

Research Article

An Indoor Video Surveillance System with Intelligent Fall Detection Capability

Ming-Chih Chen and Yang-Ming Liu

Department of Electronic Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung City 811, Taiwan

Correspondence should be addressed to Ming-Chih Chen; mjchen@nkfust.edu.tw

Received 12 September 2013; Accepted 8 October 2013

Academic Editor: Teen-Hang Meen

Copyright © 2013 M.-C. Chen and Y.-M. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a novel indoor video surveillance system, capable of detecting the falls of humans. The proposed system can detect and evaluate human posture as well. To evaluate human movements, the background model is developed using the codebook method, and the possible position of moving objects is extracted using the background and shadow eliminations method. Extracting a foreground image produces more noise and damage in this image. Additionally, the noise is eliminated using morphological and size filters and this damaged image is repaired. When the image object of a human is extracted, whether or not the posture has changed is evaluated using the aspect ratio and height of a human body. Meanwhile, the proposed system detects a change of the posture and extracts the histogram of the object projection to represent the appearance. The histogram becomes the input vector of K-Nearest Neighbor (K-NN) algorithm and is to evaluate the posture of the object. Capable of accurately detecting different postures of a human, the proposed system increases the fall detection accuracy. Importantly, the proposed method detects the posture using the frame ratio and the displacement of height in an image. Experimental results demonstrate that the proposed system can further improve the system performance and the fall down identification accuracy.

1. Introduction

In Taiwan, falls represent the second leading accidental cause of death among elderly people. The rate of falling down among the elderly ranges from roughly 15% to 40% annually, and the incidence of falling increases as they grow older. Most falls lead to hospitalization of the elderly, residing in nursing homes, and barriers to daily activities. Elderly people most commonly fall in the bathroom, toilet, living room, and bedroom. Therefore, the ability to detect the falling of elderly people quickly would decrease the rate of injuries and reduce medical treatment costs. The damage degree of falls among the elderly is often decided by the time of discovery, transport, and emergency medical services. Developing electronic technologies facilitates the integration of sensors, computer vision, and the increasingly popularity of the wireless network. Such integrated applications can help the elderly to avoid potentially dangerous situations. This automatic system also reduces neglect among individuals and achieves zero-distance medical treatment.

The rest of this paper is organized as follows. Section 2 surveys previous design methods for detecting human falls. Section 3 then introduces the proposed system design. Next, Section 4 summarizes the experimental results of the proposed system and compares them with those of other ones. Conclusions are finally drawn in Section 5, along with recommendations for future research.

2. Related Work

Recognizing human behaviors using computer vision techniques is actively researched in various fields. A simple method detects a fall by analyzing the aspect ratio of the bounding box of a moving object [1–5]. Another design [6] evaluates whether or not a fall occurs by using motion history image and the ratio and angle of ellipse. In posture recognition, a commonly used feature vector is the projection histogram, in which the pixel number of row- and column-wise foreground objects is calculated, followed by a comparison with stored posture templates to evaluate the human

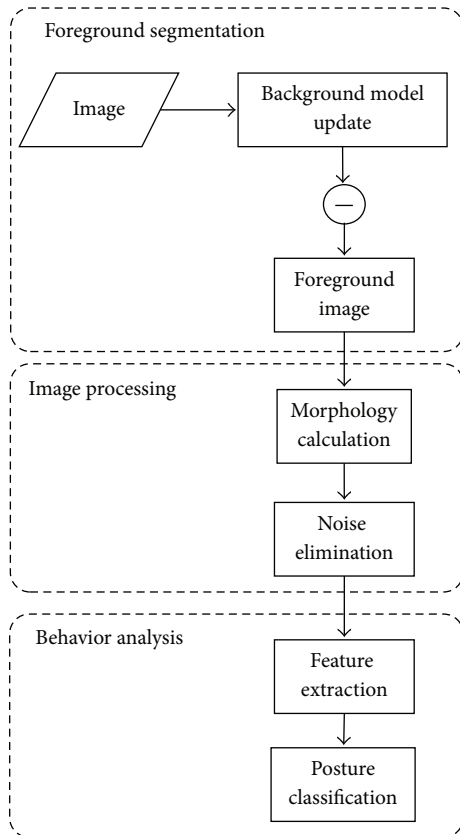


FIGURE 1: Proposed system architecture.

posture [2, 3, 7–12]. Another design [13] incorporates a fall detection method which uses combination of ellipse and skin color matches for head localization. The centroid of the feet area is identified by using the head location and medial axis. During the fall detection phase, a scene image is divided into equal-sized blocks. These blocks are then categorized into head, floor, and neutral ones. When the vertical displacement of a human head exceeds a threshold, whether the location of a human head lies in the floor block must be determined to evaluate whether or not a fall occurs.

Another design [14] identifies the centroid of a foreground object by using the 3D model of the human body and also identifies the floor plane by using the random sample consensus (RANSAC) plane detection algorithm. The system can detect whether the fall event occurs or not by calculating the distance between the centroid of human body and floor plane judge. Despite this, the above systems expend a significant amount of resources when calculating posture recognition. This work presents a novel indoor video surveillance system to recognize the falls by spending fewer resources and achieving a higher accuracy rate compared with previous ones.

3. Proposed System Architecture

Figure 1 shows the system framework, which consists of foreground segmentation, image processing, and behavior

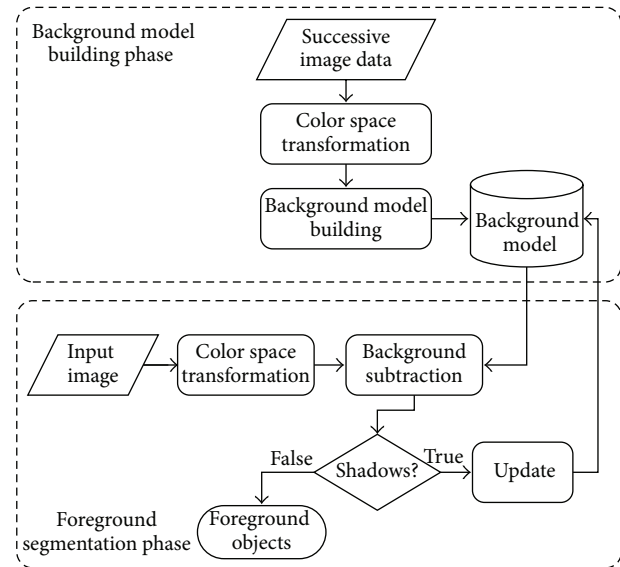


FIGURE 2: Foreground segmentation flow.

analysis. The proposed system first obtains the segmentation of moving objects by using the background subtraction method and the image processing technique to eliminate noise interference. When the moving objects are identified, the postures of these detected objects are recognized using behavior analysis.

3.1. Foreground Segmentation. While a vision surveillance system heavily emphasizes how to detect moving objects, the segmentation accuracy of moving objects can improve the performance of further analysis (e.g., object extraction or posture analysis). Figure 2 shows the proposed foreground segmentation flow. The flow first collects successive images and then transforms them into hue, saturation, and value (HSV) color space and builds the preliminary background model. When a new image is inputted, the background subtraction method described in [1] extracts the foreground image and removes shadows by using the shadow detection method. Finally, the background model is updated, owing to background subtraction.

3.2. Image Processing. When a foreground image is identified, some noises appear in the image or some holes appear in moving objects. Additionally, the morphology model is used to eliminate some small noise interference and fill holes of moving objects. Moreover, large noises are eliminated by using the information of each foreground object boundaries and foreground object area to distinguish between larger noises and regions of interest. Figure 3 shows the above procedure.

3.3. Behavior Analysis. The proposed system finally recognizes the posture of each foreground object. Features of each foreground object are first extracted for further analysis and

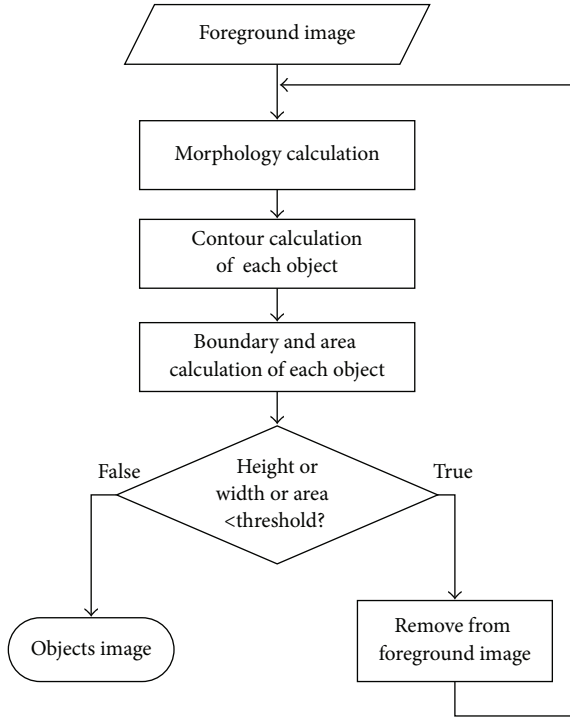


FIGURE 3: Image processing flow.

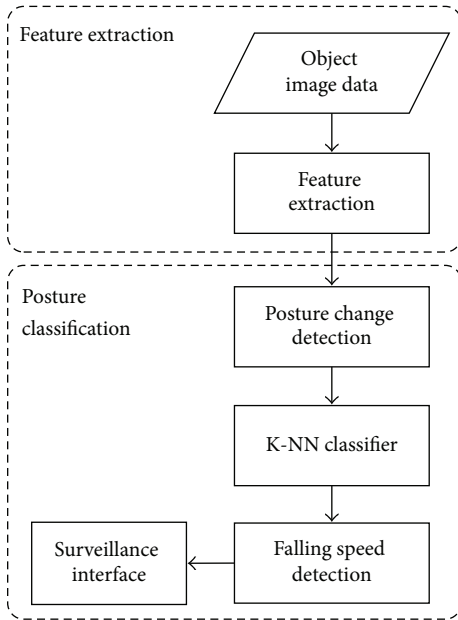


FIGURE 4: Behavior analysis flow.

classification. The classifier uses the K-Nearest Neighbor (K-NN) algorithm and combines it with posture change detection. When a change event of the posture is detected, the proposed system recognizes which posture is changed by using the K-NN based classifier. Finally, the foreground object between the real fall and only a supine position is discriminated by using the falling speed. Figure 4 shows the detailed processes of behavior analysis.

3.3.1. Feature Extraction. Extracting useful feature information in a foreground image is of priority concern in recognition processes. The proposed system extracts three main features, as described in the following.

(a) *Projection Histograms.* The horizontal and vertical projection histogram of foreground objects is derived by calculating the pixel number of row- and column-wise foreground objects. Since projection histograms vary according to location of foreground objects in the scene, the normalization step must be performed by using the discrete fourier transform (DFT) method described in [2]. By assuming that the image size is $N \times M$ pixels, the normalization step is as follows.

Step 1. Calculate the horizontal and vertical projection histogram of foreground objects:

$$\begin{aligned} \text{HZ}(y) &= \left| (x_p, y_p) \in F, y_p = y \right|, \\ \text{VT}(x) &= \left| (x_p, y_p) \in F, x_p = x \right|. \end{aligned} \quad (1)$$

Step 2. Apply DFT to $\text{HZ}(y)$ and $\text{VT}(x)$:

$$\begin{aligned} \text{DFT}_{\text{HZ}(h)} &= \frac{1}{N} \sum_{i=1}^N \text{HZ}(i) e^{-2j\pi(hi/N)}, \quad h = 0, 1, \dots, N-1, \\ \text{DFT}_{\text{VT}(v)} &= \frac{1}{M} \sum_{i=1}^M \text{VT}(i) e^{-2j\pi(vi/M)}, \quad v = 0, 1, \dots, M-1. \end{aligned} \quad (2)$$

Step 3. Normalize based on the following formula:

$$\begin{aligned} \text{NDFT}_{\text{HZ}(h)} &= \frac{|\text{DFT}_{\text{HZ}(h)}|}{|\text{DFT}_{\text{HZ}(1)}|}, \quad h = 2, \dots, 50, \\ \text{NDFT}_{\text{VT}(v)} &= \frac{|\text{DFT}_{\text{VT}(v)}|}{|\text{DFT}_{\text{VT}(1)}|}, \quad v = 2, \dots, 50. \end{aligned} \quad (3)$$

In (2), magnitudes of these DFT coefficients decay for large values of h and v . The first fifty significant DFT coefficients are selected and normalized by (3). The normalized magnitudes of fifty significant DFT coefficients of different postures are obtained by the above equations. After the above three steps are performed, the normalization of posture is completed.

(b) *Variance of Aspect Ratio in the Human Body.* The boundary of foreground objects is described using the bounding box. In most works [2, 5, 6, 9, 14, 15], the discrimination between normal state and fall state uses the ratio of human body. By using the variance in the ratio of the human body, the proposed system detects whether or not the posture has changed. Variance in the aspect ratio can be calculated as follows:

$$R(t) = \frac{w(t)}{h(t)}, \quad (4)$$

$$\mu_r(t) = (1 - \alpha) \mu_r(t-1) + \alpha R(t), \quad (5)$$

$$\sigma_r(t) = R(t) - \mu_r(t-1). \quad (6)$$

In (4), the parameters $w(t)$, $h(t)$ denote width and height of the bounding box at current time t , respectively, and $R(t)$ represents the ratio of width and height. In (5), the value $\mu_r(t)$ refers to the mean value of the aspect ratio at current time t and value $\mu_r(t-1)$ denotes the mean value of the aspect ratio at previous time $(t-1)$. Value α is the updated parameter. Value $\sigma_r(t)$ in (6) represents the variance value of aspect ratio at current time t . The variance is small if an individual does not change his or her posture. Otherwise, the variance is large if an individual changes his or her posture. When the variance exceeds a threshold, which posture has changed is determined using the K-NN based classifier.

(c) *Variance of Height in the Human Body.* Whether or not the posture has changed when using the variance in human body height is worthwhile to detect because that height varies according to different postures or image resolutions. Variance of height is calculated as follows:

$$\mu_h(t) = (1 - \alpha) \mu_h(t-1) + \alpha H(t), \quad (7)$$

$$\sigma_h(t) = H(t) - \mu_h(t-1). \quad (8)$$

In (7), values $H(t)$ and $\mu_h(t)$ denote the height and mean value of human height at current time t , respectively, while value $\mu_h(t-1)$ represents the mean value of the height at previous time $(t-1)$. Also, value α refers to the updated parameter. In (8), value $\sigma_h(t)$ denotes the mean value of height at current time t . Notably, the variance is small if an individual does not change his or her posture. Otherwise, the variance is large if an individual changes his or her posture. When the variance exceeds a threshold, which posture has changed is determined using the K-NN based classifier.

3.3.2. Posture Classification. The proposed posture classification system has two major components: K-NN classifier and falling speed detection. The system first detects whether or not the posture has changed by using the features of the aspect ratio and human body height. When a posture change is detected, exactly which posture has changed is recognized using the K-NN based classifier. Otherwise, the system selects the recent output in the classifier as the current output. Figure 5 shows the posture change detection. K-NN algorithm consists of two phases: offline training and classification. As described in [9], the algorithm trains the posture model based on five posture types, with each type having three templates. Figure 6 shows the five postures of standing, sitting, bending, lying, and lying toward. The five images are retrieved from the video surveillance and processed after eliminating the background interference.

During the classification phase, the distance between the current frame and the stored template i of posture j can be calculated by (9). The winning posture is taken by a majority vote. When the posture is lying towards the camera, the height test is conducted to distinguish between bending posture and lying towards posture. When the posture is lying or lying towards the camera, the falling speed is used to

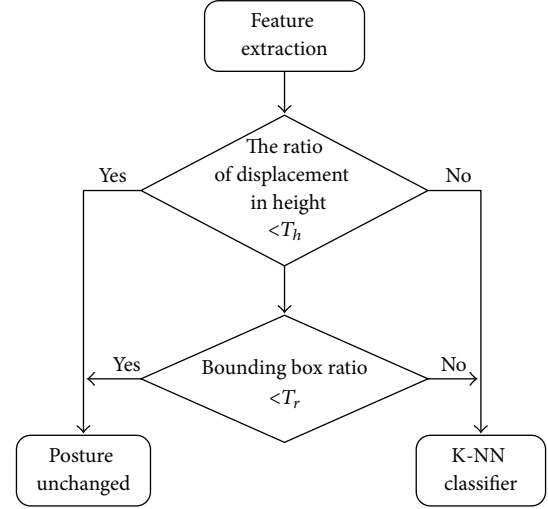


FIGURE 5: Posture change detection flow.

distinguish between a real fall and an individual in a supine position:

$$S_{i,j} = \sqrt{\sum_{y=0}^M (HZ^v(y) - HZ_{i,j}^v(y))^2 + \sum_{x=0}^N (VT^v(x) - VT_{i,j}^v(x))^2}, \quad (9)$$

where $HZ^v(y)$ denotes the horizontal projection of the current frame; $HZ_{i,j}^v(y)$ represents the horizontal projection of the stored template i of posture j ; $VT^v(x)$ refers to the vertical projection of the current frame; $VT_{i,j}^v(x)$ denotes the vertical projection of the stored template i of posture j ; and $S_{i,j}$ represents the distance between the current frame and the stored template i of posture j .

(1) *Test of Height.* The shape of lying towards posture occasionally resembles that of a bending posture, necessitating a method to distinguish between these two postures. A previous design [9] distinguishes between standing and lying towards postures by using the angle of bounding box. This work presents a novel method by using the mean ratio of the height in standing and lying towards postures to distinguish between them. The method is described in what follows:

$$\begin{aligned} &\text{lying_toward,} && \text{if } \frac{H(t)}{\mu_{\text{stand}}(t)} < T, \\ &\text{bending,} && \text{else,} \end{aligned} \quad (10)$$

where $H(t)$ denotes the height of a lying towards posture; $\mu_{\text{stand}}(t)$ represents the mean of the height in a standing posture, T is threshold, and its value ranges from around 0.4 to 0.5.

(2) *Test of Falling Speed.* A human occasionally lies without falling. To distinguish between these two postures, this work

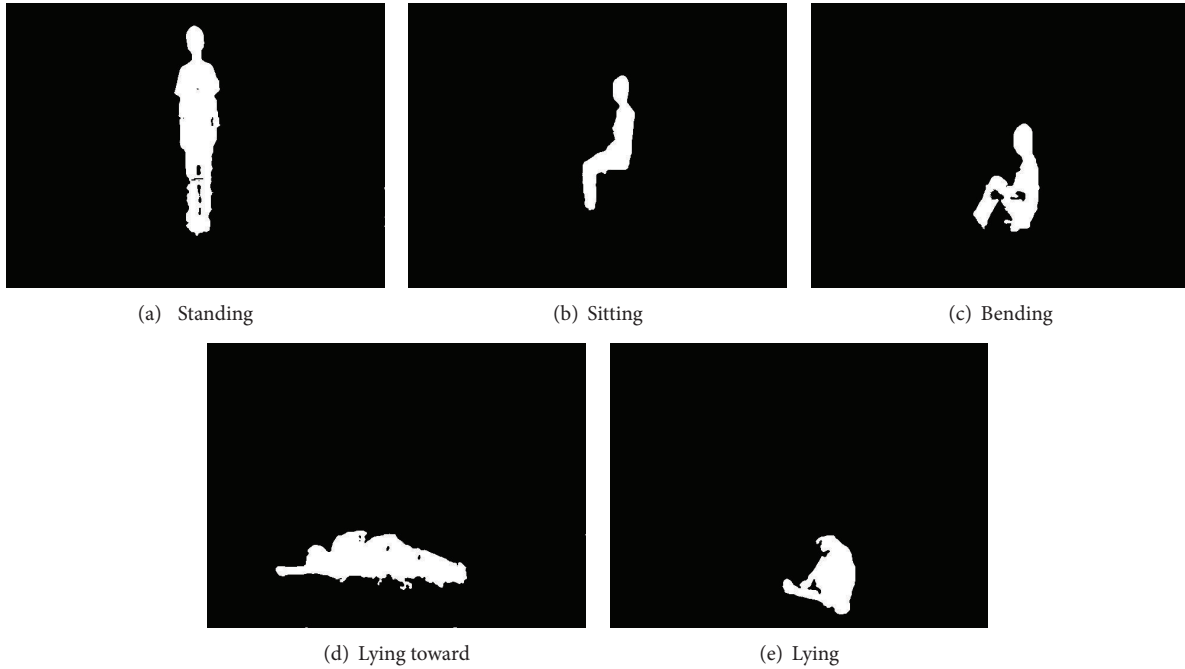


FIGURE 6: Five postures defined in our system.

adopts the method in [9], which uses temporal information including the last time t_{stand} of standing posture and the time t_{lying} of a lying posture. When the difference of time is less than threshold T , we can infer that a real fall occurs as follows:

$$\begin{aligned} & \text{fall,} && \text{if } t_{lying} - t_{stand} < T, \\ & \text{lying,} && \text{else.} \end{aligned} \tag{11}$$

4. Experimental Results

While implemented by OpenCV library and Visual C++ 2008, the proposed system runs on an Intel Core i7 3.4 GHz laptop PC with 8 GB memory. To evaluate the system performance and accuracy, the experimental environment is an indoor place with a single and fixed camera. The distance between the individual and the camera is approximately 4-5 meters. The experiment is conducted by observing the video and noting the detection results to determine whether or not the current image is classified accurately.

4.1. Comparison of Different Methods in Terms of Execution Time. To reduce the computational cost and stabilize the classifier output, this work also develops a posture change detection method to overcome these problems. Figure 7 compares the classifier output with and without the posture change detection approach. According to this figure, the output fluctuates less with the posture change detection method. Equation (12) defines the execution speed. The execution speed refers to how much time that a recognition algorithm

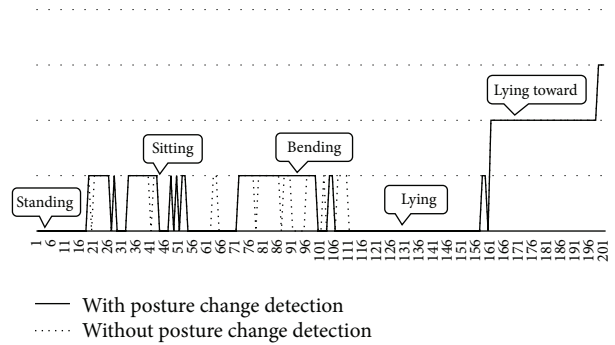


FIGURE 7: Comparison between output of classifier with and without posture change detection.

needs to finish a frame, regardless of whether or not the recognition is accurate:

$$\text{Execution speed (ms/frame)} = \frac{\text{Totally Execution time}}{\text{frames}}. \tag{12}$$

Table 1 compares different scenarios of the execution speed. Although the design in [5] has the shortest average execution time, owing to that it does not need to build the background model in this algorithm, the algorithm sacrifices the recognition accuracy of other postures. The design in [9] requires more execution time to build the background model than the proposed design.

4.2. Comparison of Different Methods in Terms of Recognition Rate. In our video clips, eight subjects of different heights

TABLE 1: Comparison of different methods in terms of the execution time.

Scenario	Method		
	Nasution and Emmanuel [9]	Silapasuphakornwong et al. [5]	Ours
Video 1 (ms)	85.25	15.00	54.26
Video 2 (ms)	88.00	15.40	60.26
Video 3 (ms)	99.34	15.27	73.23
Video 4 (ms)	111.23	15.66	82.77
Average execution time (ms)	95.96	15.33	67.63

TABLE 2: Recognition rates for various postures in the proposed system.

Posture	Recognition			
	N_a	N_c	N_f	R (%)
Standing	1459	1432	27	98.15
Sitting	419	391	28	93.31
Bending	634	603	31	95.11
Lying	297	297	0	100.00
Lying toward	236	216	20	91.53
Fall down	514	476	38	92.60

and weights were asked to participate in the project. The image resolution is 640×480 pixels. Four of the video clips were taken to train the templates of the classifier, and the remaining ones were taken to evaluate the system performance. In the posture change detection method, the threshold for the aspect ratio of the human body was set to 0.1, and the threshold for the height ratio of the human body was set to 0.15. In the height test, the threshold was set to 0.45. Table 2 summarizes the experimental results of the recognition rate for five postures in the proposed system, where N_a denotes the number of images for various postures; N_c represents the number of accurately detected images; N_f refers to the number of falsely detected images; and R denotes the recognition rate. According to this table, the best recognition appears in the lying posture with an accuracy rate of 100%, and the worst case appears in the supine posture with an accuracy rate of 91.53%.

Table 3 compares different algorithms in terms of the recognition rate. This table reveals that the design in [5] has the best execution time to complete the recognition of a frame. However, its recognition rate is extremely low especially in the bending posture with only a recognition rate of 9.30%. The design fails to recognize two postures (i.e., the sitting and supine postures) owing to the lack of background model. The design in [9] has higher recognition rates in the sitting and supine postures compared with others, because it has a more efficient foreground segmentation algorithm. According to Table 1, the design in [9] sacrifices the execution time for increasing the recognition accuracy. The worst case scenario for the posture recognition appears in the standing and binding postures. Their recognition rates are 76.56% and 63.25%, respectively. The proposed method can achieve 98.15% and 95.11% for recognizing the two postures. By using the average execution time of 95.96 ms, although the design

in [9] recognizes a frame, the proposed system uses only 67.63 ms to perform the same recognition. The proposed design can detect an individual falling down with an accuracy rate of 92.60%. In terms of execution time and recognition rate, the proposed system performs better than previous designs.

4.3. Performance Comparisons with Different Methods. This work also evaluates the performance of different fall detection methods by using a video recorded with three individuals, who participated in the project with different heights and weights. Our video clips contain 100 fall events and 100 false fall events. Two widely used criteria in fall detection systems are adopted here for comparisons [2, 6, 9, 15], as shown in the following:

$$\begin{aligned} \text{Sensitivity (\%)} &= \frac{TP}{TP + FN} \times 100\%, \\ \text{Specificity (\%)} &= \frac{TN}{TN + FP} \times 100\%, \end{aligned} \quad (13)$$

where the parameters TP, FP, FN, TN defined in (13) can be expressed as shown in Table 4.

Parameter TP refers to a subject having experienced a fall event, and the detection system can detect it accurately; however, a system having failed to detect a fall of a subject is denoted by parameter FN. Parameter TN denotes a subject having experienced a false fall, and the system can recognize it accurately. An event in which a subject experiences a false fall yet the system recognizes a fall as having occurred is denoted by parameter FP. These four parameters are used to calculate the times of the four possible recognitions. The criterion sensitivity refers to the accuracy rate of a system that can recognize the falls under all of the fall events in a video. The criterion specificity denotes the accuracy rate of a system that can recognize the false falls under all of the false fall events in a video. Table 5 reveals that the proposed system has a sensitivity of 96% and a specificity of 97%. This work has implemented the methods of Nasution and Emmanuel [9] and Rougier et al. [6], respectively. According to Table 6, the method in [9] has a higher sensitivity and specificity than those of the method in [6]. Nevertheless, the proposed system performs better than the two previous designs.

5. Conclusions

This work designs an indoor video surveillance system with fall detection capability. The proposed system can also detect

TABLE 3: Comparison of different methods in terms of recognition rate.

Posture	Method		
	Nasution and Emmanuel [9] (%)	Silapasuphakornwong et al. [5] (%)	Ours (%)
Standing	76.56	95.68	98.15
Sitting	99.28	None	93.31
Bending	63.25	9.30	95.11
Lying	66.00	100.00	100.00
Lying toward	100.00	None	91.53
Fall down	None	None	92.60

TABLE 4: Parameter definitions in (8) and (9).

System recognition	Fall incident	
	Occurs	Does not occur
Positive	TP	FP
Negative	FN	TN

TABLE 5: Experimental results of our system with four possible recognitions.

System recognition	Fall Incident	
	Occurs	Does not occur
Positive	94	3
Negative	6	97

TABLE 6: Comparison of fall detection methods.

Criterion	Method		
	Nasution and Emmanuel [9]	Rougier et al. [6]	Ours
Sensitivity (%)	83	40	94
Specificity (%)	88	63	97

and evaluate a human posture. An attempt is also made to improve the overall system performance by developing three methods to reduce the execution time for recognition and increase the recognition rate of human postures. The first method utilizes the mean ratio of the height in standing and supine postures to distinguish between these two postures. By using the posture change detection, the second method reduces the computational cost and stabilizes the classifier output. By using the height of a human body, the third method distinguishes between the bending and supine postures. Experimental results indicate that the proposed design can further reduce the execution time and increase the recognition rate of human postures. Additionally, the proposed system can achieve a recognition rate higher than 90% for each posture. Moreover, the proposed system can also detect the fall down with the accurate rate of 92.60%. Performance comparisons reveal that the proposed system performs better than previous designs. Efforts are underway in our laboratory to incorporate capabilities of multiobject tracing and face recognition in the proposed system.

Acknowledgments

This work is supported in part by Taiwan's Ministry of Education under Project no. 101B-09-027.

References

- [1] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [2] H. Foroughi, B. S. Aski, and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," in *Proceedings of the 11th International Conference on Computer and Information Technology (ICIT '08)*, pp. 219–224, Khulna, Bangladesh, December 2008.
- [3] H. Foroughi, A. Rezvanian, and A. Pazirae, "Robust fall detection using human shape and multi-class support vector machine," in *Proceedings of the 6th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '08)*, pp. 413–420, December 2008.
- [4] B. Huang, G. Tian, and X. Li, "A method for fast fall detection," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 3613–3618, Chongqing, China, June 2008.
- [5] P. Silapasuphakornwong, S. Phimolares, C. Lursinsap, and A. Hansuebsail, "Posture recognition invariant to background, cloth textures, body size, and camera distance using morphological geometry," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '10)*, pp. 1130–1135, Shangdong, China, July 2010.
- [6] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW '07)*, pp. 875–880, May 2007.
- [7] I. Haritaoglu, D. Harwood, and L. S. Davis, "GHOST: a human body part labeling system using silhouettes," in *Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1, pp. 77–82, August 1998.
- [8] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.
- [9] A. H. Nasution and S. Emmanuel, "Intelligent video surveillance for monitoring elderly in home environments," in *Proceedings of the IEEE 9th International Workshop on Multimedia Signal Processing (MMSP '07)*, pp. 203–206, Crete, Greece, October 2007.

- [10] F. Xie, G. Xu, Y. Cheng, and Y. Tian, "Human body and posture recognition system based on an improved thinning algorithm," *IET Image Processing*, vol. 5, no. 5, pp. 420–428, 2011.
- [11] M. Yu, S. M. Naqvi, A. Rhuma, and J. Chambers, "One class boundary method classifiers for application in a video-based fall detection system," *IET Computer Vision*, vol. 6, no. 2, pp. 90–100, 2012.
- [12] K. Ozcan, A. K. Mahabalagiri, M. Casares, and S. Velipasalar, "Automatic fall detection and activity classification by a wearable embedded smart camera," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 125–136, 2013.
- [13] M. Shoaib, R. Dragon, and J. Ostermann, "View-invariant fall detection for elderly in real home environment," in *Proceedings of the 4th Pacific-Rim Symposium on Image and Video Technology (PSIVT '10)*, pp. 52–57, November 2010.
- [14] G. Diraco, A. Leone, and P. Siciliano, "An active vision system for fall detection and posture recognition in elderly healthcare," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '10)*, pp. 1536–1541, Dresden, Germany, March 2010.
- [15] N. Noury, A. Fleury, P. Rumeau et al., "Fall detection—principles and methods," in *Proceedings of the 29th Annual International Conference of IEEE-EMBS, Engineering in Medicine and Biology Society (EMBC '07)*, pp. 1663–1666, Lyon, France, August 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

