IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 27, 2020, accepted March 16, 2020, date of publication March 23, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982438

Design of Wireless Vision Sensor Network for Smart Home

TSUNG-HAN TSAI, (Member, IEEE), CHIH-CHI HUANG, CHIH-HAO CHANG, AND MUHAMMAD AWAIS HUSSAIN

Department of Electrical Engineering, National Central University, Taoyuan 32001, Taiwan

Corresponding author: Tsung-Han Tsai (han@ee.ncu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-008-078-MY3.

ABSTRACT In recent years, wireless vision sensor network (WVSN) is being used to retrieve video content from different image sensors which are connected to different devices wirelessly. This information is used to do video analysis which can help to automate different tasks such as video surveillance. For such systems, power consumption during processing and communicating information has been a challenge because of limited energy sources at the node. To deal with the energy consumption problem, in this paper WVSN is proposed with its algorithm and hardware implementation for a smart home application. The computation tasks have been divided between the sensor node and the central server. While taking care of privacy issues during the transmission of data, a low complexity system has been developed for sensor nodes. For video analysis, foreground segmentation, object labeling, and object tracking have been performed. An efficient binary data compression technique has been proposed to compress the information during the labeling process. The proposed system has a high recognition rate for gesture recognition and human tracking. The system can achieve eight frames per second during processing information on Raspberry Pi board. This system can be extended further to include other smart home applications.

INDEX TERMS Background segmentation, intelligent surveillance system, video surveillance system, wireless sensor network, wireless vision sensor network.

I. INTRODUCTION

Smart home is become a mainstream trend in products during recent years due to a lot of advancements to facilitate human life. With the control of different appliances, such as TV, lights, air conditioner, etc., smart homes are getting more and more common in people's life. Although many functions have been added to smart home systems, still other features can be included to make human life easier. The concept of a smart home with several different applications is shown in Fig. 1.

As of today, most of the deployed applications for a smart home are constructed for the analysis of physical parameters, such as temperature, pressure, humidity, location of objects, etc. Video-based sensing was rarely included in a smart home until recently. Video-based information processing can be utilized in many smart home applications, among which one includes a monitoring system. Instead of a person looking at different events on the screen, information can be analyzed automatically through different image processing techniques.

For a smart home, data is transmitted over the wireless network from different sensors to make the devices user-friendly as devices won't need many connection wires. So, in the case of video data, the captured data is transmitted wirelessly to a central processing node for processing and perform operations accordingly. One such system has been accomplished as Wireless Vision Sensor Network (WVSN) [1], [2]. The recent developments in CMOS vision sensors have made vision sensors able to acquire rich vision content from the environment and have fostered the development of WVSN. Figure 2 shows the topology of WVSN, which includes sensor nodes (SN), application nodes (AN), and a base station (BS). At sensor nodes, vision sensors can be deployed to perform the video analysis. To realize WVSN, it exposes to several different problems.

Generally, three different types of strategies are adopted for the implementation of WVSN [3]–[7], which has been described in Fig. 3. In the first type, no information is

The associate editor coordinating the review of this manuscript and approving it for publication was Nan Zhao.

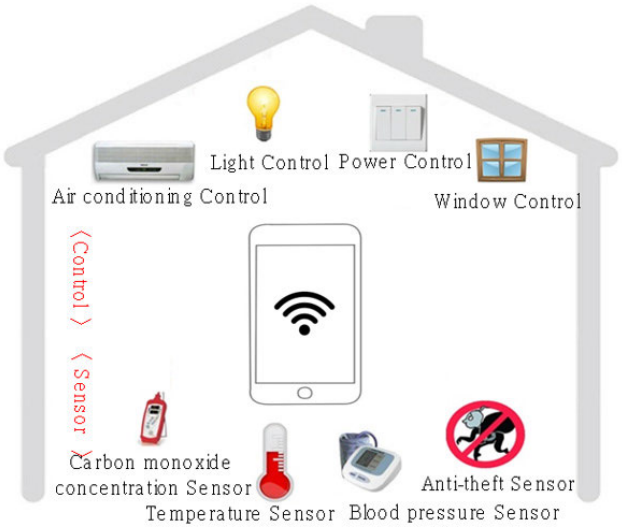

FIGURE 1. Schemes of smart home.

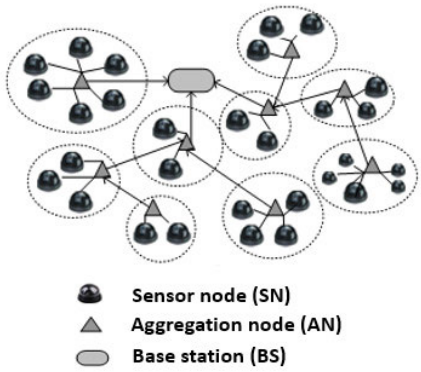

FIGURE 2. The topology of WVSN.

processed at the vision sensor network (VSN) and the raw information is sent to the server for all the processing. Although this type of strategy is simple, where only raw data is captured at VSN and sent to the server, it requires high bandwidth and high data transmission time [4]. Moreover, this strategy lacks the feature of data privacy as raw data is directly transferred to the server, which may lead to unsecure access to transmitted data.

In the second type of strategy, image processing is performed at VSN and feature results, which are acquired after image processing algorithms, are transferred to the server for further analysis. Examples of such systems include DSPcam and CMUcam3 [5], [7]. As compared to the first type, this method results in preserving the privacy of the data because not all of the data is sent to the server. This method has the bottleneck of the low battery as the hardware platform is required to be connected at VSN for processing information. Moreover, hardware designs are complex for this type as processing information is mostly dependent on software platforms that require high resource usage to execute different tasks. Referring to the third strategy, it is interpreted as the partial vision-based technique as complex tasks, such as labeling, feature extraction, etc., are performed at the server. This strategy results in the reduction of the processing information and the design complexity as the computation-intensive tasks are migrated to the server.

In this work, the third strategy has been utilized for the WVSN on a smart home. A high-speed algorithm has been proposed to perform image processing on captured data. The proposed system can be divided into image processing tasks, such as foreground segmentation, object labeling, and object tracking. Binary data compression and wireless transmission are also the main part of the proposed system, which are performed after the image processing. Foreground segmentation is one of the main operations to perform intelligent vision processing. It splits the binary foreground mask and then the object labeling with its mask information is performed which can be used in various applications. The earlier methods of labeling have used 4-neighbor or 8-neighbor labeling methods. Although these methods are simple, the processing speed is very slow, which does not help in the implementation of an intelligent processing system. In this study, the labeling method from [8] has been modified. The modified version has less complex architecture and consequently, it is suitable

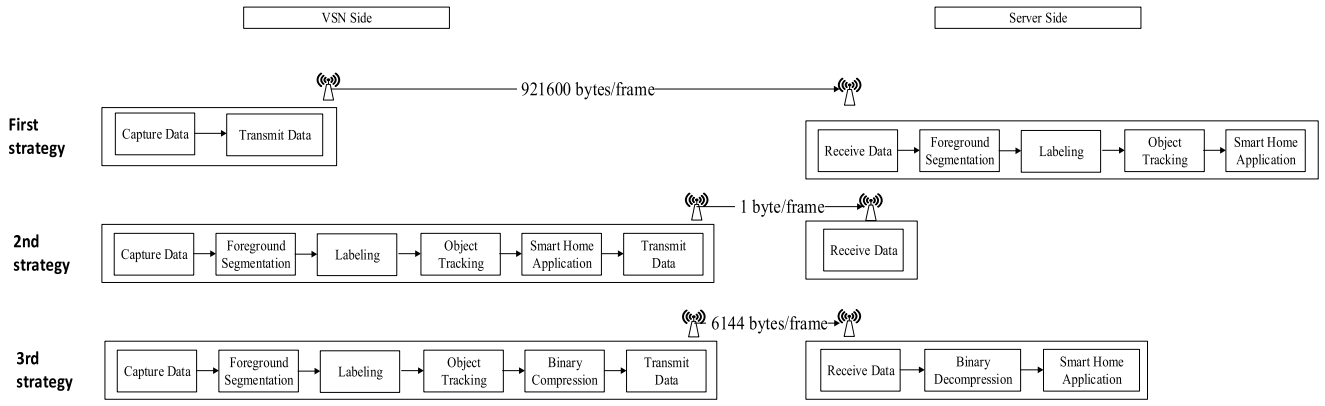

FIGURE 3. Three strategies on WVSN architecture.

for hardware platforms due to the low usage of resources on VSN. Referring to the tracking part, a kernel-based algorithm is used for the identification of the object movement. The object labeling information is used directly to find the object kernel when an object is moving. If the object is stationary, the target can be found using the mean-shift algorithm.

Furthermore, to deal with the limited bandwidth issue during data transmission, compression is performed on the foreground data. The foreground is a binary image and image compression can be performed on such images [2], [9]. A lossless image compression algorithm has been developed to maintain image quality. The duplicated row data is ignored to achieve a high compression ratio. In the final step, the whole system has been implemented on a well-known WSN platform, Raspberry Pi.

The main contributions of the paper include the following:

1) The combination of computer vision processes and image compression: the computer vision part contains the operations of segmentation, labeling, and tracking. The proposed algorithm can share information between the vision part and the compression part, which helps in higher compression of data.
2) Data privacy is maintained: by only sharing the extracted features from video to server node, data privacy is ensured which helps to protect sensitive data from intruders.
3) High precision recognition results: via the proposed recognition algorithm on the central server, the hand gesture recognition is performed with a high recognition rate.
4) Whole WVSN system is implemented: the whole system based on Zigbee transmission has been implemented as a demo system.

The remainder of this paper is organized as follows. Section II includes a brief review of the relevant literature which covers the different aspects of smart home and the related design challenges. Section III presents the proposed algorithm in detail. The experimental results are analyzed in section IV. Conclusions are presented in Section V.

II. RELATED WORK

This paper targets the processing of information captured through wireless video sensor nodes. The proposed system is constructed on the concept of WVSN. The details about WVSN are discussed first then vision processing techniques along with the challenges are discussed.

A. WVSN

Figure 2 shows the construction of WVSN which is based on the sensor network. These sensors are mostly resource-constrained in terms of battery life, computation power, etc. [10]. In WVSN, each sensor node (SN) captures video data and can perform different video analysis tasks to extract useful information.

The information extracted from the captured video is sent to an aggregation node (AN). The processing of the collected data and delivering important information to the base station (BS) are responsibilities of AN [2], [11]. A wide range of important applications has been proposed based on the WVSN including security monitoring, environment tracking, and the assisted living [12], [13]. Design for indoor robot localization using a wireless network system has been implemented in [14]. Quaritsch *et al.* have proposed a multi-camera tracking system using adjacent cameras [15]. A multi-view fall detection system has been implemented using the LHMM-based approach in [16]. To perform video surveillance in traffic scenarios, Quast and Kaup [17] have proposed an automatic system based on Gaussian Mixture Model (GMM). An implementation of WVSN for Characterization of Particles in Fluids has been proposed in [18] and [19]. As compared to the proposed work, the work of [18] and [19] uses FPGA for data processing at the node. An FPGA board is expensive board as compared to the normal embedded board and it is also difficult to program due to Verilog coding as compared to C or C++ implementation of the algorithm.

Currently, a well-established WSN, Zigbee, is widely used in smart home applications. It is a short-range communication protocol based on IEEE's 802.15.4 personal area network standard. Indoor data transmission distance for Zigbee is about 10 to 100 meters in line-of-sight. Zigbee has a flexible network structure and it can support the topology of start shape, tree shape and mesh shape [20]. Concerning to the low-cost implementation issue on Zigbee and short-range transmission distance in a smart home, the star topology is used. Start topology is easy to maintain and it has better network performance in small families of devices due to low power consumption [21].

B. DESIGN CHALLENGES

WVSN is achieved by merging the communication and computation techniques through signal processing, control theory, and embedded computing. Multiple factors influence the design which are addressed as follows;

1) FOREGROUND DETECTION TECHNIQUE

Through video surveillance system on WVSN, video data can be analyzed to monitor the people, and track activities at a place to automatically interpret the events happening at the site. An intelligent visual surveillance system includes several signal processing tasks, such as foreground detection, object tracking, and activity analysis. Foreground detection is a very important task during content analysis in surveillance systems. The foreground detection is performed to differentiate between the objects of interest in videos, such as human and vehicle, from the background. The performance of an intelligent system is affected by the precision of the object detection as the content analysis tasks include analyses of the detected objects.

In the earliest foreground segmentation methods, the difference between the current and previous frame is detected. If the pixel intensity is more than the specific threshold, the pixel is classified as a foreground or

background [22]–[25]. Although these methods are simple but low accuracy is a barrier in improving the system performance. This lead to the development of a mixture of Gaussian method, which is used to train the mixture background model [26]–[31]. The Gaussian distribution is needed to be initialized; thus, the precision of foreground segmentation would be affected as results depend upon the initialization of Gaussian distribution. In [24], it updated the background in faster and more stable. Lin *et al.* [32] have developed a background model which uses classification to learn about background scene. The model of Lin *et al.* [32] is based on updating background after regular intervals, yet without updating background model, background subtraction can be performed, such as done by [33].

Tracking is a critical task after foreground detection as the wrong tacking may lead to unwanted results. Traditionally it has many practical applications in the military, surveillance systems, and precision targeting. The principal-axis based contour descriptor has been proposed in [34] to detect the identity of the player in the match videos using their jersey number. In [35], a mean-shift algorithm is proposed where it is a non-parametric statistical method to seek the closest sample distribution.

2) HIGH BANDWIDTH DEMAND

Video streams require high transmission bandwidth because of currently available sensors. As modern-day sensors can produce high-quality data resolution, so it leads to higher bandwidth while transferring the data. Although WVSN allows the sensor to extract the raw data from the environment to execute video processing algorithms in the network. Yet this requires new distributed architecture as the processing power is limited, so it leads to performing the filtering and extracting semantic information from the sensor network.

As shown in Fig. 3, the vision-related processing can be performed either on the sensor node or the complex server node or even partially on both sides. Traditional approach relies on simple camera at node only which does not have any ability to process visual information, seen as Strategy 1 in Table 1. Modern approach stresses the feasibility of vision-based sensor node, as Strategy 3 in Table 1. In third type of approach, the vision-related operations are partitioned between the WVSN and the server. The amount of redundant transmission is reduced by using a local hardware platform, which is used to perform the preprocessing, segmentation and light-weight bi-level video coding. It is evident from Fig. 3 that the third strategy maintains the balance between data processing at the sensor node and data transmission towards the server node. The first strategy needs to transmit the data of 921600 bytes for one frame as compared to the second strategy, where all data is processed at a sensor node, which transmits only 1 byte of data to the server node. The third strategy needs to transmit only 6144 bytes per frame, which helps in the reduction of data transmission between the server and sensor node.

TABLE 1. The comparison of different WVSN strategies.

	Strategy1	Strategy2	Strategy3
Complexity	Low	High	Medium
Power Consumption	Low	High	Medium
Transmitted Data	Very Large	Very Less	Medium
Privacy	Low	High	High
Recognition Rate	High	High	Low
Remark	Simple camera	Vision-based camera	Partial vision-based camera
Example Implementation	[3], [4]	[5], [7]	[6], Proposed

3) SOURCE CODING TECHNIQUE

Data compression can reduce the data by using a proper video coding technique. However, there is a tradeoff between visual quality and computation complexity while using video coding techniques. The currently available video coding techniques, such as MPEG-4 need complex computation at the encoder side and consequently consumes large energy on the node [36]–[38]. Nonetheless, the designing of the decoders was intended to be simple because video codecs were designed for scenarios in which video is encoded once and decoded many times. WVSN [38] has constrained resources, so traditional codecs are not suitable to be used in WVSN. Region of interest (ROI) coding, change coding, a bi-level compression scheme, and a Huffman-run length code mixer are the basic components of a video encoding. The codec can select a small bitstream autonomously from four coding types i.e., image coding, change coding, ROI coding and change-ROI coding.

III. PROPOSED SYSTEM

In this section, the details of the proposed system for the smart home has been provided. The system can be divided into two main categories based on the processing i.e., sensor node processing part and server processing part as shown in Fig. 4. The sensor node processing part can be divided into sub-modules, such as foreground segmentation, object labeling, object tracking, and binary data compression. In server processing, data is decompressed and then extracted information is used for the smart home application.

A. FOREGROUND SEGMENTATION

Figure 5 illustrates the proposed foreground segmentation algorithm. The first step being performed is the background modeling. The background subtraction is the second step and then the area filter is applied. For moving-object detection, and background modeling, the background subtraction method has been successfully utilized. In this work, the algorithm of [39] has been modified to perform background modeling with a higher convergence rate. Gaussian mixture models (GMM) and adaptive mixture learning have been

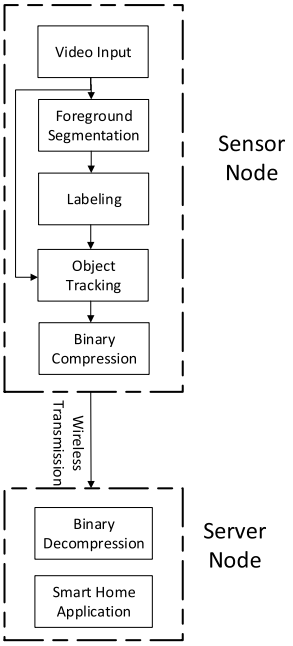

FIGURE 4. Flowchart of the proposed processing system for smart home.

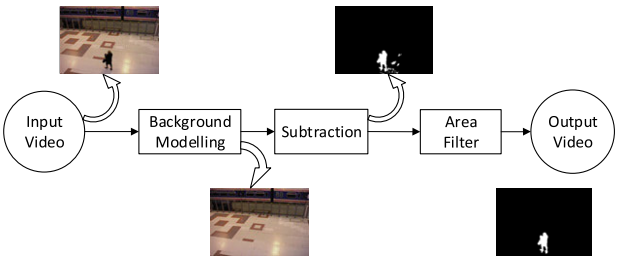

FIGURE 5. Illustration of foreground segmentation.

utilized for the proposed algorithm. In this method, each pixel is framed as a mixture of Gaussians. At any time t, the composition value of any pixel in the image can be described as $X = \{X_1, \ldots, X_t\}$, where the pixel is framed as a mixture of K Gaussian distributions. The probability of observing the current pixel value $P(X_t)$ is as follows:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * p\left(X_t, \mu_{i,t}, \alpha_{i,t}\right) \tag{1}$$

where K is the number of distributions, $\omega_{i,t}$ is the weight of the i-th Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i-th Gaussian in the mixture at time t, and $\sum_{i,t}$ is the covariance matrix of the *i-th* Gaussian in the mixture at time t.

For the Gaussian probability density function p can be defined as

$$p\left(X_t, \mu, \sum\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\sum\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \sum^{-1}(X_t-\mu_t)} \tag{2}$$

According to the parameters of each Gaussians, each pixel value can be evaluated whether it complies with background illumination or not. Each new pixel's value is compared with the existing K Gaussians until a matched Gaussian is found. This matching process is based on an online learning approximation. If the difference between pixel illumination and mean is lower than 2.5 covariance then it is a match condition for Gaussians else it is not matched. In case of a matched Gaussian, the different parameters of Gaussian are updated as:

$$\omega_{k,t} = (1-\alpha)\,\omega_{k,t-1} + \alpha\left(M_{k,t}\right) \tag{3}$$

$$\mu_{k,t} = (1-\rho)\mu_{k,t-1} + \rho X_t \tag{4}$$

$$\sigma^2_{k,t} = (1-\rho)\sigma^2_{k,t-1} + \rho(X_t - \mu_{k,t})^T(X_t - \mu_{k,t}) \tag{5}$$

and

$$\rho = \alpha\eta(X_t|\mu_{k,t}, \sigma_{k,t}) \tag{6}$$

At time t, $\omega_{k,t}$ represents the weight of the *k-th* distribution and α represents the learning rate. The value of $M_{k,t}$ is set to 1 for the matched condition model, and 0 for the other models. $\mu_{k,t}$ shows the mean value of the distribution and $\sigma_{k,t}$ represents the covariance of the K distribution. ρ is the learning factor for adapting the current distribution.

The adaptive mixture learning can be described as:

$$\theta_t = (1-\eta_t)\cdot\theta_{t-1} + \eta_t \cdot x_{t-1} \tag{7}$$

where t stands for the number of frames that passed during the time period from the start of the program until now. $\theta(t)$ is the updated parameter. $x(t-1)$ is the local estimate and $\eta(t)$ is rate control. If $\eta(t)$ equals to $1/t$, $\eta(t)$ will decrease as time increases to get a fast convergence rate. If $\eta(t)$ equals to α, the convergence rate will slow down. To achieve a faster convergence rate and higher precision segmentation, the proposed algorithm applies the updated Gaussian model. The formula is listed as follows:

$$\omega_{k,t} = (1-\eta_{k,t})\cdot\omega_{k,t-1} + \eta_{k,t} \tag{8}$$

$$\mu_{k,t} = (1-\eta_{k,t})\cdot\mu_{k,t-1} + \eta_{k,t}\cdot x \tag{9}$$

$$\sigma^2_{k,t} = (1-\eta_k)\cdot\sigma^2_{k,t-1} + \eta_{k,t}\cdot(x-\mu_{k,t-1})^2 \tag{10}$$

where

$$\eta_{k,t} = \frac{\alpha}{counter_{k,t}+1} + \alpha \tag{11}$$

Here α is a constant and at time t, $\omega_{k,t}$ represents the weight of the *k-th* distribution. The rate control is represented as $\eta_{k,t}$, and counter represents the number of matched Gaussians. The background modeling has multiple steps which are as follows. In the first step, the comparison of all Gaussians is performed to find out the matched Gaussian and the matched Gaussian is updated accordingly. In the case of a matched Gaussian, the counter is increased otherwise decreased in case of a non-matched Gaussian, the counter is decreased. All parameters of Gaussian do not change except the counter. The Gaussian with the largest value of ω/σ is found after the parameters of Gaussian have been updated. The pixel illumination of background is represented as μ_B, which is the mean of the Gaussian.

B. OBJECT LABELING

As the binary foreground mask is acquired, the connected pixels are labeled. As object labeling takes a large amount of time to process data, high speed and low complexity are the basis of designing the object labeling algorithm. Curie [8] have used the combination of run-length encoding (RLC) and object labeling algorithms to perform object labeling. This work modifies the work of Curie [8]. Figure 6 shows the flowchart of the proposed object labeling algorithm. Figure 7 shows the neighbors' labeling and calculation of RLC. The labeling is performed line by line in the proposed approach. During the labeling process, only foreground data is required, which has been represented as a white color in Fig. 7. As shown in Fig. 7 (a), the data at position one to five are connected together in the first line N1. These are labeled the same number, where 1 (marked as red) is the run of position one to five, and 2 (marked as green) is run of position eight. The same operations are performed on the second line N2. Then a run-length representation (run1, length1) and (run2, length2) are derived for N1 and N2 respectively.

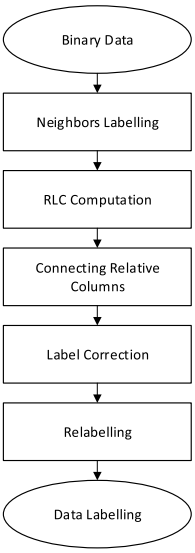

FIGURE 6. Flowchart of object labeling.

A connected RLC between two lines is shown in Fig. 8. The larger label value will be substituted by the smaller label value if the data is connected. Since the position 1 in N1 is connected to N2, the value 3 in N2 will be substituted by the value 1.

C. OBJECT TRACKING

When the labeling process is finished, object tracking is performed according to object labeling data. A color distribution model is formed based on the color feature. Then a kernel-based algorithm is used. If the model exists already, then only tracking is performed otherwise a new model is created. In the case of the stationary object, the candidate model is found by using the mean-shift algorithm otherwise the object labeling information is used directly to find the moving object.

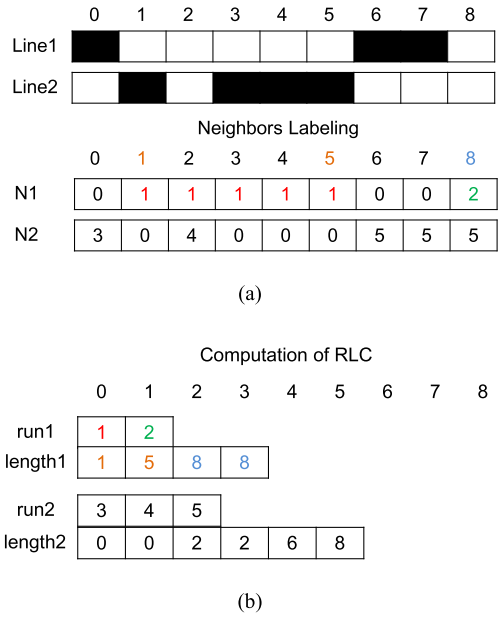

FIGURE 7. (a) Neighbors labeling and (b) RLC.

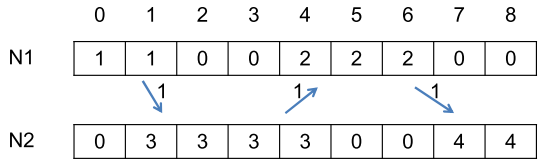

FIGURE 8. The connected RLC data between two lines.

There are a number of constraints when kernel-based tracking is used to perform tracking operation. While using kernel-based tracking, the moving speed of the object cannot be faster than a specific speed. Moreover, too often change in the kernel of the object will affect the results. Figure 9 represents the higher-level flowchart to perform object tracking. In order to establish the color distribution model, each pixel has RGB values ranging from 0 to 255. N equal segments are used with

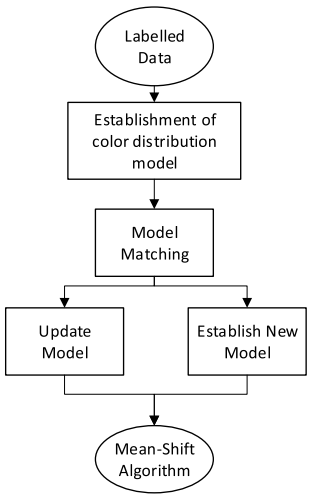

FIGURE 9. Flowchart of the object tracking.

a range of 0 to 255. As all pixel values can be used to classify different segments so u types of histogram have been used to classify the segments. The formula is as follows:

$$p_u = \frac{1}{C}\sum_{i=1}^{n}\left[\left|\frac{y - x_i^*}{h}\right|\right]\delta(b(x_i) - u) \quad (u = 1, 2, \ldots, N^3) \tag{12}$$

where y is the center of the kernel, x is a square near the center point of the scan, h is a constant, and C is a normalization factor. $b(x)$ is determined to find out where x belongs in histograms. Based on this kernel model, we can compare the kernels between the current frame and the previous frame by the following formula;

$$\rho = \sum_{u=1}^{N^3}\sqrt{p_u \cdot q_u} \tag{13}$$

where p is the kernel in the previous frame and q is the kernel in the current frame. The higher value of ρ means that two kernels are more similar. If the object is not matched, then a new object is created. If the object is stationary, the mean shift algorithm is used. It is an accelerated algorithm with an iterative method to quickly find the highest similarity to the kernel in the vicinity instead of performing a blind search on the wide range. The equation is as follows:

$$y_1 = \frac{\sum_{i=1}^{N^3} x_i \omega_i (y_0 - x_i)}{\sum_{i=1}^{N^3} \omega_i (y_0 - x_i)} \tag{14}$$

where

$$\omega_i = \sum_{u=1}^{N^3}\sqrt{\frac{q_u}{p_u}}\delta[b(x_i) - u] \tag{15}$$

y_0 are the iterative coordinates. When the iteration on the coordinates does not differ by more than one, this is named as convergence.

D. BINARY DATA COMPRESSION

Till this stage, object tracking, label assignment, and coordinates' recording have been completed. In binary data compression, all of this recorded information is used. The mask of the foreground object is sent to the server. It also includes the coordinate information of the mask along with the mask size. The information about the mask's coordinates and size is shown in Fig. 10. Binary data compression is used to effectively compress the transmitted data. Figure 11 shows the flowchart of the proposed binary data compression algorithm. Since each object has been tracked separately, this tracking information is used to calculate the size of each object.

Run-length coding has been used along with a skip-line method for data compression. For any foreground object, some rows usually have quite a high similarity, as shown in Fig. 12. When many lines of data are the same, this same part is utilized as the skip-line symbol. Figure 13 shows an example where the skip-line symbol is added. When the data between the first two lines are different, no skip-line symbol is added. While in the line two, three and fourth are the same, so a ski-line symbol is inserted. The value of two is assigned

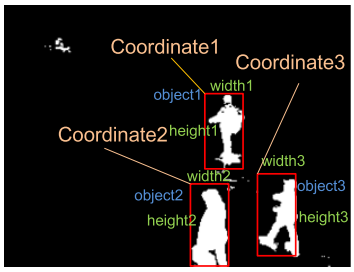

FIGURE 10. The object's mask with related information.

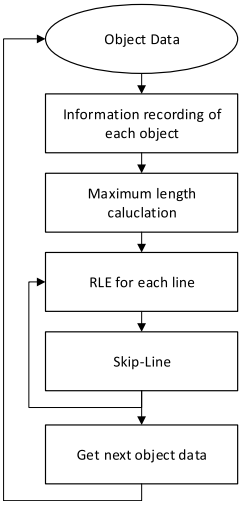

FIGURE 11. Flowchart of binary data compression.

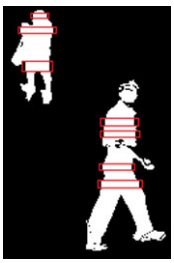

FIGURE 12. The foreground with a high degree of similarity.

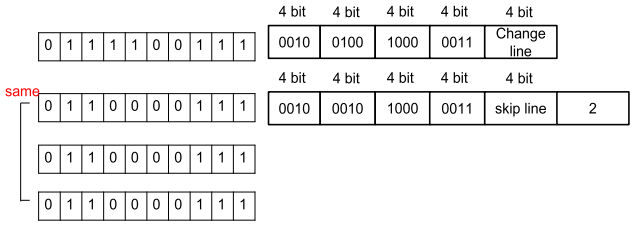

FIGURE 13. RLC with the skip-line algorithm.

to the symbol, which represents the presence of two same rows in data.

Figure 14 shows the format of the data packaging. The most important information to represent an object is recorded in the header. In the run-length coding, determination of the length is crucial for the compression ratio. If the length is too long, it will need a large data range to record it. Otherwise, if the length is too short, too many runs are generated accordingly. Here the width of each object is calculated to determine the length of run-length encoding. The calculation is performed by the following formula:

$$Length = \lfloor \log_2(width + 2) \rfloor + 1 \tag{16}$$

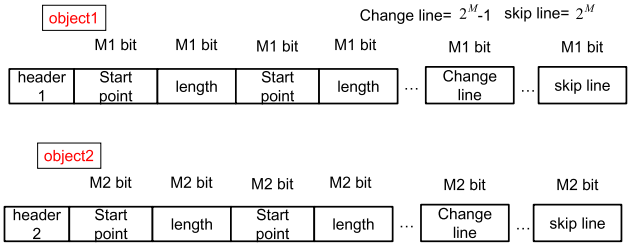

FIGURE 14. Data packaging format.

IV. EXPERIMENT RESULTS

The proposed algorithms have been verified to explore the performance and do performance comparisons in different scenarios. Furthermore, the whole system has been implemented on an embedded board as a demonstration system.

A. SYSTEM OVERVIEW

The star topology, as shown in Fig. 17, has been used to implement the WVSN. It has the advantages of simple design and low-cost implementation. As a star topology is constructed, all devices are connected to the coordinator. A central server collects the sent data for analysis and performs a variety of operations for smart home applications. The overall system of a smart home is illustrated in Fig. 18. The central server is acting as a coordinator, and all the end node devices are the sensors which can only send data to a central server. According to the transmission format of Zigbee, 100 Byte is grouped as a transmitted packet unit as in Fig. 19.

B. SMART HOME APPLICATIONS

By using wireless transmission, the mask data is sent to a central server to be used in some smart home applications. First, original data is retrieved from the compressed data, then the information about the number of objects with their coordinates, length and width information are recovered from compressed data. Based on this useful information, many back-end applications are invoked at the processing server. Firstly, humanoid tracking is performed. In the home environment, the people's location is critical information when some event is happening especially in the security situation, as illustrated in Fig. 15. Secondly, home appliances are controlled using hand gesture recognition. In this application, hand shape recognition is performed using skin

FIGURE 15. Human tracking.

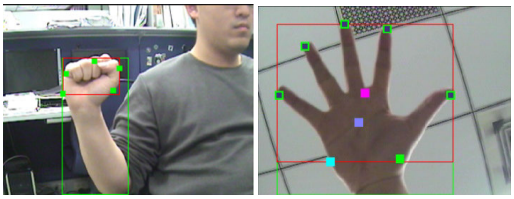

FIGURE 16. Results for hand gesture recognition with (a) zoom-out view, and (b) zoom-in view.

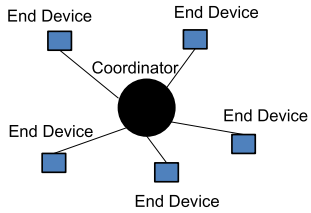

FIGURE 17. Star topology on WVSN.

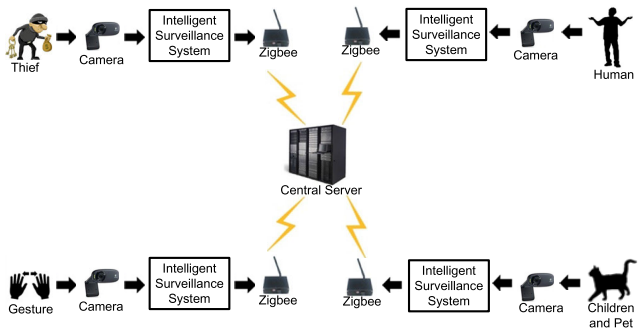

FIGURE 18. The overall WVSN system.

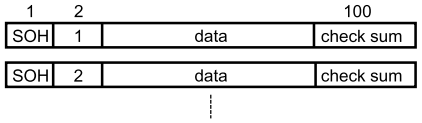

FIGURE 19. The format of the transmitted packet data on Zigbee.

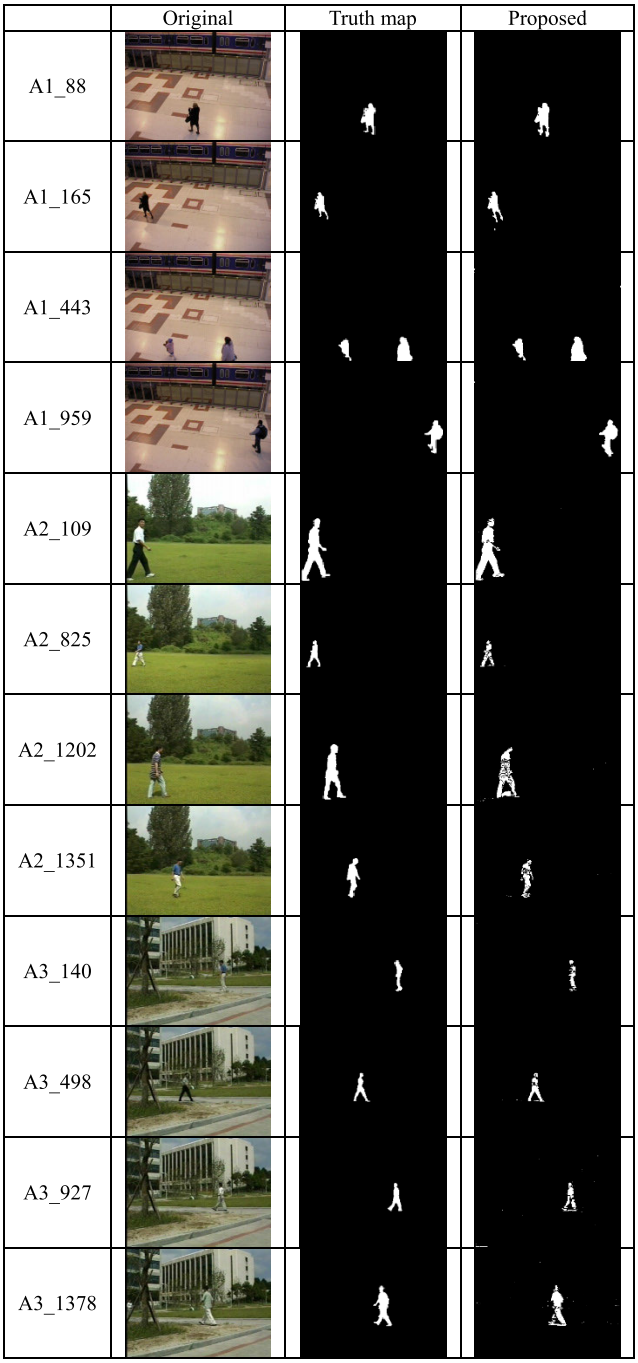

FIGURE 20. The subjective view of the compression result.

color detection. The range of skin color is defined as:

$$\begin{aligned} 0 < Y < 210 \\ 70 < U < 128 \\ 130 < V < 180 \end{aligned} \tag{17}$$

where YUV is a color space in terms of one luma (Y) and two chrominance (UV) components. As shown in a zoom-out view in Fig. 16 (a), the arm area could be long so the arm is removed and only the palm is left to increase the

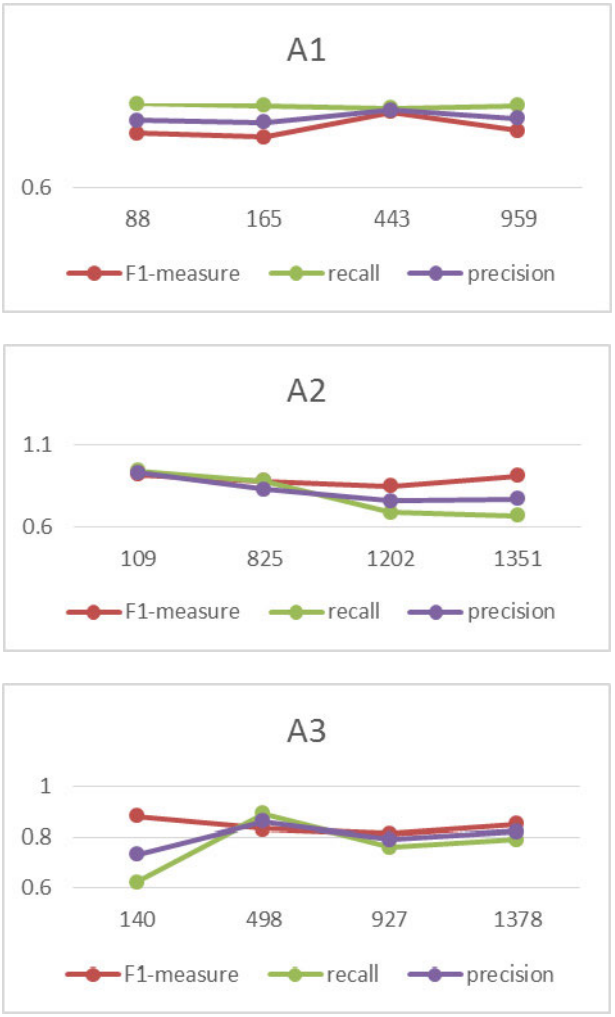

FIGURE 21. The evaluation of the foreground segmentation.

recognition rate. The area inside the green rectangle presents the full-arm while the area inside red rectangle presents and the palm part. The length determination method has been used to perform this operation. To find the convex hull points, Jarvis March algorithm has been utilized. As shown in a zoom-in view in Fig. 16 (b), the black dots with a green rectangle indicate the prominent convex hull points. Only the top convex hull points are taken into consideration for further processing. The positions of the convex hull points are used to find the center of gravity, which is shown in purple color. Using the different parameters of the palm such as length and width, different types of hand gestures are determined.

C. FOREGROUND SEGMENTATION ANALYSIS

First, the analysis of the performance of foreground segmentation has been performed. The accuracy performance of the system is directly affected by the foreground segmentation. If the redundant portions of the image are processed then it will result in spending extra time processing information. So, it will slow down the system. Moreover, if the cut out of the

interesting object is less than the truth map, it will result in either complete failure or some error while doing recognition and tracking. In Fig. 20, the three sequences A1∼A3 are shown with their truth maps and the prospective cut out by the developed algorithm. In these test sequences, F1-measure of above 0.8 has been achieved. As shown in Fig. 21 x-axis and y-axis represent the frame number of respective sequences and achieved values of F1-measure, recall, and precision, respectively.

D. COMPRESSION RATE ANALYSIS

In the compression rate analysis, the sequences are compressed and recalled, as in Fig. 22, for the subjective view evaluation. No data quality loss exists between the source and decoded result as the proposed binary data compression method is a lossless data compression technique. The existed defects are from the segmentation issue which has been discussed in the previous subsection. Figure 23 shows the comparison of the compression ratio where the horizontal axis

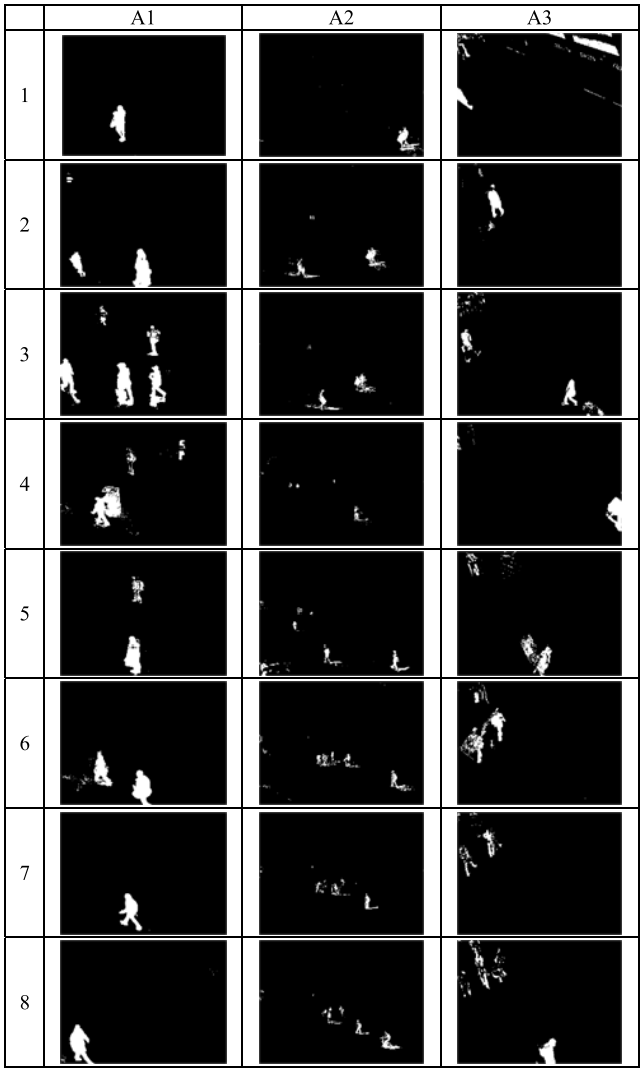

FIGURE 22. The subjective view of the compression result.

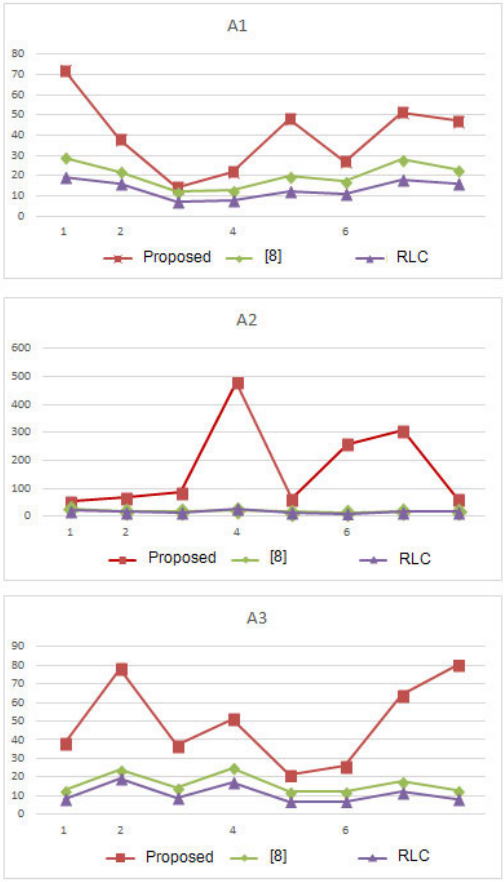

FIGURE 23. The comparison of compression ratio.

is the frame number and the vertical axis is the compression ratio. The compression ratio of the proposed method changes from twenty to five hundred in some frames. On average, the compression ratio of fifty times has been achieved. This design has been compared with two reference designs. One is the traditional RLC method and the other is the reference design of Sung and Kuo [9]. In [9], a skip mode concept has been introduced to reduce the data rate. Figure 23 shows that the proposed method performs better than the other methods of compression in [9].

E. SYSTEM INTEGRATION ON SMART HOME

Based on the proposed WVSN in Fig. 18, the physical implementation of the smart home system has been performed. The VSN has been envisioned by plugging in a camera in an embedded system board. The camera and Zigbee modules have been used with the Raspberry Pi board. The designed embedded system can be seen in Fig. 24. The processor used in Raspberry Pi board is a quad-core ARM Cortex-A53 processor. USB cameras and USB Zigbee modules have been utilized for physical implementation. To create the effect of multiple sensors, two cameras have been used. PC is serving as the central server, and it uses a Zigbee receiver module to receive data. The developed algorithm on PC has been

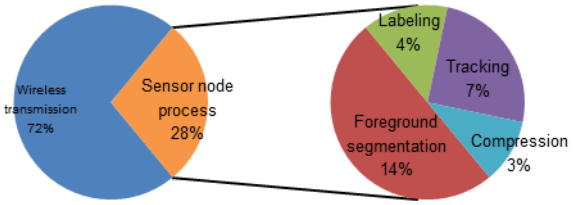

FIGURE 24. Time consumption for different perations in the embedded system.

developed on MFC (Microsoft Foundation Classes) build Windows UI.

In the sensor node, the first profiling of the execution time of the whole algorithm has been performed. All these experiments are made on the embedded system environment. Some optimizations are performed to increase the performance of the system. Figure 24 shows the average execution time of the whole system. It indicates that the data communication time takes most of the execution time. The main reason is that the applied Zigbee hardware module is a commercial module that could not be optimized by the general user. This limitation influences the proposed system performance. However, for a physical demonstration on smart home, still, eight frames per second can be achieved which can support a real-time application in a smart home scenario. By looking at the profiling results of the algorithm's computation part, the foreground segmentation takes the most of the time during computations. Without the overhead on Zigbee transmission work, 22∼24 frames per second can be achieved.

Several gestures are being recognized at the central server, which can be seen in Fig. 25. In each gesture, four different hands and different background cases are simulated. As for the objective evaluation, the corresponding recognition rate for different hand gestures is shown in Table 2. Table 3 shows the comparison of the proposed work with other state of the art works.

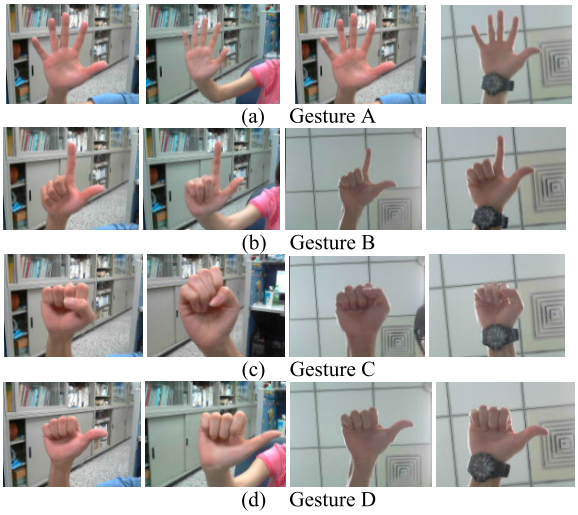

FIGURE 25. Four kinds of hand gesture.

TABLE 2. Accuracy of hand gesture recognition.

Gestures	Recognized as gesture A	Recognized as gesture B	Recognized as gesture C	Recognized as gesture D
Gesture A	445	21	0	55
Gesture B	3	374	6	1
Gesture C	0	0	443	4
Gesture D	11	0	0	433
Miss	31	105	51	7
Accuracy	89%	75%	89%	87%

TABLE 3. Comparison with related work.

	ViBe [40]	SSOBS [41]	SOBS [42]	MHTSS [43]	Proposed Method
Rec.	0.649	0.510	0.740	0.660	0.655
Pre.	0.811	0.893	0.710	0.770	0.772
F1-score	0.721	0.649	0.720	0.710	0.699

V. CONCLUSION

In this paper, a WVSN has been implemented for the smart home, which includes the features of the VSN and the server. A complete system-level algorithm has been developed to perform a vision-based analysis. The different number of test sequences have been tested and verified to evaluate the performance of the proposed system. Moreover, the hardware implementation of WVSN system is demonstrated.

REFERENCES

[1] M. Al Nuaimi, F. Sallabi, and K. Shuaib, "A survey of wireless multimedia sensor networks challenges and solutions," in *Proc. Int. Conf. Innov. Inf. Technol.*, Apr. 2011, pp. 191–196.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Comput. Netw.*, vol. 51, no. 4, pp. 921–960, Mar. 2007.

[3] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto, "Balancing computational and transmission power consumption in wireless image sensor networks," in *Proc. IEEE Symp. Virtual Environments, Hum.-Comput. Interface Meas. Syst.*, 2005, p. 6.

[4] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Adv. Multimedia*, vol. 2009, pp. 1–21, Jul. 2009.

[5] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh, "CMUcam3: An open programmable embedded vision sensor," in *Proc. Int. Conf. Intell. Robot. Syst.*, May 2007, pp. 1–17.

[6] M. Imran, N. Ahmad, K. Khursheed, M. A. Waheed, N. Lawal, and M. O'Nils, "Implementation of wireless vision sensor node with a lightweight bi-level video coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 3, no. 2, pp. 198–209, Jun. 2013.

[7] A. Kandhalu, A. Rowe, and R. Rajkumar, "DSPcam: A camera sensor system for surveillance networks," in *Proc. 3rd ACM/IEEE Int. Conf. Distrib. Smart Cameras (ICDSC)*, Aug. 2009, pp. 1–7.

[8] L. Lacassagne, M. Milgram, and P. Garda, "Motion detection, labeling, data association and tracking, in real-time on RISC computer," in *Proc. 10th Int. Conf. Image Anal. Process.*, Sep. 1999, pp. 520–525

[9] H. Sung and W.-Y. Kuo, "A skip-line with threshold algorithm for binary image compression," in *Proc. 3rd Int. Congr. Image Signal Process.*, Oct. 2010, pp. 515–523.

[10] V. Kaseva, T. D. Hämäläinen, and M. Hännikäinen, "A wireless sensor network for hospital security: From user requirements to pilot deployment," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, p. 15, Dec. 2011.

[11] B. S. Kim, H. S. Park, K. H. Kim, D. Godfrey, and K. Il Kim, "A survey on real-time communications in wireless sensor networks," *Wireless Commun. Mob. Comput.*, vol. 2017, Oct. 2017, Art. no. 1864847.

[12] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: A review," *IEE Proc. Vision, Image, Signal Process.*, vol. 152, no. 2, pp. 192–204, 2005.

[13] I. Akyildiz, T. Melodia, and K. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 32–39, Dec. 2007.
[14] X. Dong, B. Su, and R. Jiang, "Indoor robot localization combining feature clustering with wireless sensor network," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 175, Dec. 2018.
[15] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous multicamera tracking on embedded smart cameras," *EURASIP J. Embedded Syst.*, vol. 2007, no. 1, pp. 1–10, 2007.
[16] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1522–1532, Nov. 2008.
[17] K. Quast and A. Kaup, "AUTO GMM-SAMT: An automatic object tracking system for video surveillance in traffic scenarios," *EURASIP J. Image Video Process.*, vol. 2011, pp. 1–14, Dec. 2011.
[18] M. Imran, K. Khursheed, N. Lawal, M. O'Nils, and N. Ahmad, "Implementation of wireless vision sensor node for characterization of particles in fluids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 11, pp. 1634–1643, Nov. 2012.
[19] K. Khursheed, M. Imran, A. W. Malik, M. O'Nils, N. Lawal, and B. Thörnberg, "Exploration of tasks partitioning between hardware software and locality for a wireless camera based vision sensor node," in *Proc. 6th Int. Symp. Parallel Comput. Electr. Eng. (PARELEC)*, Apr. 2011, pp. 127–132.
[20] K. Vats, P. Jain, L. Jaiswal, and S. Singh, "ZigBee based WSN topology simulation investigation and performance analysis using OPNET," *Int. J. Adv. Sci. Res. Technol.*, vol. 3, no. 2, pp. 667–677, 2012.
[21] J.-M. Dricot, S. Van Roy, G. Ferrari, F. Horlin, and P. De Doncker, "Impact of the environment and the topology on the performance of hierarchical body area networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, p. 122, Dec. 2011.
[22] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.
[23] R. Cutler and L. Davis, "View-based detection and analysis of periodic motion," in *Proc. 14th Int. Conf. Pattern Recognit.*, vol. 1, 1998, pp. 495–500.
[24] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 105–112, Aug. 2002.
[25] S. Chen, J. Zhang, Y. Li, and J. Zhang, "A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 118–127, Feb. 2012.
[26] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.
[27] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M.-T. Sun, "Robust detection of abandoned and removed objects in complex surveillance videos," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 5, pp. 565–576, Sep. 2011.
[28] A. Shimada, D. Arita, and R.-I. Taniguchi, "Dynamic control of adaptive Mixture-of-Gaussians background model," in *Proc. IEEE Int. Conf. Video Signal Based Surveill.*, Nov. 2006, p. 5.
[29] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.
[30] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proc. Workshop Motion Video Comput.*, 2002, pp. 22–27.
[31] T.-H. Tsai, W.-T. Sheu, and C.-Y. Lin, "Foreground object detection based on multi-model background maintenance," in *Proc. 9th IEEE Int. Symp. Multimedia Workshops (ISMW)*, Dec. 2007, pp. 151–159.
[32] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "Learning a scene background model via classification," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1641–1654, May 2009.
[33] D.-M. Tsai and S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009.
[34] S.-W. Sun, W.-H. Cheng, Y.-L. Hung, I. Fan, C. Liu, J. Hung, C.-K. Lin, and H.-Y.-M. Liao, "Who's who in a sports video? An individual level sports video indexing system," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 937–942.
[35] C.-H. Chen, C.-Y. Lin, S.-Y. Li, and T.-H. Tsai, "High level feature: Head and body co-trakcing by Kalman filter," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 725–728.
[36] J. J. Ahmad, H. A. Khan, and S. A. Khayam, "Energy efficient video compression for wireless sensor networks," in *Proc. 43rd Annu. Conf. Inf. Sci. Syst.*, Mar. 2009, pp. 629–634.
[37] N. Imran, B.-C. Seet, and A. C. M. Fong, "A comparative analysis of video codecs for multihop wireless video sensor networks," *Multimedia Syst.*, vol. 18, no. 5, pp. 373–389, Oct. 2012.
[38] K. R. Vijayanagar and J. Kim, "Real-time low-bitrate multimedia communication for smart spaces and wireless sensor networks," *IET Commun.*, vol. 5, no. 17, pp. 2482–2490, Nov. 2011.
[39] T.-H. Tsai, W.-T. Sheu, and C. Y. Lin, "Foreground object detection based on multi-model background maintenance," in *Proc. IEEE Int. Symp. Multimedia Workshops*, Taipei, Taiwan, Dec. 2007, pp. 151–159.
[40] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[41] K. Sehairi, F. Chouireb, and J. Meunier, "Comparative study of motion detection methods for video surveillance systems," *J. Electron. Imag.*, vol. 26, no. 2, Apr. 2017, Art. no. 023025.
[42] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?" in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 21–26.
[43] M. Chandrajit, R. Girisha, and T. Vasudev, "Motion segmentation from surveillance video using modified Hotelling's T-Square statistics," *Int. J. Image, Graph. Signal Process.*, vol. 8, no. 7, pp. 41–48, Jul. 2016.

TSUNG-HAN TSAI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1990, 1994, and 1998, respectively. From 1999 to 2000, he was an Associate Professor of electronic engineering with Fu Jen Catholic University. He joined National Central University, in 2000, where he has been a Full Professor with the Department of Electrical Engineering, since 2008. He is currently the Director of the Intelligent Chip and System Center, National Central University, and the Principal Investigator of the National Program for Intelligent Electronics. He has been awarded more than 30 patents and 220 refereed articles published in international journals and conferences. His research interests include VLSI signal processing, video/audio coding algorithms, DSP architecture design, wireless communications, and system-on-chip design. He is a member of the Audio Engineering Society (AES) and the Institute of Electronics, Information and Communication Engineers (IEICE). He is also a member of the Technical Committee of IEEE Circuits and Systems Society and the Technical Program Committee Member or the Session Chair of several international conferences. He received the Industrial Cooperation Award from the Ministry of Education, Taiwan, in 2003.

CHIH-CHI HUANG received the M.S. degree in electrical engineering from National Central University, Taiwan, in 2013. His areas of interest are video signal processing and computer vision problems on surveillance systems.

CHIH-HAO CHANG received the M.S. degree from the Department of Electrical Engineering, National Central University, Taiwan. His areas of interest are video signal processing and computer vision problems on surveillance systems.

MUHAMMAD AWAIS HUSSAIN received the B.S. degree in electrical and electronics engineering from the Namal College, Pakistan, in 2015, and the M.S. degree from National Central University, Taiwan, in 2018, where he is currently pursuing the Ph.D. degree in electrical engineering. His researches focus on low-power VLSI design for compression algorithms, image processing, and compression algorithms.