Compressed-Domain Fall Incident Detection for Intelligent Homecare

Chia-Wen Lin*, Zhi-Hong Ling*, Yeng-Cheng Chang*, and Chung J. Kuo+

*Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan

+Components Business Group
Delta Electronics, Inc.
Taoyuan 333, Taiwan

Submitted to
Journal of VLSI Signal Processing- Systems for Signal, Image, and Video Technology: Special Issue on Audio-Visual Signal processing for Intelligent Security Systems

Submitted, August 2005
Revised, December 2005
Accepted, January 2006

Corresponding Author:
Prof. Chia-Wen Lin
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
Phone: 886-5-272-0411 ext. 33120
Fax: 886-5-272-0859
Email: cwlin@cs.ccu.edu.tw

Compressed-Domain Fall Incident Detection for Intelligent Homecare

Chia-Wen Lin*[1], Zhi-Hong Ling*, Yeng-Cheng Chang*, and Chung J. Kuo[+]

*Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan

[+]Components Business Group
Delta Electronics, Inc.
Taoyuan 333, Taiwan

Abstract -- This paper presents a compressed-domain fall incident detection scheme for intelligent homecare applications. For object extraction, global motion parameters are estimated to distinguish local object motions from camera motions so as to obtain a rough object mask. We then perform change detection and/or background subtraction on the DC+2AC images extracted from the incoming coded bitstream to refine the object mask. Subsequently, an object clustering algorithm is used to automatically separate the individual video objects iteratively. After detecting the moving objects, compressed-domain features of each object are then extracted for identifying and locating fall incidents. Our experiments show that the proposed method can correctly detect fall incidents in real time.

Index Terms- video surveillance, fall incident detection, home care, compressed-domain signal processing

[1] Corresponding author

1. INTRODUCTION

Electronic visual surveillance systems are an emerging application field involving multidisciplinary technologies spanning from image/video processing to communication, pattern recognition, and computer vision [1][2]. The ever-increasing demands on public area monitoring, transportation facilities (subways, highways, tunnels, etc.) monitoring, and indoor monitoring (homecare, home/office security, etc.) have been urging the development and deployment of new-generation visual surveillance systems. New-generation visual surveillance systems can benefit from new advances in digital video communication (video compression, bandwidth reduction, and convenient networking), digital video processing, and broadband access network infrastructures [3][4]. For example, digital video compression allows efficient transmission and recording of video events. Video enhancement algorithms can be used to enhance the quality of video under poor illumination conditions or low-resolution video captured by a low-cost camera. Video streaming and real-time video networking can provide flexible and ubiquitous video monitoring from remote locations. Automatic alarms can be generated and sent through networks or pagers to notify the users of abnormal situations. Research work on advanced video processing techniques for robust video transmission, color-video processing, event-based attention focusing, model-based sequence understanding in surveillance applications has been providing more and more interesting and useful features, thanks to the availability of low-cost high-performance computers, and mobile and fixed multimedia communications. In an intelligent visual surveillance system, it would be very helpful to provide features of automatically detecting and locating unusual events, such as, fall incident detection, intruder detection and tracking, and fire/smoke detection.

Automatically monitoring abnormal activities of the elderly and children using video cameras at home is an important issue for homecare. In the case of elderly people living on their own, there is a particular need for monitoring their behavior, such as a fall, unusual squatting, or a long period of inactivity. Falls amongst the elderly are particularly serious and often lead to injury, restricted activities, fear, or death. It is shown in [5] that 28-34% elderly people in the community experience at least one fall every year, and 40-

60% of the falls lead to injury. The main reasons elderly people become bedridden are apoplectic ictus, decrepitude, falls, and fractures [6]. Fall-related injuries have also been among the five most common causes of death amongst the elderly population [7]. The early detections and recording of fall incidents can help the elderly to obtain in-time medical treatments as well as help identify reasons of incidents while sustaining a fall.

Most of the existing fall detection schemes described in [6]-[9] propose to use specially designed sensors and circuitry, which may not be convenient for the elderly to wear or bring all the time. Recently, several computer vision based techniques, such as object tracking, behavior understanding and description, personal identification, and event detection, have been developed for visual surveillance and homecare applications due to the wide deployment of low-cost video cameras [4][10][11]. A few computer vision based methods have been proposed for detecting falls or other events at home. In [12], the authors propose a method of detecting portions of a video which are likely to contain a dynamic event from a compressed video. The events are assumed to happen in discontinuities in a motion field, or nonlinear changes of sizes in a moving region. Detection of specific events was not addressed in [12]. The method presented in [13] and [14] uses an omni-directional camera to track a video object modeled with an ellipse contour using a particle filter. The tracked object trajectory within different regions of a living room is analyzed by temporal segmentation so as to train and annotate the models of different activities using Gaussian mixture models (GMMs). Abnormal events such as falls and unusual inactivity can be classified using the trained GMMs. The method, however, may not be able to differentiate the activities of humans and pets or similar activities such as fall and squatting.

Many networked video cameras currently deployed are equipped with a video encoder in order to achieve efficient bandwidth consumption. Their computing power and storage capacity are, however, usually still rather limited due to the cost consideration. Because detecting events such as fall incidents in a video clip usually needs to process a sustained period of video data (e.g., 1~2 seconds for fall detection), pixel-domain processing would require a large size of frame buffer, leading to prohibitively expensive memory cost and high power consumption to the low-cost cameras. As a result, event detection often needs to be done by using video post-processing in a surveillance control

center, in which relatively powerful computers are equipped and videos are stored/received in compressed formats. Compressed-domain processing techniques are efficient in terms of computational complexity and storage cost because they can take advantage of the information already carried in a compressed video bitstream without the need of decoding the compressed video into pixel values, thereby drastically reducing the amount of data to be processed. Should the event detection be performed in a video camera, the camera can also use the information available in the compressed video bitstream (e.g., motion information and coding modes of macroblocks) to reduce the computation for event detection significantly.

In this work, we focus on compressed-domain fall incident detection schemes. The first task for vision-based fall incident detection is to detect human objects. There have been some research works for video object segmentation in the compressed domain [15]-[17]. For example, the method in [15] proposes an EM (Expectation-Maximization) approach to estimate the camera parameters so as to generate the object masks. Similarly, the method in [16][17] also proposes to extract object by applying the EM algorithm. Both the two methods utilize the motion vectors (MVs) available in a standard video encoder to segment object. However, the MVs are usually irregular and coarsely sampled, due to the use of "non-sophisticated" block matching motion estimation algorithm in generating the MV field, so the results of object segmentation may not be precise enough for the use of event detection.

In this paper, we propose a compressed-domain vision-based fall detection scheme for intelligent homecare applications. The proposed scheme can detect and track moving objects from a compressed video captured by a fixed or a pan-tilt-room (PTZ) camera in the compressed domain. In addition to using motion information to obtain an initial object segmentation mask, we propose to utilize DC+2AC image to perform change detection and/or background subtraction to refine the object mask. After detecting the moving objects, compressed-domain features of each object are then extracted for identifying and locating fall incident. The proposed system can also differentiate fall-down and squatting by taking into account the event duration. The main contributions of this work are three-fold. First, we propose a novel integral compressed-domain framework for fall incidents detection, involving compressed-domain object

segmentation, feature extraction, and statistical decision. Second, We introduce a new adaptive object mask refinement procedure using DC+2AC coefficients and DCT-MC for enhancing the resolution of segmentation so as to achieve better accuracy of event detection compared to the motion-based methods [15]-[17]. Third, we propose three useful feature parameters which can effectively identify falls and suggest a statistical method to determine appropriate thresholds for the feature parameters.

The remainder of this paper is organized as follows. Sec. 2 presents the proposed system architecture and describes the compress-domain feature extraction and distance metrics used in our work. Sec. 3 describes the proposed fall incident detection scheme. The experimental results are provided in Sec. 4. We present our concluding remarks in Sec. 5.

2. OVERVIEW OF THE PROPOSED SCHEME

Fig. 1 shows the conceptual diagram of the proposed intelligent networked visual surveillance system. The control center contains a server which is responsible for receiving compressed video bitstreams from mobile surveillance cameras, recording video data on the storage device, and managing the video access from remote users. The video captured by a camera is compressed using an MPEG-4 encoder, and the compressed video is subsequently sent to the server via UDP (User Datagram Protocol) packets. Remote users can access the surveillance video data ubiquitously using different multimedia terminal devices through the Internet. An automatic fall-incident detection scheme is implemented in the system for intelligent homecare applications.

The flowchart of the proposed compressed-domain fall incident detection scheme is given in Fig. 2. The proposed scheme involves two steps: compressed-domain object extraction and fall-down detection. At first, the MVs and the DC+2AC image [18] of each video frame are extracted from the incoming bitstream for subsequent processing. The MVs extracted from the incoming bitstream are fed into the Global Motion Estimation (GME) module to estimate the global motion (GM) parameters. As a result, the global motion and local object motion(s) are separated, and then those macroblocks with significant local motions are grouped together to obtain a rough object mask.

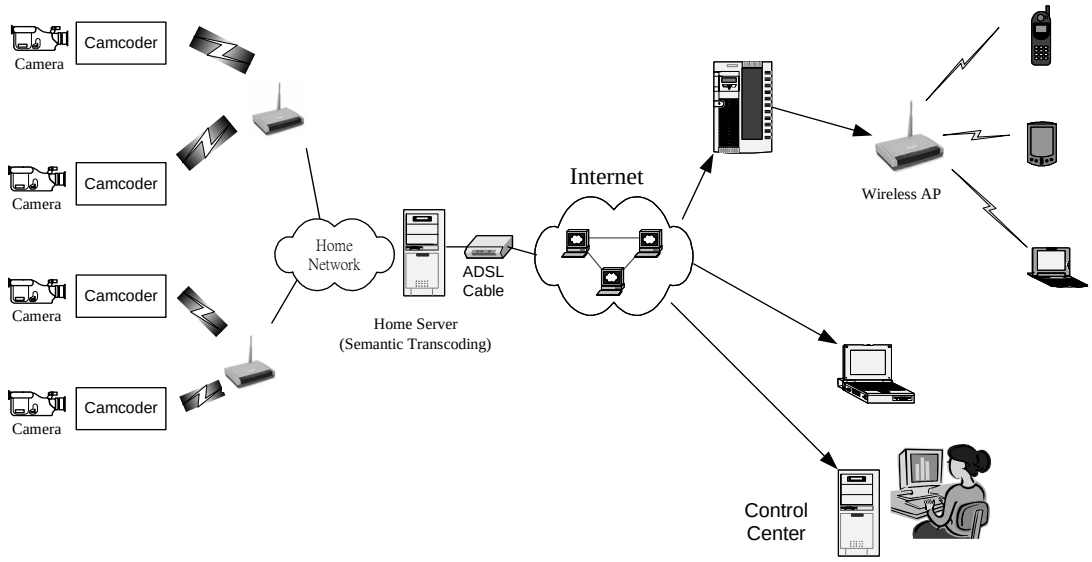


Fig. 1. Conceptual diagram of proposed intelligent networked home surveillance.

If the video shot contains global motion, the GM-compensated Change Detection operation is performed to refine the object mask. Otherwise, the Change Detection module is used to refine the object mask. For frames that contain more than a single object, the object clustering operation is performed to separate the object mask into multiple individual object sub-masks.

After extracting the video object, the fall-down detection module uses three feature parameters: the centroid of a human object, the maximum vertical projection histogram value, and the duration of an event to identify and locate fall-down events. Object tracking is activated in our method when the video has more than one object. The Object Labeling module is used to find the correspondence of video objects between two consecutive frames so as to obtain the associated feature parameters of each object.

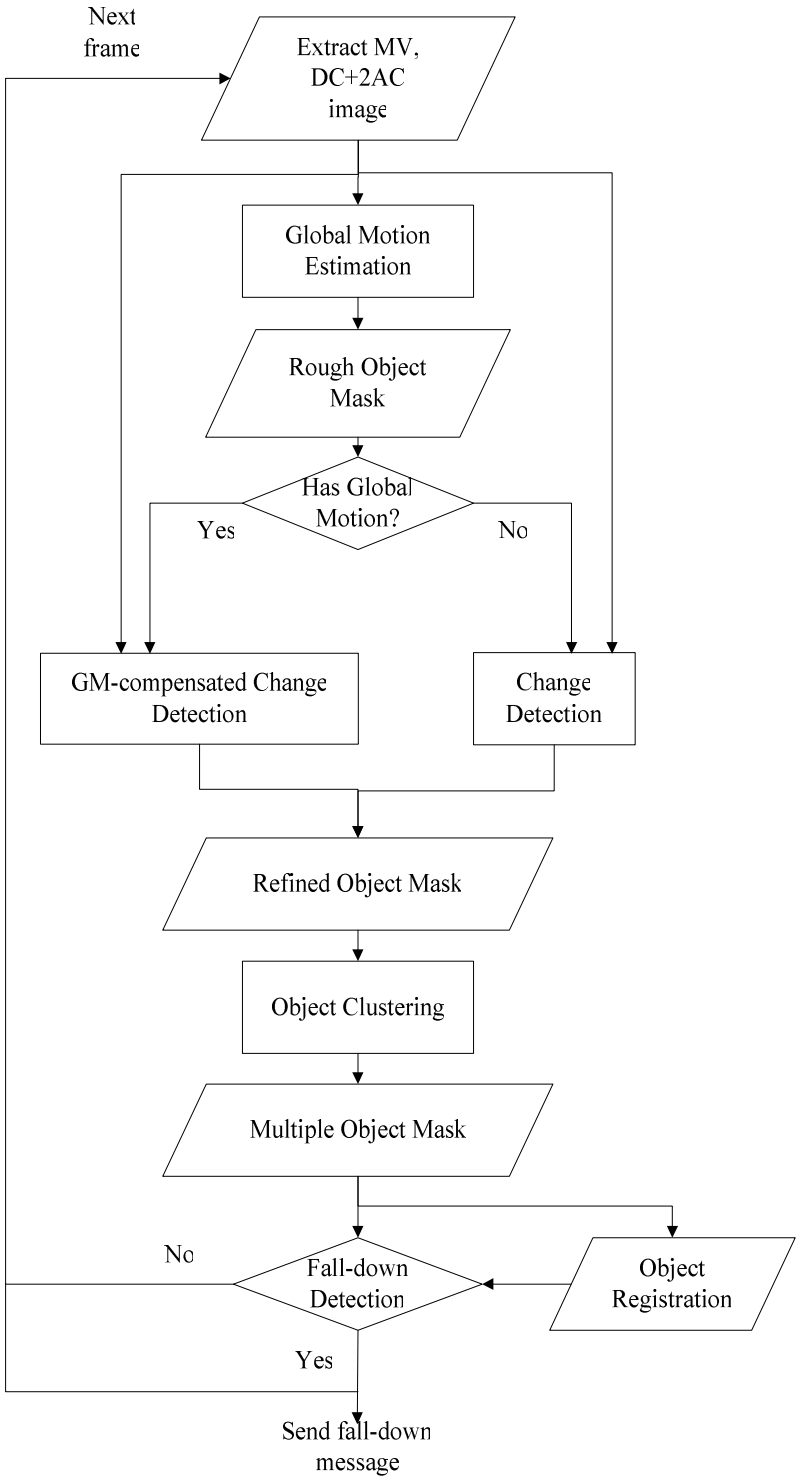


Fig. 2. Flowchart of the proposed compressed-domain fall incident detection scheme.

3. COMPRESSED-DOMAIN OBJECT EXTRACTION USING GLOBAL AND LOCAL MOTION INFORMATION

3.1. Initial Object Segmentation

In order to separate motion and local object motions, the global motion needs to be estimated first. In this work, we modify the compressed-domain GME method proposed in [19] to estimate the GM parameters between two consecutive video frames using the coarsely sampled macroblock MVs carried in the compressed video. In our method, the incoming MVs are first filtered using a 2-D median filter with a 3×3 mask to remove the noise due to the inaccurate block-wise motion estimation performed in video encoding. The global motion is then obtained by minimizing the fitting error between the input MVs and the MVs generated from the estimated motion model using the Newton-Raphson method with outlier rejections [19]. The six-parameter affine model shown in (1) is adopted to estimate the GM parameters.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \end{bmatrix} \tag{1}$$

where (x',y') and (x,y) represent the pixel coordinates in the reference frame and the current frame, respectively; [a_1, a_2, a_3, a_4, a_5, a_6] denotes the set of GM parameters.

Outlier rejection can improve the robustness of GME by removing the unreliable MVs which tend to have largest fitting errors from the data set for GME [19]. However, because the object mask size may change largely in time, the method in [19], which adopts a fixed threshold for outlier rejection, may not be accurate enough. In our method, the outlier regions are initially defined as those blocks with the largest MV fitting errors in the first iteration. In the second and later iterations, the outlier regions are instead defined as the local moving object macroblocks extracted. In each iteration, the GM parameters are estimated first. Macroblocks with MVs significantly different from the estimated global motion are subsequently classified as belonging to local moving objects using the following rule:

$$\mathrm{MB}_m = \begin{cases} Object & \text{if } \left|MVx_m - MVx_m^{\mathrm{GM}}\right| + \left|MVy_m - MVy_m^{\mathrm{GM}}\right| > TH_{\mathrm{GM}} \\ Background & \text{otherwise} \end{cases} \quad (2)$$

where MB_m denotes the segmentation mask of the mth macroblock. (MVx_m, MVy_m) represents the incoming MV of the mth macroblock. TH_{GM} is a threshold, which is set empirically and fixed in every iteration of GME in our implementation. It can also be made adaptive according to the statistics of fitting errors of extracted object macroblocks. (MVx_m^{GM} , MVy_m^{GM}) represents the MV of the mth macroblock mapped from the GM parameters as calculated by (3).

$$\begin{cases} MVx_m^{\mathrm{GM}} = \left(a_1 x_m + a_2 y_m + a_3\right) - x_m \\ MVy_m^{\mathrm{GM}} = \left(a_4 x_m + a_5 y_m + a_6\right) - y_m \end{cases} \quad (3)$$

where (x_m, y_m) stands for the spatial coordinate of the mth macroblock.

3.2. Object Mask Refinement

After the initial classification, we obtain a rough object mask with granularity of 16×16 pixels (i.e., the macroblock size). Such granularity, however, may be too coarse to represent the object shape with enough accuracy for the subsequent fall-incident detection. To achieve finer granularity, we propose to refine the segmentation result by using the change detection masks (CDMs) of DC+2AC images [18]. As shown in Fig. 3, the CDM-based refinement procedure is divided into two parts: one performing change detection by taking the previous frame as the reference frame, whereas the other performing background subtraction that takes the background frame as the reference. Using the previous frame as the reference frame for change detection usually performs well when there are significant object movements. However, should there be no significant object movement, the change detection scheme may fail; instead, the background subtraction scheme can be used to cope with such situation. According to our observations, if an object has significant movement, its corresponding object sizes in CDMs of the current frame ($SIZE_{\mathrm{CDM}}^n$) and previous frame tend to be close. Otherwise, $SIZE_{\mathrm{CDM}}^n$ would be

significantly smaller than $SIZE_{\mathrm{CDM}}^{n-1}$. Based on this, the following rule is used to determine whether or not a video object has moved.

if $(SIZE_{\mathrm{CDM}}^{n} > K_{\mathrm{SIZE}} \times SIZE_{\mathrm{CDM}}^{n-1}) \;\&\&\; (SIZE_{\mathrm{CDM}}^{n} > TH_{\mathrm{SIZE}})$

Use the CDM and background subtraction for object refinement

else

Perform background subtraction, and use the result for refinement

where K_{SIZE} and TH_{SIZE} are two parameters obtained empirically. The CDM obtained in the above procedure is used to refine the object masks. The extracted background information is subsequently used to update the background frame memory for use in processing the subsequent frames.

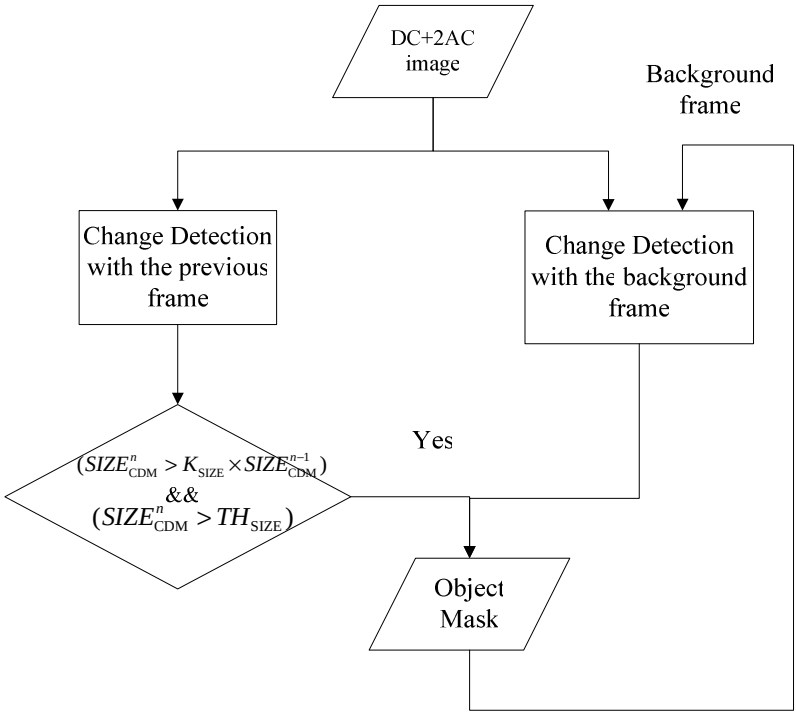


Fig. 3. Block diagram of object segmentation mask refinement.

For a video clip containing global motion, global motion compensation should be performed prior to the CDM-based refinement. Otherwise, most part of non-still background may be misclassified as moving objects. Before extracting the DC+2AC

image, we have to compute all the DCT coefficients of the current frame using the DCT-domain motion compensation (DCT-MC) scheme [20]. After extracting the DC+2AC image from the GM-compensated DCT coefficients, The CDM obtained by using the previous frame as the reference frame is then used to refine the object segmentation masks.

The CDM-based refinement procedure is described below. First, the CDM is refined to the granularity of 4×4 pixels (SEG_{CD}) using the extracted DC+2AC DCT coefficients, while the rough object mask obtained from the GME module is also enlarged to the same granularity (SEG_{GME}). If the objects move significantly, we consider both the two masks, SEG_{CD} and SEG_{GME}, are reliable enough. Otherwise, only SEG_{CD} is considered reliable. In the case of no significant object movements, the MVs of object and background macroblocks are almost the same, thus SEG_{GME} may be unreliable. We use the average MV magnitude to determine which object masks should be used to obtain the refined object mask as described in (4).

$$SEG_{\mathrm{final}} = \begin{cases} SEG_{\mathrm{CD}} & \text{if } \frac{1}{N_{\mathrm{obj}}} \sum_{MB_i \in \mathrm{object}} \left(\left| MVx_i - MVx_i^{\mathrm{GM}} \right| + \left| MVy_i - MVy_i^{\mathrm{GM}} \right| \right) < TH_{\mathrm{MV}} \\ SEG_{\mathrm{GME}} \,\&\, SEG_{\mathrm{CD}} & \text{otherwise} \end{cases} \quad (4)$$

where N_{obj} represents the number of object macroblocks in a frame; TH_{MV} denotes the threshold of the average MV magnitude of object macroblocks.

3.3. Object Clustering and Labeling

Since a video frame may have multiple moving objects, after the above refinement procedure, an iterative object clustering algorithm is performed to automatically separate individual objects by clustering the foreground macroblocks with distinct local motions from the refined segmentation mask. In the clustering algorithm shown in Fig. 4, morphological filtering is first performed on the binary object mask to fill small holes and remove noise. Connected component labeling [21] is then used to obtain a labeled image in which the value of each pixel is the label of its connected components. Figs. 5(a) and (b) illustrate a binary object mask and its corresponding connected components labeled

image, respectively. The local motion model of a cluster of macroblocks with the same label is used to verify whether or not the object group has more than one object. Object macroblocks with homogeneous local motions and spatially contiguous locations are grouped as an object iteratively until all the objects are extracted.

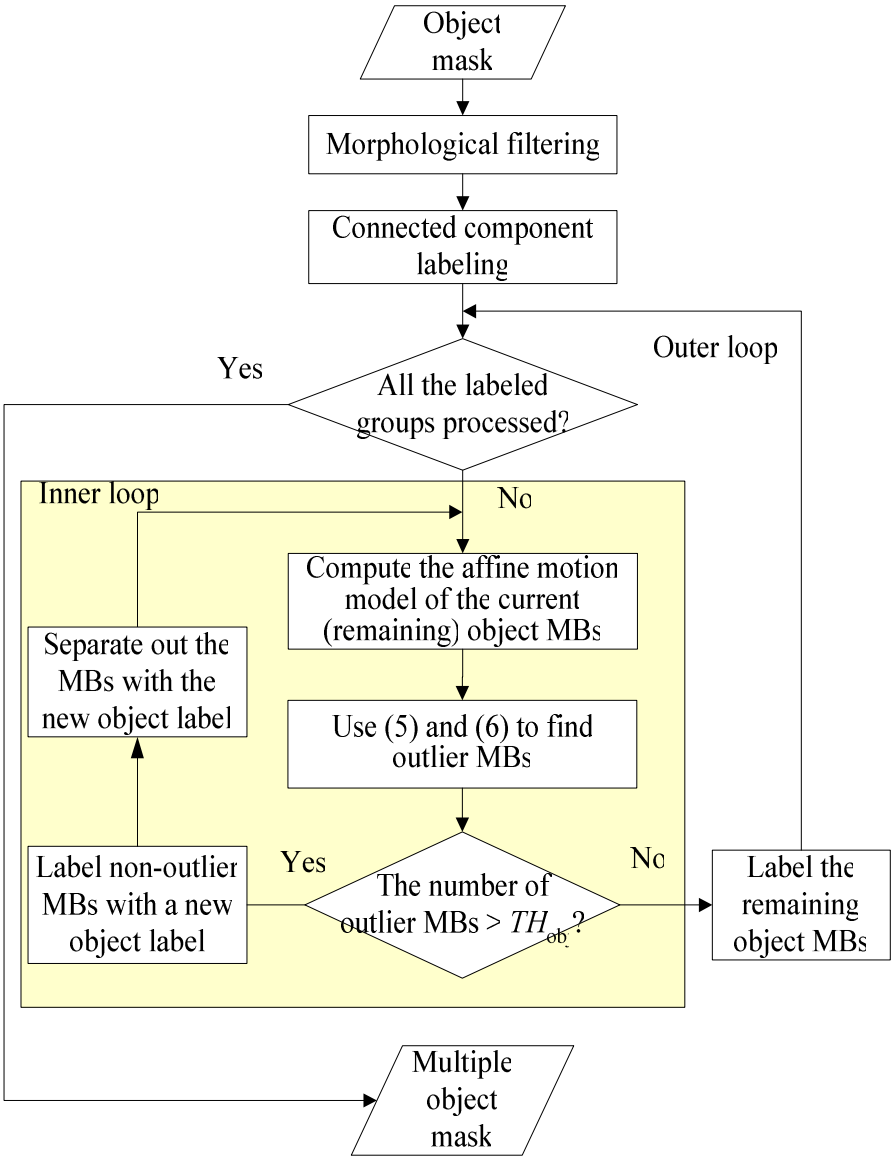


Fig. 4. Flowchart of the proposed object clustering scheme.

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
0	0	0	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

(a)

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
0	0	0	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

(b)

Fig. 5. An example of connected component labeling: (a) a binary object mask and (b) the corresponding labeled image.

After labeling connected components, if object macroblocks in the object mask are labeled with different labels, they are considered as belonging to different objects and thus should be dealt with separately. The proposed object clustering algorithm comprises two iteration loops: the outer-loop iteratively handles individual object groups of the labeled macroblocks; whereas the inner-loop recursively clusters the object group labeled with a same label. The inner-loop first estimates the object motion using the six-parameter affine model of a classified object group as follows:

$$\begin{cases} OMVx_{i,n} = b_1^i x_{i,n} + b_2^i y_{i,n} + b_3^i - x_{i,n} \\ OMVy_{i,n} = b_4^i x_{i,n} + b_5^i y_{i,n} + b_6^i - y_{i,n} \end{cases} \quad (5)$$

where $(OMVx_{i,n}, OMVy_{i,n})$ represents the MV of the *n*th macroblock of the *i*th object group mapped from the object motion model; $\{b_1^i, b_2^i, b_3^i, b_4^i, b_5^i, b_6^i\}$ denotes the set of affine motion parameters of the *i*th object group obtained by least squares estimation using MVs of macroblocks belonging to this group; ($x_{i,n}$, $y_{i,n}$) represents the spatial coordinate of the *n*th macroblock of the *i*th object group.

The local object motion is then used to determine which macroblocks are outliers (i.e., those with MVs that are significantly different from the object motion model) as shown in (6). If the number of outlier macroblocks is greater than the threshold TH_{obj}, implying that the object group has more than one object, the object macroblocks with MVs which conforms to the object motion model is labeled with a new object label. The remaining macroblocks are then iteratively processed until all the video objects are

separated. When all the objects are separated form the *i*th object mask, the process jumps to the outer loop to deal with the next object mask. As a result, we can obtain the multiple object masks.

$$Obj_{i,n}^{\text{new}} = \begin{cases} 1 & \text{if } \left|OMVx_{i,n} - MVx_{i,n}\right| + \left|OMVy_{i,n} - MVy_{i,n}\right| > TH_{\text{OM}} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where ($MVx_{i,n}$,$MVy_{i,n}$) represents the MV of *n*th macroblock of the *i*th object group; TH_{OM} is the threshold for outlier classification.

if $(\sum_{n} Obj_{i,n}^{\text{new}} > TH_{\text{obj}})$

Non-outlier macroblocks are labeled with a new object label. The remaining outlier macroblocks are processed iteratively in the inner loop.

else

Jump to the outer loop to process the next object mask.

After all objects in a frame have been clustered and labeled, the motion model of a labeled object in the current frame is used to find the object's counterpart in the previous frame. The shape and location of a current's frame object in the previous frame are first estimated by using the object's motion model obtained by (5). The object's best match in the previous frame is determined by finding the object which has the maximum overlapping area with the estimated object mask provided that the overlapping area exceeds a predetermined threshold (e.g., 50% of the object size). The color histograms of DC+2AC images of the two corresponding objects can be further compared to ensure the correctness of correspondence, while the complexity will be increased. If the mapping between the two frames' objects is one-to-one and onto, we assume all the correspondences are correct and there is no occlusion (e.g., without object merge/split and new/vanishing objects). Otherwise, we use the relationship defined in Table I to identify the occlusion states: an object leaving a frame, an object entering a frame, merging of multiple objects into a single object, and splitting of an object into multiple objects, as listed in Table II [22].

Table I

Relationship between object i in the current frame and object j in the previous frame. R_{ij} = 1 if object i and object j overlap with each other; otherwise R_{ij} = 0

Object in the current frame \ Object in the previous frame	1	…	j	…	N	SOR
1	R_{11}	…	R_{1j}	…	R_{1N}	$SOR_1 = \sum_{j=1}^{N} R_{1j}$
⋮	⋮	…	⋮	…	⋮	⋮
i	R_{i1}		R_{ij}		R_{iN}	$SOR_i = \sum_{j=1}^{N} R_{ij}$
⋮	⋮		⋮		⋮	⋮
M	R_{M1}		R_{Mj}		R_{MN}	$SOR_M = \sum_{j=1}^{N} R_{Mj}$
SOC	$SOC_1 = \sum_{i=1}^{M} R_{i1}$	…	$SOC_j = \sum_{i=1}^{M} R_{ij}$	…	$SOC_N = \sum_{i=1}^{M} R_{iN}$	

Table II

The corresponding values of SOC and SOR to object states.

$SOR_i = 0$	A new object *i* enters the current frame
$SOC_j = 0$	Object *j* leaves the current frame
$SOR_i \geq 2$	Multiple objects merge into object *i*
$SOC_j \geq 2$	Object *j* is split into multiple objects

In Table I, R_{ij} = 1 if object i in the current frames corresponds to object j in the previous frame; otherwise R_{ij} = 0.. We can know the states and positions of objects according to Tables I and II. In Table I and Table II, two indices, SOC (Sum Of Column) and SOR (Sun Of Row) as defined in (7) and (8), are used to characterize the relationship of labeled objects between two consecutive frames.

$$SOC_j = \sum_{i=1}^{M} R_{ij} \quad , j = 1,...,N \tag{7}$$

$$SOR_i = \sum_{j=1}^{N} R_{ij} \quad ; i = 1,...,M \tag{8}$$

where M and N represent the numbers of objects in the current and previous frames, respectively. The SOC and SOR values correspond to different object states as listed in Table II.

4. FEATURE-BASED FALL INCIDENT DETECTION FROM THE OBJECT MASKS

To identify and locate a fall incident of a person, we found that three features can be used to effectively capture fall-down events according to our experiments. First, as illustrated in Fig. 6(a), a fall incident usually occurs in a short time period with a typical range of 0.4s~0.8s. Second, Fig. 6(b) depicts that a falling person's centroid changes significantly and rapidly during the falling period. Third, the vertical projection histogram is also a useful feature for detecting a fall-down event because the vertical projection histogram of a falling person also changes significantly during the falling period as shown in Fig. 6(c).

In order to obtain the three feature values: the centroid of a human object, the vertical projection histogram, and the duration of an event detected, the human objects need to be extracted using the proposed compressed-domain segmentation method. After extracting a foreground object, the vertical projection histogram of the object is computed as follows.

$$H(x,y)=\begin{cases}1 & \text{if } (x,y) \text{ is an object pixel}\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

$$V(x)=\sum_{y} H(x,y) \tag{10}$$

Since $V(x)$ in (10) is an one-dimensional distribution, we can use some distance metrics, such as the Bhattacharyya distance [23] in (11), to measure the similarity of two vertical projection histograms (e.g., $V_1(x)$ and $V_2(x)$) of video frames within a sliding time window so as to identify significant changes of vertical projection histogram in contiguous frames due to fall incidents.

$$d(V_1,V_2)=\sum_{x}\sqrt{\frac{V_1(x)}{\sum_u V_1(u)}\frac{V_2(x)}{\sum_v V_2(v)}} \tag{11}$$

However, the computational complexity of computing (11) is high. To reduce the computation, we propose to use the maximum of a vertical projection histogram defined in (12), which maps the vertical project histogram into a single value.

$$V_{\max}=\max_x V(x) \tag{12}$$

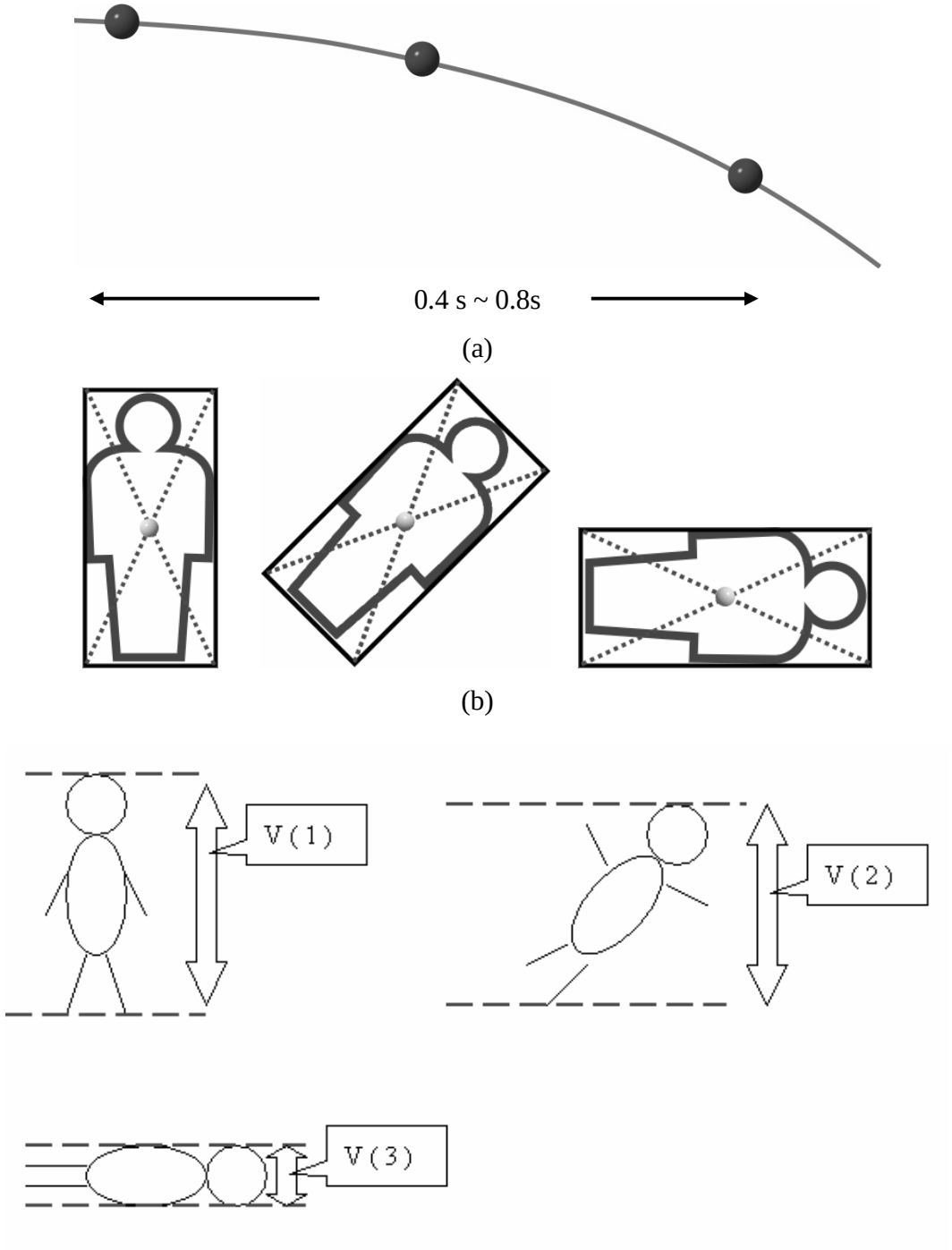
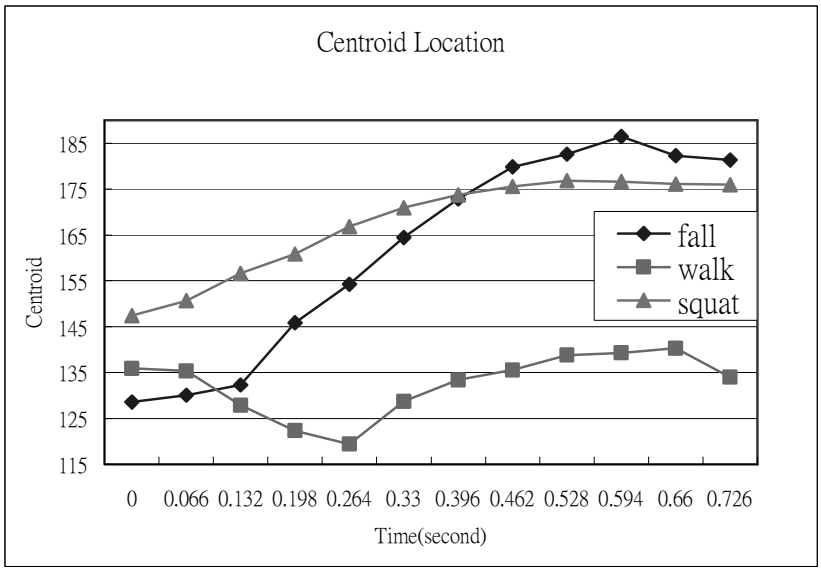
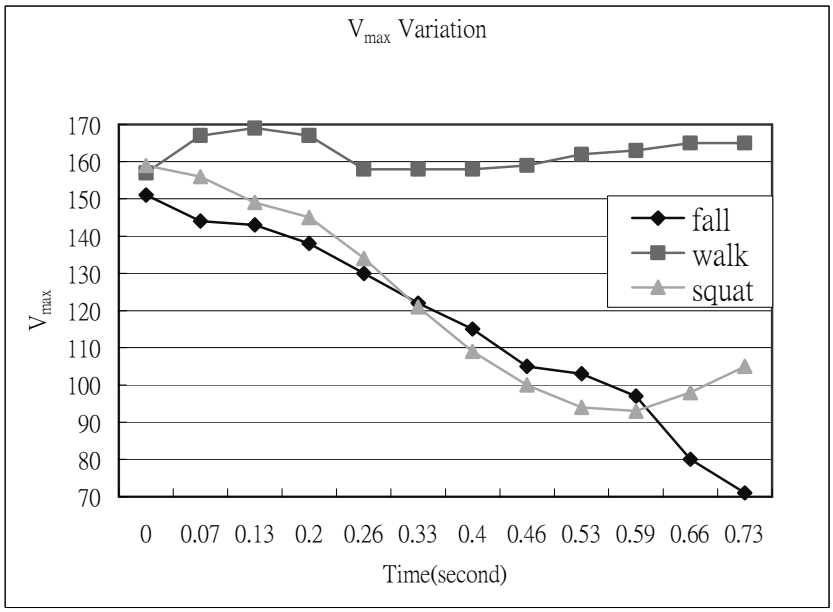


Fig. 6. Three features used for detecting a fall incident: (a) the duration of an event; (b) The location and change rate of the centroid of the human object; (c) the vertical projection histogram of the human object.



(a)

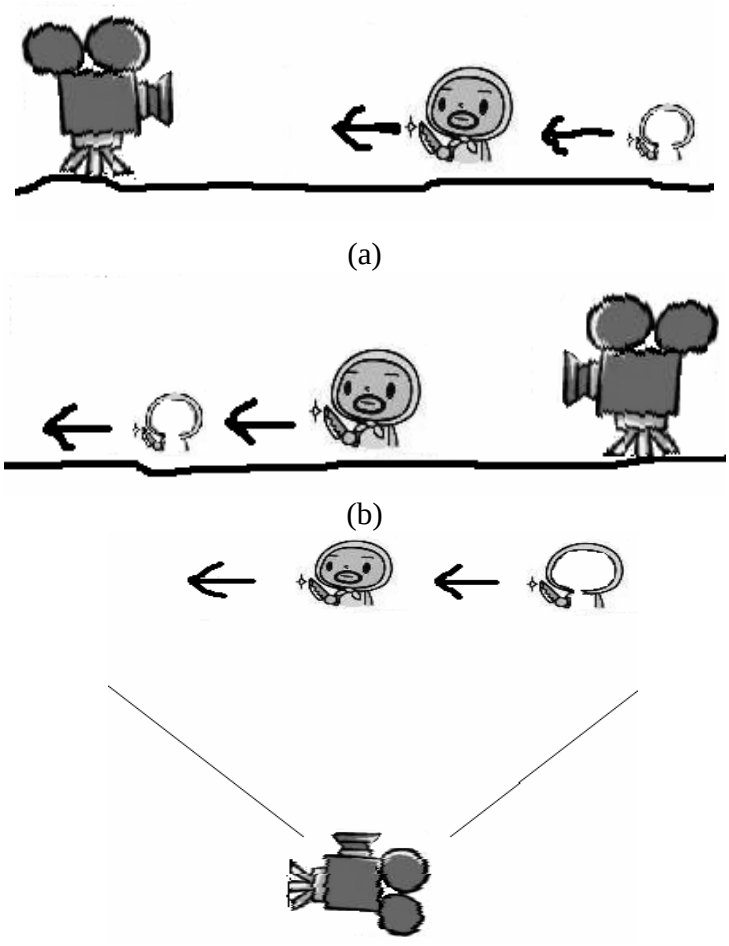


(b)

Fig. 7. Comparison of two feature values for a normal-walking person and a falling down person and a squatting person: (a) the centroid locations of objects versus time; (b) the vertical projection histogram values of objects versus time.

Fig. 7 compares the centroid locations and the V_{max} values of three different cases: walking, squatting, and falling. We can see that both feature values change significantly and repidly during the falling period. In this example, the centroid locations before and after falling down are 128 and 186, respectively. The V_{max} values before and after falling

down are 151 and 70, respectively. The duration of the event is about 0.59 s which is within the typical time range of a fall-down event.



(a) (b) (c)

Fig. 8. Three different motion types: (a) a person going toward the camera; (b) a person going away from the camera; (c) a person walking horizontally in front of the camera.

Because the above two feature values may vary with different object locations and object sizes, adopting fixed threshold values are not appropriate. We propose to use the following feature vector consisting of two normalized feature values for fall incident detection. The two feature values also take into account the effect of event duration.

$$\mathbf{x}(n) = \left[\frac{\left| f_{\text{cent}}(n-SW) - f_{\text{cen}}(n) \right|}{f_{\text{cent}}(n-SW)} \quad \frac{\left| V_{\max}(n-SW) - V_{\max}(n) \right|}{V_{\max}(n-SW)} \right]^T \tag{13}$$

where $f_{\text{cent}}(n)$ represents the location of the object centroid in the nth frame; $V_{\max}(n)$ denotes the maximum of vertical projection histogram of the object in the nth frame; SW

stands for the length of sliding window, which is in the typical range of a fall incident's duration.

The relation between the direction of a moving object and the camera would affect the distribution of feature values extracted for fall incident detection. Fig. 8 illustrates three types of object motions: the first type is a human object going toward the camera; the second is the human object going away from the camera; whereas the third is the object walking horizontally in front of the camera. Other types of motions can be represented as the combinations of Type 1 and Type 3 or the combinations of Type 2 and Type 3. Because the distributions of feature vectors with different motion types are different as will be shown later, we use different threshold values for the three motion types, respectively.

Squatting has similar behavior with falling in terms of the centroid location. However, the change rates of the centroid of squatting is much slower than those of fall incidents as illustrated in Fig. 7. The characteristics can be used to differentiate normal squatting events (slow change rate) from fall-down events (fast change rate) by choosing appropriate thresholds. Typically, falling and squatting have significantly different centroid changes (128→186 for falling and 147→176 s for squatting, respectively). Using appropriate threshold can detect these two events as well as achieve good differentiation accuracy.

5. EXPERIMENTAL RESULTS

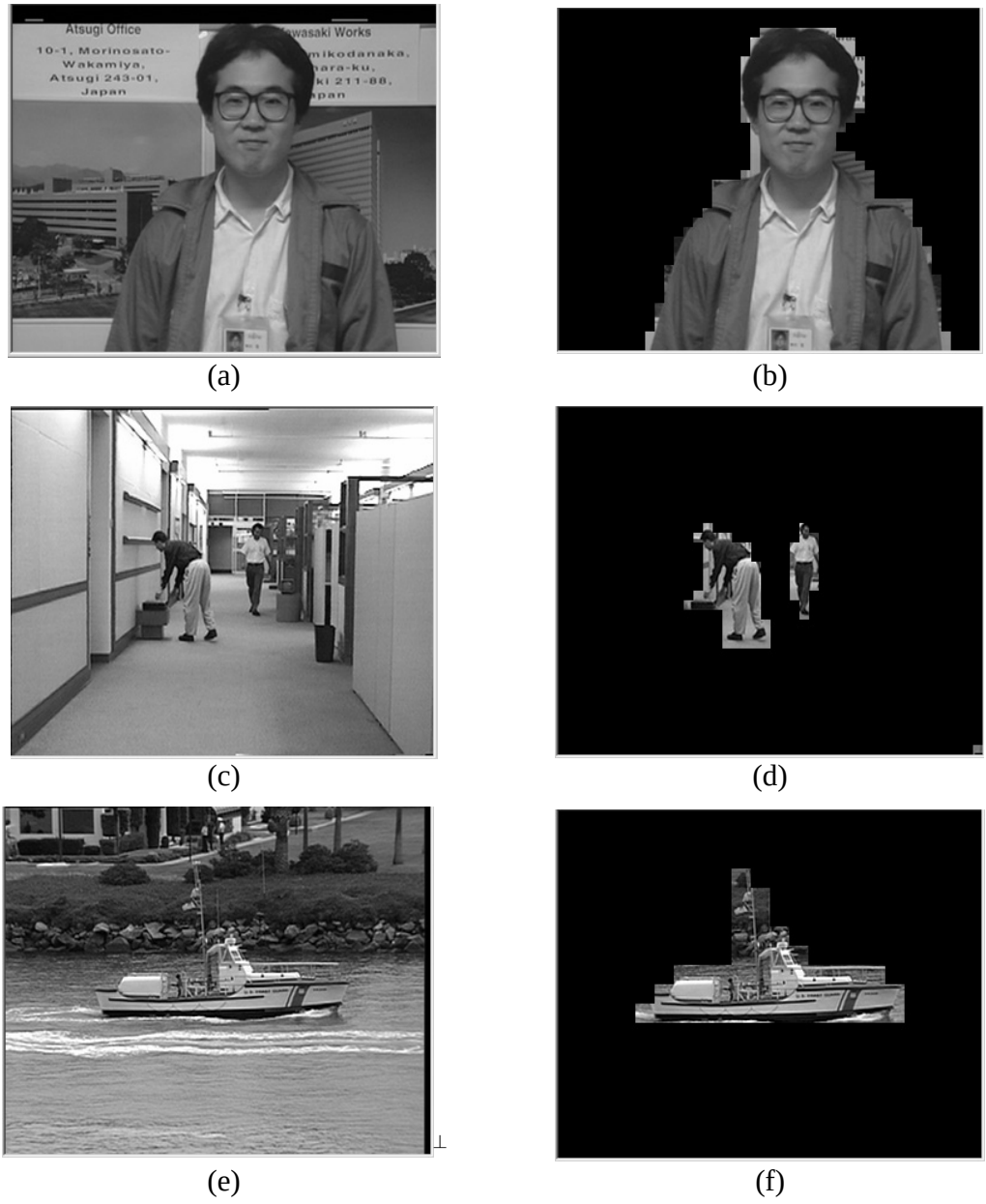
Three CIF (352×288) test sequences: *Pamphlet* (one object without global motion), *Hall* (two objects without global motion), and *Coastguard* (two objects with global motion), were encoded using an MPEG-4 encoder as the inputs to evaluate the proposed compressed-domain object segmentation scheme. Fig. 9 depicts three snapshots and the corresponding segmentation masks of the three test sequences, respectively. We compare the extracted object masks with the ground-truth masks to calculate three performance indices for each frame: the number of missing blocks, the number of false positive blocks, and the average correctness ratio. The following objective metric presented in [24] is used to evaluate the average correctness ratio of object segmentation:

$$d(M_t^{\text{ref}}, M_t^{\text{seg}}) = 1 - \frac{\sum_{(x,y)} M_t^{\text{ref}}(x,y) \oplus M_t^{\text{seg}}(x,y)}{\sum_{(x,y)} M_t^{\text{ref}}(x,y)} \tag{14}$$

where M_t^{ref} represents the ground-truth mask of the *t*th frame; M_t^{seg} represents the extracted object masks of the *t*th frame; (*x*,*y*) denotes the index of block location.

Note that the ground-truth masks are of 4×4 block-wise granularity, rather than pixel-wise accuracy. This is because the objects are extracted in the compressed domain without being decoded into pixel values. Therefore we cannot obtain object shapes with pixel-wise accuracy. Generally, block-wise accuracy is good enough for object-based event detection for video surveillance applications.

Fig. 10 and Table III show the performance of the proposed compressed-domain segmentation method. Fig. 10 depicts the numbers of object blocks, missing blocks, and false positive blocks counted for each frame of three test sequences, respectively. Since there is only one single object for the *Pamphlet* sequence and no camera motion in the sequence, we obtain relatively better segmentation accuracy (less missing blocks and false positive blocks and higher correctness ratio) compared to the results for the other two sequences. As for the *Hall* sequence, because a new object appears since the 78th frame, the number of the missing blocks increases a little bit since then as shown in Fig. 10(b). Because the *Coastguard* sequence contains global motion and two objects, the segmentation accuracy is relatively lower, but is still good enough. The average correctness ratios for the three sequences are 92.8%, 71.1%, and 75%, respectively as listed Table III. The experiments were performed on an AMD Athlon 1GHz PC. The processing speed is about 13-18 CIF fps, depending on the motion characteristics of sequences.

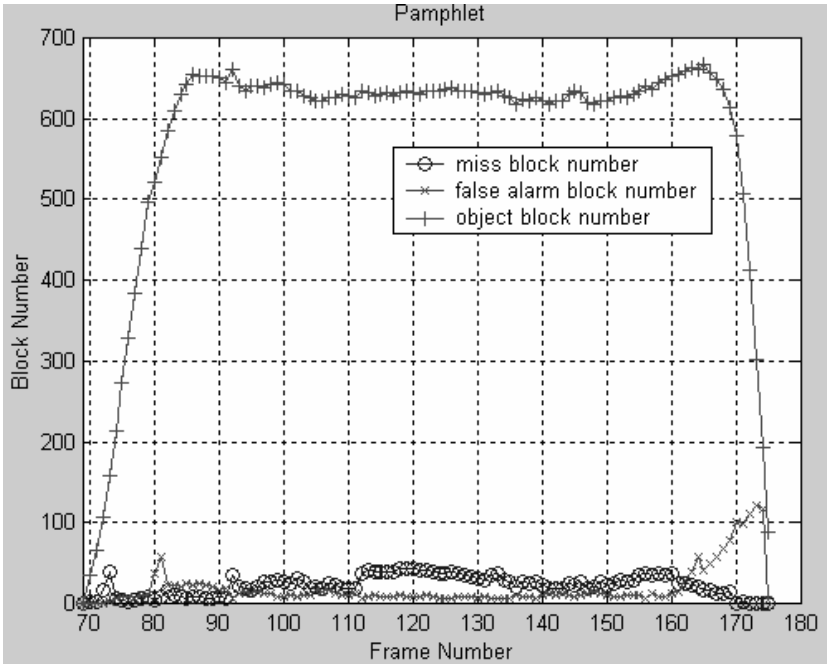


(a) (b)

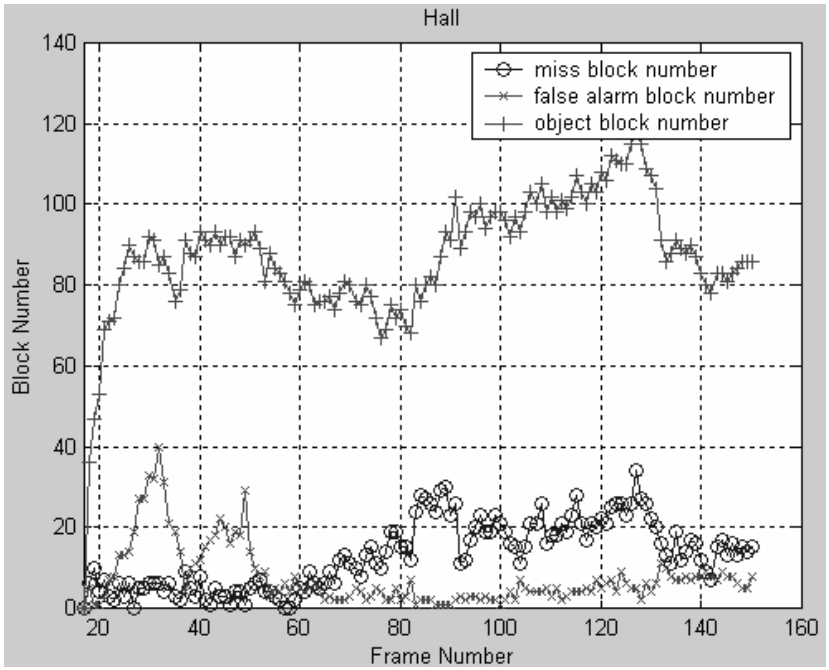
(c) (d)

(e) (f)

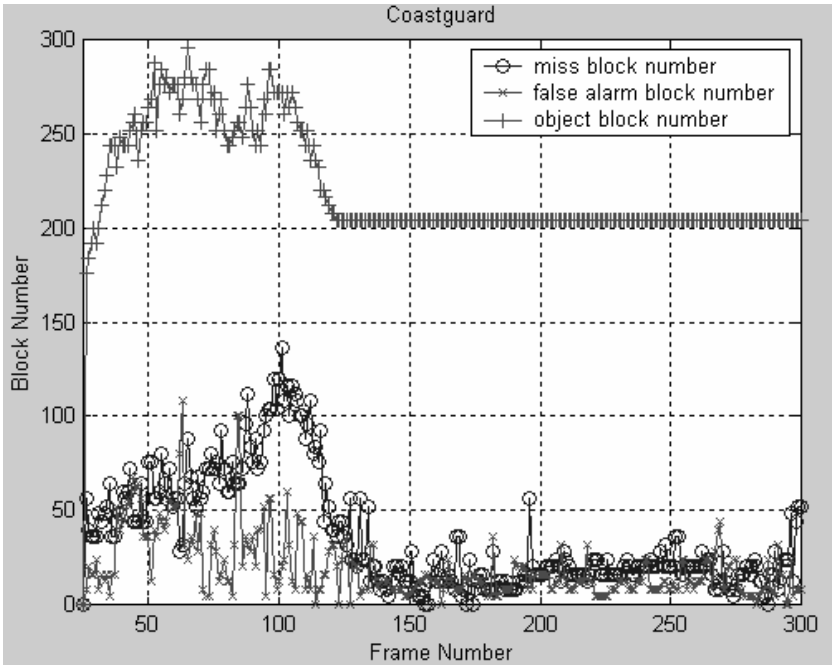
Fig. 9. Snapshots and the resulting segmentation masks of the three test sequences: (a)-(b) *Pamphlet*; (c)-(d) *Hall*; and (e)-(f) *Coastguard*.


Pamphlet
miss block number
false alarm block number
object block number
Block Number
Frame Number
0
100
200
300
400
500
600
700
70
80
90
100
110
120
130
140
150
160
170
180

(a)


Hall
miss block number
false alarm block number
object block number
Block Number
Frame Number
0
20
40
60
80
100
120
140
20
40
60
80
100
120
140
160

(b)



(c)

Fig. 10. Frame-by-frame segmentation performance indices for three test sequences: (a) *Pamphlet*, (b) *Hall*, and (c) *Coastguard*.

Table III
Performance evaluation of the proposed compressed-domain object segmentation method for three test sequences

Sequence	Average # of object blocks per frame	Average missing ratio	Average false-positive ratio	Average ratio of correctness
Pamphlet	559	3.8%	3.4%	92.8%
Hall	87	14.9%	8.0%	77.1%
Coastguard	220	16.4%	8.6%	75.0%

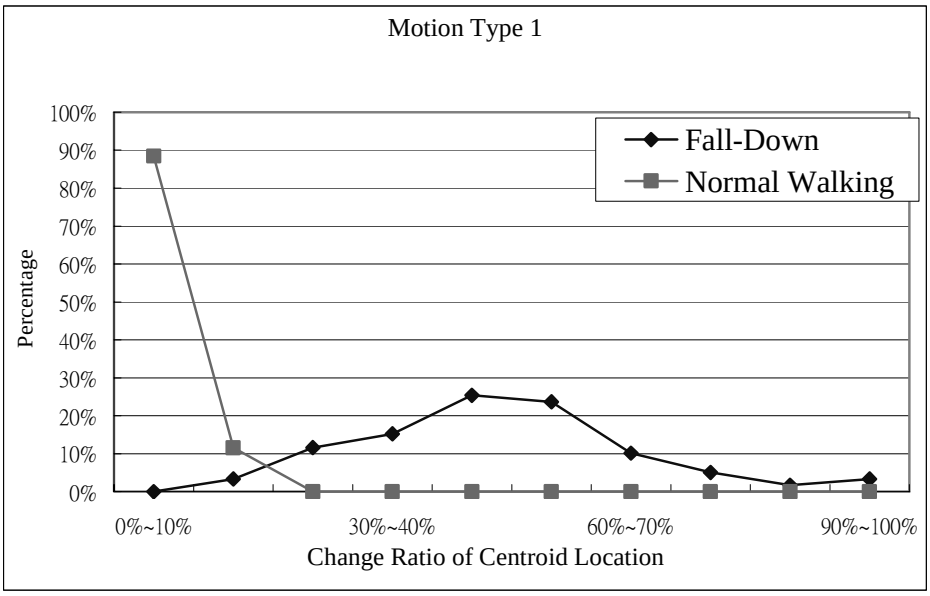
For fall incident detection, totally 78 sequences including 48 training sequences and 30 test sequences were used in our experiments. The 48 training sequences containing three different motion types (16 sequences for each motion type) were used to determine the thresholds for the three motion types, respectively. Among every 16 sequences for each motion type, eight sequences consist of fall incident events, whereas the other eight sequences contain no fall incidents. Fig. 11 depicts the statistical distributions of the

centroid location and V_{max} change ratios collected from the training sequences for the three motion types, respectively. The change ratios are calculated according to (13) with a sliding interval of 0.6 second between two frames. As shown in Fig. 11, the two normalized feature values (i.e., the horizontal axes) are both divided into 10 bins, each containing 10% of the whole range. For each object motion type, we choose a threshold for each feature value. Each threshold is chosen to minimize the error rate of event detection according to the associated distribution in Fig. 11. The thresholds for the three motion types are listed in Table IV. Since the motion behavior of a human may be a combination of two of the three motion types, we use a linear combination of the two corresponding thresholds to calculate the threshold according to the motion types determined by using the trajectory of centroid and the change rate of object height.

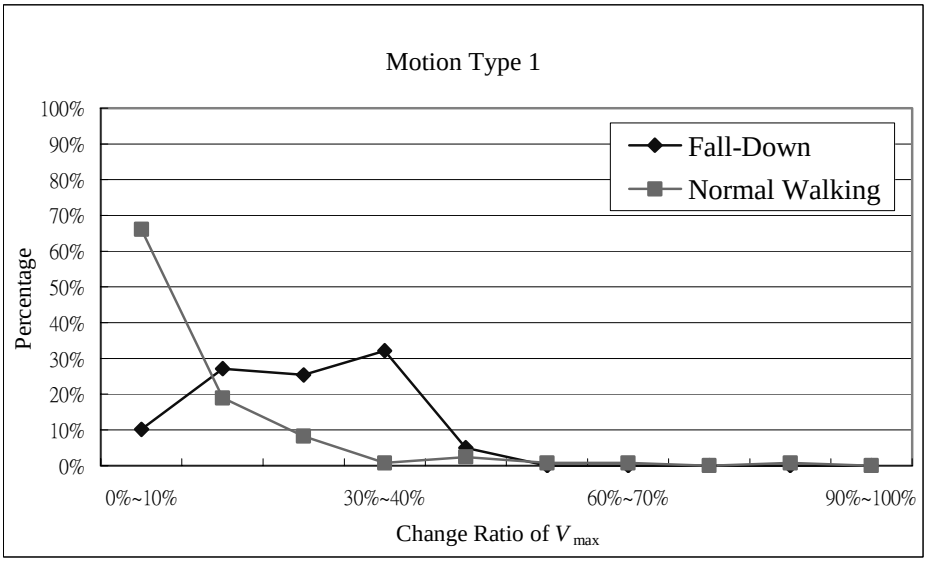
We used 30 test sequences with different motion types of fall incidents to evaluate the performance of the proposed fall-incident detection algorithm. These sequences consist of 15 sequences with fall incidents and 15 normal walking sequences. Our system can correctly detect 28 events including 13 fall incidents and 15 normal ones from the 30 sequences; whereas two fall incidents were missed. The correctness ratio is about 93%. The miss ratio is 13% and the ratio of false positives is 0%. The reason of unsuccessfully detecting the two fall incidents was that the human objects in the two sequences has small V_{max} values, which was due to some false-segmentation caused in part by show noise. Because a small V_{max} value of an object leads to a small change amount of the object's centroid location and V_{max}, it would become relatively difficult to detect a fall in such sequences correctly.

Table IV
Thresholds for each motion type

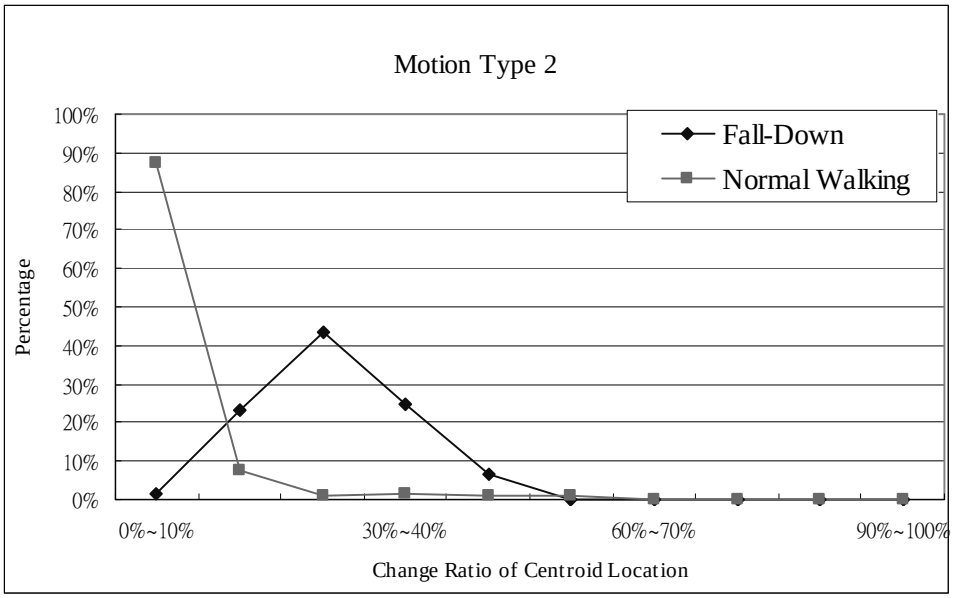
Motion type	Change ratio of centroid location	Change ratio of V_{max}
Type 1	24%	25%
Type 2	25%	30%
Type 3	12%	20%


Motion Type 1
Fall-Down
Normal Walking
Percentage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0%~10%
30%~40%
60%~70%
90%~100%
Change Ratio of Centroid Location

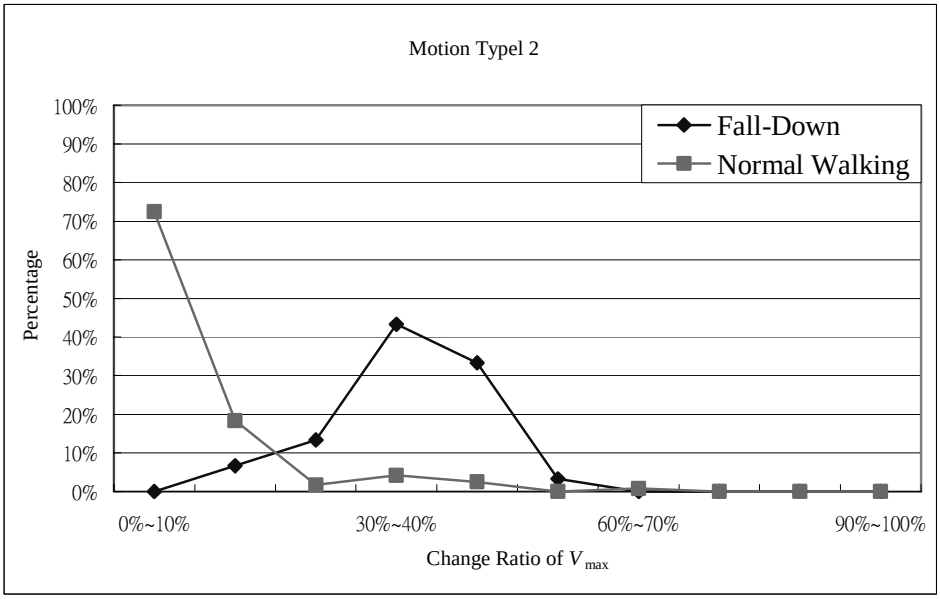
(a)


Motion Type 1
Fall-Down
Normal Walking
Percentage
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0%~10%
30%~40%
60%~70%
90%~100%
Change Ratio of V_{max}

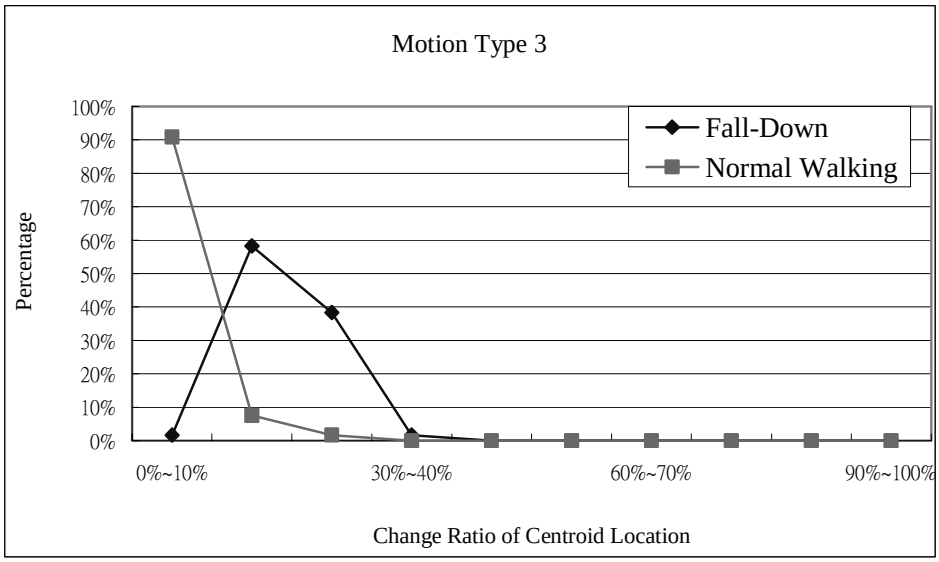
(b)


Motion Type 2
Fall-Down
Normal Walking
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage
0%~10%
30%~40%
60%~70%
90%~100%
Change Ratio of Centroid Location

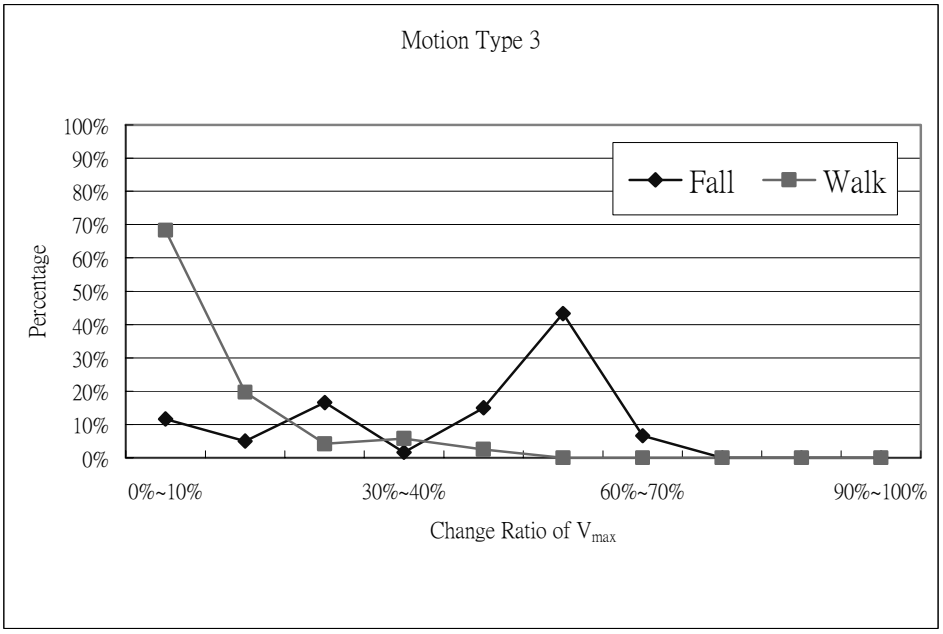
(c)


Motion Typel 2
Fall-Down
Normal Walking
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage
0%~10%
30%~40%
60%~70%
90%~100%
Change Ratio of V_{max}

(d)



(e)



(f)

Fig.11. Histograms of the centroid location and V_{max} change ratios between fall incidents and normal walking for the three motion types: (a)-(b) Motion Type 1; (c)-(d) Motion Type 2; (e)-(f) Motion Type 3.

5. CONCLUSION

We have presented a feature-based compressed-domain fall-down detection scheme for intelligent surveillance applications. The proposed scheme involves two steps: compressed-domain object extraction and fall incident detection. In the object extraction step, the MVs and the DC+2AC image of each frame are firstly extracted. GME is then performed to distinguish moving object MBs from background MBs to obtain a rough object segmentation mask. The CDM is then used to refine the object mask. Should the video shot contain GMs, the GM compensation is performed prior to the change detection operation. Finally, object clustering is performed to separate the object mask into multiple individual objects. In the second step, three feature values: the change ratio of the centroid of a human object, the change ratio of the maximum of vertical projection histogram, and the duration of an event detected are used to identify and locate fall-down events. The proposed object segmentation method can extract moving objects with or without cameral motions, thereby being useful for video surveillance applications equipped with still or pan-tilt-room cameras. Our experimental results show that the proposed method can detect fall incidents with high accuracy in real-time.

REFERENCES

[1] C. S. Regazzoni, G. Fabri, and G. Vernazza, ed., *Advanced Video-based Surveillance Systems,* Kluwer Academic Publishers, 1999.

[2] G. L. Foresti,, P. Mahonen, and C. S. Regazzoni ed., *Multimedia Video-based Surveillance Systems*: *Requirements, Issues and Solutions,* Kluwer Academic Publishers, 2000.

[3] C. S. Regazzoni, V. Ramesh, and G. L. Foresti, "Scanning the issue/technology-Special issue on video communication, processing and understanding for third generation video surveillance systems," *Proc. IEEE,* vol. 89, no. 10, pp. 1355-1367, Oct. 2001.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Systems, Man, and Cybernetics- Part C: Applications and Reviews,* vol. 34, no. 3, pp. 334-352, Aug. 2004.

[5] J. Teno, D. Kiel, and V. Mor, "Multiple strumbles: a risk factor for falls in community-dwelling elderly," *J. America Geriatrics Society,* vol. 38, no. 12, pp. 1321-1325, 1990.

[6] T. Tamura *et al*., "An ambulatory fall monitor for the elderly," in *Proc. IEEE Int. Conf. IEEE Int. Conf. Microtechnologies in Medicine and Biology*, pp. 2608-2610, July 2000., Chicago, IL.

[7] G. Williams *et al*., "A smart fall & activity monitor for telecare applications," in *Proc. Proc. IEEE Int. Conf. IEEE Int. Conf. Microtechnologies in Medicine and Biology*, vol. 20, no. 3, pp. 1151-1154, 1998.

[8] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring behavior in home using a smart fall sensor and position sensors," in *Proc. IEEE Int. Conf. Microtechnologies in Medicine and Biology*, pp. 607-610, Oct. 2000, Lyon, France.

[9] N. Noury, "A smart sensor for the remote follow up of activity and fall detection of the elderly," in *Proc. IEEE Int. Conf. Microtechnologies in Medicine and Biology*, pp. 314-317, May 2002, Lyon, France.

[10] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, Aug. 2000.

[11] C. Kim and J.-N. Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1128-1138, Dec. 2002.

[12] K. Yoon, D.F. Dementhon, and D. Doermann, "Event detection from MPEG video in the compressed domain," in *Proc. IEEE Int. Conf. Pattern Rcognition,* 2000, Barcelona, Spain.

[13] H. Nait-Charif and S. J. McKenna, "Activity summarisation and fall detection in a supportive home environment," in *Proc. IEEE Int. Conf. Pattern Rcognition*, vol. 4, pp. 23-26, Aug. 2004, Cambridge UK.

[14] S. J. McKenna and H. Nait-Charif, "Summarising contextual activity and detecting unusual inactivity in a supportive home environment," *Pattern Analysis and Applications*, vol. 7, no. 4, pp. 386-401, Dec. 2004.

[15] R. Wang, H.-J. Zhang, and Y.-Q. Zhang, "A confidence measure based moving object extraction system built for compressed domain," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 5, pp.21 -24, May 2000, Geneva, Switzerland.

[16] R. V. Babu and K. R. Ramakrishnan, "Compressed domain motion segmentation for video object extraction," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4 , pp. 3788 –3791, 2002.

[17] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: A compressed domain approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 462-474, AApr. 2004.

[18] B. L. Yeo, *Efficient Processing of Compressed Images and Video*, Ph.D. thesis, Princeton University, Jan.1996.

[19] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, pp. 628-631, Mar. 2003, Bangkok, Thailand.

[20] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, pp. 1-11, 1995.

[21] L. G. Shapiro and G. C. Stockman, *Computer Vision*, New Jersey, US: Prentice Hall, 2000.

[22] C.-J. Chang, J.-W. Hsieh, Y.-S. Chen, and W.-F. Hu, "Tracking multiple moving objects using a level-set method," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 2, pp. 101-125, 2004.

[23] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Comm. Technology*, vol. 15, pp. 52-60, 1967.

[24] M. Wollborn and R. Mech, "Refined procedure for objective evaluation of video object generation algorithms," Doc. ISO/IEC JTC1/SC29/WG11 M3448, Mar. 1998.