nearest neighbor ( $k$ NN) method (in tier 2).

For the PCA classification in tier 1, the most significant 20% of the total eigenvectors were taken to represent the edge features. The Mahalanobis distance classifier is used for both PCA classifications. As for  $k$ NN classification, we take  $k=1$  to only consider the closest matching face class to the input face image.

The conditional cascaded classification scheme allows the system to use an alternative feature or classification method, or both to re-classify the detected face. For simplicity, we proposed a three-tier system though it is possible to implement more tiers to further improve recognition success rate. When matching in tier 1 fails, the classification process is repeated in tier 2 using the Facial Feature Vector (FFV). Similarly, failure in tier 2 in turn causes the classification to repeat again using the PCA feature representation on the FFV in tier 3 instead of the earlier NN classifier.

In terms of time-complexity, the tiers were designed in the order of decreasing computation cost (tier 1 the highest cost, and tier 3 the least cost). This approach improves the recognition success rate while preserving a low complexity system as subsequent tiers would have lesser significant effect on the overall computation cost. In a way, the proposed system also establishes an efficient mechanism

that utilises both global and local features of a face image as identification cues.

### B. Human Motion Detection System

Human motion detection can be defined as the process of recognizing a human being regardless of its identity, based on certain characteristic patterns or features of the exhibited motion. In many computer vision problems, the detection of human motion may often be used as a sub-algorithm of more high-level problems such as head tracking and gait recognition. However, in recent years, it has also begun to find its role in consumer applications such as home security [3], surveillance control [11] and intrusion detection [1].

The proposed human motion detection system uses a novel fuzzy rule-based classification scheme to identify the presence of human motion based on moving blob regions in a captured video stream [12]. The algorithm can be divided into five main tasks (in order of flow) – image acquisition, image segmentation, blob identification, characteristic extraction and motion classification. Fig. 6 shows the proposed human motion detection algorithm flow.

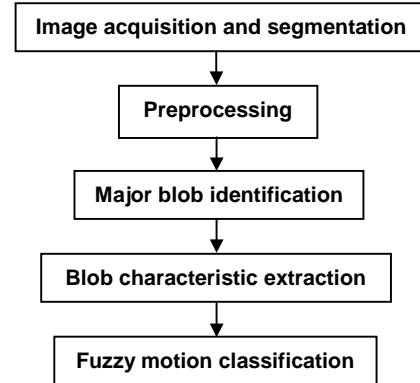


Fig. 6. Human motion detection algorithm flow

In the first step, the continuous video stream is subsampled at a certain frame rate to acquire individual frames for further processing. Next, all possible motion are segmented from these frames using adjacent frame subtraction or background subtraction depending on video frame rate.

Adjacent frame subtraction simply produces the difference image,  $D(x,y,t)$  between an input frame,  $F(x,y,t)$  and the next successive frame,  $F(x,y,t+c)$  after a fixed time interval  $c$ :

$$D(x, y, t) = abs\{F(x, y, t) - F(x, y, t + c)\} \quad (11)$$

This is an appropriate technique for high frame rate sequences, where small changes of motions are often traceable. Meanwhile, background subtraction is used for low frame rate sequences where the change of motion is larger in successive acquired frames. It is simply denoted as

$$D(x, y, t) = abs\{F(x, y, t) - B(x, y, t_0)\} \quad (12)$$

where  $B(x, y, t)$  is the background image updated at time  $t_0$ . This technique requires constant updating of the background to cope with changing environment factors such as lighting and moved objects. The choice of frame segmentation technique is often crucial to obtain an accurate region of motion. Optionally, Gaussian spatial filtering can be applied to the acquired frames to remove motion noise and changes in illumination before temporal differencing is performed. Then, a *motion image* is produced by thresholding the difference image  $D(x, y, t)$  by a certain threshold level  $\lambda_D$ .

The second stage involves some preprocessing tasks that need to be implemented to prepare the motion image. The segmented motion image may contained many *blobs*, or regions of detected motion. Before the blobs can be labeled, holes and gaps within the motion image can be closed using morphological operators such as closing and flood-fill.

In the third stage, blobs that are highly potential to be considered human motion, or *major blobs*, are identified through a selection process. Selection criteria is based on camera projection measurements and anthropometric estimations of human body area. Assuming that the monitored scene length is perpendicular to the camera projection axis, the estimated area of the human body,  $A_{Hp}$  can be projected as

$$A_{Hp} = \frac{A_{Hm} D_{Sp}^2}{D_{Sm}^2} \quad (13)$$

in pixel terms, where  $A_{Hm}$  is the average cross-sectional area of a human body,  $D_{Sm}$  is the real scene length, and  $D_{Sp}$  is the equivalent width of the input image in pixels.

Given that  $\rho$  is the major blob cutoff level, major blobs can be determined through the following decision rules:

$$\text{If } Area(blob(i)) \leq \rho D_{Sp} \text{ where } \rho=0.1-0.3, \quad (14)$$

**then**  $blob(i)$  is discarded,

$$\text{If } Area(blob(i)) > A_{Hp} / 2, \quad (15)$$

**then**  $blob(i)$  is chosen as  $MB(j)$ ,  
for the  $i$ th blob and  $j$ th major blob,  $MB(j)$ .

The next step involves the extraction of three characteristics from each major blob – motion vector, texture weight and ellipse coverage. To fully utilize the

determined motion blobs, both motion vector and texture weight characteristics are obtained by performing motion estimation using 2-D motion blobs as proposed by Shio and Sklansky [13], which was adapted from Horn's optical flow. In this technique, a local motion vector is estimated from each successive pair of frames  $F(x, y, t)$  and  $F(x, y, t + c)$  based on a quasi-cross-correlation within local regions of the frames with a  $W$ -by- $W$  aperture (window) size.  $\vec{\delta}_x$  and  $\vec{\delta}_y$  are the discrete vector functions of spatial coordinates  $x$  and  $y$  in the 2-D vector space, and  $t$  denotes time.

$$\vec{\delta} = (\vec{\delta}_x, \vec{\delta}_y) = \left\{ (m, n) \mid \min_{m, n \in \{-l, \dots, \pm l\}} \{C(m, n)\} \right\} \quad (16)$$

where

$$C(m, n) = \sum_{x=-1}^1 \sum_{y=-1}^1 |F(x, y, t + c) - F(x + m, j + n, t)| \quad (17)$$

and the estimated motion direction is restricted to an integer value between  $-l$  and  $l$  for both  $m$  and  $n$ . In this computation, we take  $l=1$  to represent 9 possible motion directions, and an aperture size of  $W=3$  to keep the overall computation load manageable. Thus, the local motion vector of the  $j$ th major blob in frame  $F(x, y, t)$  during instance  $t$  can be computed as

$$\vec{\delta}_j = -\frac{1}{P_j} \left( \sum_{p_j=1}^{P_j} (\vec{\delta}_x)_{p_j}, \sum_{p_j=1}^{P_j} (\vec{\delta}_y)_{p_j} \right) \quad (18)$$

where  $P_j$  is the total number of pixels of  $j$ th major blob.

The second characteristic, texture weight is defined as the average gray scale value of each  $j$ th major blob,

$$\zeta_j = \frac{\sum_{p_j=1}^{P_j} (F(x, y, t))_{p_j}}{P_j} \quad (19)$$

This attribute is a good cue as it is often consistent for a detected object that moves across the monitored scene.

The third characteristic, ellipse coverage is computed using a simple brute force ellipse fitting to accommodate variations in gait postures. This brute force method iterates through a number of possible combination of various scales and pose tilt angles. Ellipse coverage is simply defined as

$$\%CA = \frac{\text{Area of major blob covered by ellipse}}{\text{Area of ellipse}} \times 100\% \quad (20)$$

Finally, these three characteristics – motion vector distance,  $X_1$ , texture weight change,  $X_2$ , and ellipse coverage area,  $X_3$ , are fed into a fuzzy rule base system, which consists of a set of 5 fuzzy rules,  $R=\{R_1, R_2, R_3, R_4, R_5\}$ . The classification output,  $Y$ , is the confidence value of classifying the major blob as a human or non-human

motion:

- Rule R<sub>1</sub>** : If  $X_1$  is NEAR and  $X_2$  is SMALL and  $X_3$  is UNFIT  
then  $Y$  is REJECT Else
- Rule R<sub>2</sub>** : If  $X_1$  is NEAR and  $X_2$  is BIG and  $X_3$  is UNFIT  
then  $Y$  is REJECT Else
- Rule R<sub>3</sub>** : If  $X_1$  is FAR and  $X_2$  is SMALL and  $X_3$  is UNFIT  
then  $Y$  is REJECT Else
- Rule R<sub>4</sub>** : If  $X_1$  is FAR and  $X_2$  is BIG and  $X_3$  is UNFIT  
then  $Y$  is REJECT Else
- Rule R<sub>5</sub>** : If  $X_1$  is NEAR and  $X_2$  is SMALL and  $X_3$  is FIT  
then  $Y$  is accept (21)

Trapezoidal membership function is used in all fuzzification of inputs and the fuzzy rules are represented by the Mamdani-min (minimum) implication operator. The output of the aggregation process is defuzzified using the centroid defuzzification method to obtain the final confidence level. In our implementation, a strict confidence level of 90% is used to evaluate the output of the fuzzy classification. For each  $j$ th major blob,  $MB(j)$

- If  $Y_j > 0.9$   
then  $MB(j)$  is classified as a human motion (22)

Despite the seemingly complex nature of the algorithm, the overall computation speed is very fast and it is able to achieve real-time speeds (as will be discussed in the section IV). The human motion detection system may need to process detected motion while simultaneously maintaining continuous image acquisition from the camera. Here, distributed processing architecture is able to overcome this runtime overloading efficiently. This architecture is also essential in overcoming the inadequacy of a single processor in handling both system overhead tasks (interface, file organization, log history) and also process tasks.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results of both systems will be discussed separately. All experiments were implemented on Intel Pentium IV 2.63 GHz, 504 MB RAM machines.

##### A. Face Recognition System

The proposed face recognition system is tested with a widely used face dataset. Popular datasets are principal benchmark test sets for many established face recognition algorithms. Moreover, they also have a consistent compilation of comparative results between face recognition algorithms. In future work, we plan to conduct our experiments with proper enrolment, training and probing procedures using larger data sets.

The Olivetti Research Laboratory (ORL) Database of Faces [14] was used in our comparative experiments. Originally, the database contains 10 different images of 40 distinct persons – a total of 400 unique images altogether.

All images were taken in an upright, near-frontal pose. For some subjects, the images were taken at different times (spanning 2 years period) with variations in facial expression (open/close eyes, smiling/non-smiling), facial details (glasses/no glasses), scale (up to  $\pm 10\%$ ), face pose/orientation (up to 20 degrees of lateral tilt), and illumination condition (slightly dark/bright) between images of the same subject. The resolution of each image is  $112 \times 92$  pixels, with 256 grey levels (8-bit) per pixel.

In our experiments, we reduced the original ORL data set to a more compact 200 images, by randomly selecting 25 subjects from the initial 40 subjects, as enrolled subjects. The training set comprised of 5 images per subject and the remaining 5 images from each subject were used as part of the input test set. This is the standard partitioning method employed by most known algorithms that use the ORL database. Another 75 test images were then randomly selected from any of the unused images from the remaining 15 subjects. Thus, the final input test set is a combination of both known and unknown subjects.

In the performance evaluation, two main criteria were considered – the recognition rate and computational time. Common recognition performance measures were used — False Acceptance Rate (FAR), False Rejection Rate (FRR) and Overall Recognition Rate (ORR).

$$FAR = \frac{\text{Number of falsely accepted imposters } n_a}{\text{Number of imposter attempts/accesses } N_u} \times 100\% \quad (23)$$

$$FRR = \frac{\text{Number of falsely rejected enrollees } n_r}{\text{Number of enrollee attempts/accesses } N_k} \times 100\% \quad (24)$$

$$ORR = \left( 1 - \frac{\text{Falsely accepted users} + \text{Falsely rejected users}}{\text{Total number of tested images}} \right) \times 100\% \\ = \left( 1 - \left( \frac{n_a + n_r}{N_u + N_k} \right) \right) \times 100\% \quad (25)$$

In a real-time face recognition system, the overall computation time should only consists of the preprocessing and feature extraction time, and classification time. The training process is usually only carried out whenever there are new enrolments or new image updates.

Table I and Table II shows the recognition and computation performance of the implemented face recognition system. The proposed algorithm obtained a relatively overall good recognition rate (ORR) of 92% and more importantly, it achieved a very low computation time (training time of approximately 5 seconds and classification time of just half a second). In a real life scenario, it would take about 2.6 seconds to identify an individual.

$$\text{Real-time computation duration (RCD)} \\ = \text{Pre-processing \& feature extraction time} + \text{classification time} \quad (26) \\ = 2.5941 \text{ seconds}$$

**TABLE I**  
**FACE RECOGNITION RESULTS: RECOGNITION PERFORMANCE**

| Recognition Performance Measure | FAR  | FRR  | ORR   |
|---------------------------------|------|------|-------|
| Results (%)                     | 6.67 | 8.80 | 92.00 |

**TABLE II**  
**FACE RECOGNITION RESULTS: COMPUTATION PERFORMANCE**

| Computation Time                         | Results (s) |
|--|-------------|
| Pre-processing & feature extraction time | 2.1379      |
| Training time                            | 4.9050      |
| Classification time                      | 0.4562      |

**TABLE III**  
**ALGORITHM PERFORMANCE FOR DIFFERENT COMBINATION OF TIERS**

| Tier Combination     | Method  | FAR (%) | FRR (%) | ORR (%) | RCD(s) |
|----------------------|---------|---------|---------|---------|--------|
| Tier 1               | EE+PCA  | 6.67    | 12.8    | 89.5    | 2.2360 |
| Tier 1 + Tier 2      | FFV+NN  | 6.67    | 9.6     | 91.5    | 2.5151 |
| Tier 1+Tier 2+Tier 3 | FFV+PCA | 6.67    | 8.8     | 92.0    | 2.5941 |

**TABLE IV**  
**PERFORMANCE COMPARISON BETWEEN PROPOSED ALGORITHM AND 15 OTHER ALGORITHMS**

| #  | Method  | Training Time | Classification Time | Recognition Rate (%) |
|----|---|---------------|---------------------|----------------------|
| 1  | Eigenface [10] [15]   | N/A           | N/A                 | 90.0                 |
| 2  | Top-down HMM + grey tone features [16]                                  | N/A           | 25 s                | 87.0                 |
| 3  | Pseudo-2D HMM + grey tone features [15]                                 | N/A           | 4 min               | 94.5                 |
| 4  | PDBNN [17]  | 20 min        | < 0.1 s             | 96.0                 |
| 5  | SOM + CN [18]   | 4 hr          | < 0.5 s             | 96.2                 |
| 6  | Elastic matching [19]   | N/A           | N/A                 | 80.0                 |
| 7  | 2D-DCT + Adaptive metric NN [20]  | 2.95 min      | 0.1 s               | 97.6                 |
| 8  | Top-down HMM + 2D DCT coefficient [21]                                  | N/A           | 2.5 s               | 84.0                 |
| 9  | Ergodic HMM + DCT coefficient [22]                                      | 23.5 s        | 3.5 s               | 99.5                 |
| 10 | Continuous n-tuple (16-level quantization) [23]                         | 2 min         | 0.013 s             | 96.4                 |
| 11 | Point matching and correlation  | N/A           | 4-6 s               | 84.0                 |
| 12 | Fractal transformation [25]   | 61 s          | 3.23 s              | 98.3                 |
| 13 | Moving window classifier [26]   | N/A           | N/A                 | 96.9                 |
| 14 | ICA [27]  | 66 min        | 0.6 s               | 85.0                 |
| 15 | Gabor filters + rank correlation [28]                                   | 35 min        | 1 s                 | 91.5                 |
|    | <b>Feature fusion (Eigenedges + FFV) with multi-tier classification</b> | 4.905 s       | 0.4562 s            | 92.0                 |

The algorithms above are listed in order of oldest to newest.

Table III shows that the recognition (or reciprocally, error rates) improve steadily with very minute increase of computation time (RCD). The architecture of the multi-tier classification structure allows subsequently added tiers to be

computationally less expensive than previous tiers and this re-classification is able to improve performance rates.

Performance comparison was also conducted between the proposed algorithm and 15 widely known face recognition algorithms [10], [15]-[28] that used the ORL database in their experiments. These algorithms spanned a large period of time (1994 to 2002) and a variety of methods (eigenfaces, neural networks, statistical models). With a recognition rate of 92%, the proposed algorithm fared reasonably well in mid-position, ranking 9th out of 16 algorithms in total. In terms of computation time (training and classification time) and algorithm complexity, the proposed algorithm is far superior compared to all 15 algorithms. Table IV shows the overall performance of the proposed algorithm in comparison with the 15 algorithms.

### B. Human Motion Detection System

The human motion detection system was tested in two experimental settings – the database test and the field test.

Since most gait databases available for research are aimed towards gait recognition work, we find little relevance to conduct our experiments entirely based on these databases. In the database test, a compound database consisting of the combination of two different data sets was constructed. Database I consists of human movements in various gait poses, non-human object motions and a combination of them, obtained from our motion capture system. Sequences were captured on a Logitech QuickCam Express with a resolution of 240×320 pixels on a field view of approximately 2.5 m. Database II consists of sequences taken from the University of Southampton Human ID Gait Database [29] in three different camera angles (normal, oblique, and normal-elevated tracks) of a walking subject in various conditions (type of clothing, objects carried, walking speed). Sequences in Database I are of low frame rate whereas sequences in Database II are of high frame rate. In total, 25 sequences were used from each database.

In the second setting, a real-time field test experiment was conducted in a simple laboratory room to detect human motion. Similar settings as that used earlier to acquire sequences for Database I, were also used in this experiment.

As the algorithm contains many parameters and environment settings, we have identified a probable range of values through trial-and-error in experiments. Table V shows the good estimations of the algorithm parameters.

**TABLE V**  
**PROBABLE RANGE OF VALUES FOR ALGORITHM PARAMETERS**

| Computation Time                                       | Results (s)                                    |
|--|--|
| Motion image threshold, $\lambda_D$                    | 0.05 – 0.2 (>0.1 for low frame rate sequences) |
| Average cross-sectional area of a human body, $A_{Hm}$ | 0.35 – 0.5                                     |
| Major blob cutoff level, $\rho$                        | 0.1 (10%)                                      |



Fig. 7. Sample images of sequences taken from Database I and II.

In a preliminary motion-free test, the system took 616 seconds to process 1,000 image frames (equivalent to approximately  $1.62 \text{ fps}^2$ ). In order to continuously run frame acquisition at a short capture interval of about 0.6 seconds, motion processing is performed by a secondary dedicated workstation to maintain real-time efficiency. With this simple distributed processing, a successful detection task only took about 2 to 4 seconds on average.

Performance of the system is evaluated by the detection rate,  $DR$ , which is taken as

$$DR = \frac{\text{Persons detected} - \text{FRP} - \text{FDO}}{\text{Persons counted}} \times 100\% \quad (27)$$

where FRP denotes *falsely rejected persons* and FDO denotes *falsely detected objects*.

TABLE VI  
HUMAN MOTION DETECTION RESULTS: DETECTION RATE BASED ON EXPERIMENTS

| Experiment                                  | Correct Detection | Incorrect Detection | Detection Rate (%) |
|---|-------------------|---------------------|--------------------|
| Database Test<br>(Database I + Database II) | 46                | 4                   | 92.00              |
| Real-time Field Test                        | 44                | 3                   | 93.75              |

For the database test, an overall detection rate (combination of both Database I and II) of 92% was achieved. The real-time field test was left to run for a duration of about 2 hours and it achieved a good detection rate of 93.6%, which is better than the database test result. Table VI shows a summary of the results.

In comparison, earlier predecessor techniques using multi-feature classification metric [30] and fuzzy rule-based reasoning of motion cues [31] only yield accuracy rates of 82.8% and 92.6% respectively. However, the recent *Concept Coding* technology by Video IQ [32] have shown vast improvement of accuracy rates of up to 95% using highly-complex cues such as shape, colour and behavioral patterns. Fuzzy and statistical approaches have shown tremendous capability in improving the robustness of decision making process. Thus, future improvements may further incorporate these techniques.

## V. CONCLUSION

In this paper, an integrated vision-based home security system that assimilates both the face recognition and

motion detection subsystems into a self-functional automated system has been proposed. Novel techniques have also been proposed for both subsystems, and experimental tests have shown promising results in both accuracy rates and computation time. The integration of both subsystems is built upon a distributed processing architecture in which, the main operations of both subsystems are controlled by the main server while all the number crunching is performed by a dedicated powerful secondary workstation. While many issues and system considerations have been discussed here, the proposed system has shown to be feasible and practical for implementation in consumer environments.

In future work, various aspects of system design, such as hardware setup, costs, system requirements and ergonomical issues can be further studied to improve on the present integration proposal. As for the subsystem algorithms, future efforts can concentrate on improving reliability and robustness of both recognition and detection tasks to achieve better performance. There are also plans to implement a large-scale prototype in real-life environments.

## ACKNOWLEDGMENT

The authors thank Prof. Mark Nixon for providing the Southampton Human ID Gait Database [29].

## REFERENCES

- [1] Y.-K. Choi, K.-M. Kim, J.-W. Jung, S.-Y. Chun, and K.-S. Park, "Acoustic intruder detection system for home security," *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 130-138, Feb. 2005.
- [2] R. C. Luo, T. Y. Hsu, T. Y. Lin, and K. L. Su, "The development of intelligent home security robot," *Proc. of 2005 IEEE Int. Conf. on Mechatronics*, pp. 422-427, Taipei, Taiwan, July 10-12, 2005.
- [3] K. Hagiwara, Y. Chigira, N. Yoshiura, and Y. Fujii, "Proposal for a world wide home security system using PC-cameras: The e-Vigilante Network Project," *SICE 2004 Annual Conference*, pp. 1514-1517, vol. 2, Sapporo, Japan, Aug. 4-6, 2004.
- [4] F. Zuo, and P. H. N. de With, "Real-time embedded face recognition for smart home", *IEEE Trans. Consumer Electron.*, vol. 51, no. 1, pp. 183-190, Feb. 2005.
- [5] Identix's FaceIt website, <http://www.identix.com/trends/face.html>
- [6] VisionSphere Technologies' Unmask Plus website, [http://www.visionspheretech.com/unmask\\_plus.htm](http://www.visionspheretech.com/unmask_plus.htm)
- [7] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003.
- [8] R. Brunelli and T. Poggio, "Face recognition: Features vs. templates," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. 15, no. 10, pp.1042-1052, 1993.
- [9] J. See, M. Hanmandlu, and S. Vasikarla, "Fuzzy-based parameterized Gaussian edge detector using global and local properties," *Proc. Int. Conf. on Info. Tech.: Coding and Computing 2005*, pp. 101-106, Las Vegas, USA, Apr. 5-7, 2005.
- [10] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cog. Neuroscience.*, vol. 3, pp.72-86, 1991.
- [11] B. Zhou, Y. Gu, B. Li, G. Zhang, and T. Tian, "A practical algorithm for exception event detection for the home video security surveillance," *Proc. of 2001 Int. Conf. on Info-tech and Info-net*, vol 3, pp. 202-208, Beijing, China, Oct. 29-Nov. 1, 2001.
- [12] J. See, S. W. Lee, and M. Hanmandlu, "Human motion detection using fuzzy rule-base classification of moving blob regions," *Proc. Int. Conf. on Robotics, Vision, Information and Signal Processing 2005*, pp. 398-402, Penang, Malaysia, Jul. 20-22, 2005.

<sup>2</sup> fps denotes frames-per-second, the standard measure of frame rate

- [13] A. Shio, and J. Sklansky, "Segmentation of people in motion," *Proc. IEEE Workshop on Visual Motion*, pp. 325-332, 1991.
- [14] AT&T Laboratories Cambridge: The Database of Faces website, <http://www.uk.research.att.com/facedatabase.html>
- [15] F. S. Samaria, "Face recognition using hidden Markov models," Ph.D Thesis, Engineering Department, Cambridge University, UK, 1994.
- [16] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," *Proc. 2nd IEEE Workshop on Applications of Comp. Vis.*, pp. 138-142, 1994.
- [17] S. -H. Lin, S. -Y. Kung, and L. -J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. on Neural Network*, vol. 8, no. 1, pp. 114-132, 1997.
- [18] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural network approach," *IEEE Trans. on Neural Network, Special Issue on Neural Network and Pattern Recognition*, vol. 8, no. 1, pp.93-113, 1997.
- [19] J. Zhang, Y. Yan, and M. Lades, "Face recognition: eigenface, elastic matching and neural nets," *Proc. of IEEE*, vol. 85. no. 9, pp. 1423 -1435, 1997.
- [20] T. Satoonaka, T. Baba, T. Otsuki, T. Chikamura, and T. H. Meng, "Object recognition with luminance, rotation and location invariance," *Int. Conf. on Image Processing*, vol. 3, pp. 336-339, 1997.
- [21] A. V. Nefian, and M. H. III Hayes, "Hidden Markov models for face recognition," *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, pp. 2721-2724, 1998.
- [22] V. V. Kohir, and U. B. Desai, "Face recognition using a DCT-HMM approach," *4th IEEE Workshop on Applications of Comp. Vis.*, pp. 226-231, 1998.
- [23] S. M. Lucas, "Continuous n-tuple classifier and its application to real-time face recognition," *IEE Proc. Vis. Image, Sig. Proc.*, vol. 145, no. 5, pp. 343-348, 1998.
- [24] K. -M. Lam, and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 20, no. 7, pp. 673-686, 1998.
- [25] T. Tan, and H. Yan, "Face recognition by fractal transformations," *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, vol. 6, pp. 3537-3540, 1999.
- [26] M. S. Hoque, and M. C. Fairhurst, "Face recognition using the moving window classifier," *Proc. of British Machine Vision Conf.*, 2000.
- [27] P. C. Yuen, and J. H. Lai, "Face representation using independent component analysis," *Pattern Recog.*, vol. 35, no. 6, pp. 1247-1257, 2002.
- [28] O. Ayinde, and Y. Yang, "Face recognition approach based on rank correlation of Gabor-filtered images," *Pattern Recog.*, vol. 35, no. 6, pp. 1275-1289, 2002.
- [29] J. D. Shutler, M. G. Grant, M. S. Nixon, and J. N. Carter, "On a large sequence-based human gait database," *Proc. of 4th Int. Conf. on Recent Advances in Soft Comp.*, pp. 66-71, 2002.
- [30] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," *Proc. of IEEE Workshop on App. Of Comp. Vis.*, pp. 8-14, 1998.
- [31] L. Li, and M. K. H. Leung, "Fusion of two different motion cues for intelligent video surveillance," *Proc. of IEEE Int. Conf. on Electrical and Electronic Tech. TENCON*, vol. 1, pp. 341-344, 2001.
- [32] VideoIQ website, <http://www.geindustrial.com/cwc/products/ge-interlogix?id=VideoIQ>



security monitoring. He is a student member of IEEE.



**John See** (StM'06) received both his B.Eng. (Hons) in Electronics and MEng.Sc. degrees from Multimedia University, Malaysia in 2002 and 2005 respectively. He joined the Faculty of Information Technology, Multimedia University in 2004, where he is currently a lecturer and Ph.D student in facial biometrics. His research interests includes computer vision, pattern recognition, face and gait recognition, human motion analysis, biometrics and security monitoring. He is a student member of IEEE.

**Sze-Wei Lee** (M'99) was born in Malaysia in 1970. He obtained BEng (Hons) in Electronics and Optoelectronics, MPhil., and PhD from University of Manchester Institute of Science and Technology, UK in 1995, 1996, and 1998 respectively. He joined the Faculty of Engineering, Multimedia University, Malaysia in 1999, where he is now an associate professor. His research interests include digital signal processing and digital communications.