南　　華　　大　　學

資訊管理學系

碩士論文

使用快速移動歷史影像方法之視訊物件行為識別之研究

Video Object Behavior Recognition using Fast MHI Approach

研 究 生：陳 威 志

指導教授：廖 怡 欽

中華民國　九十九　年　六　月　二十九　日

南 華 大 學

資 訊 管 理 學 系

碩 士 學 位 論 文

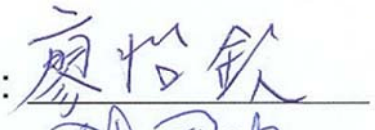
使用快速移動歷史影像方法之視訊物件行為識別之研究

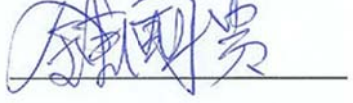
Video Object Behavior Recognition using Fast MHI Approach

研究生： 

經考試合格特此證明

口試委員： 

指導教授： 

系主任(所長)： 

口試日期：中華民國　九十九 年　六　 月 廿九　日

南華大學資訊管理學系碩士論文著作財產權同意書

立書人：陳 威 志 之碩士畢業論文

中文題目：使用快速移動歷史影像方法之視訊物件行為識別之研究

英文題目：Video Object Behavior Recognition using Fast MHI Approach

指導教授： 廖怡欽 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

☑ 共同享有著作權

☐ 共同享有著作權，學生願「拋棄」著作財產權

☐ 學生獨自享有著作財產權

學 生：陳威志

指導老師：廖怡欽

中 華 民 國 99 年 6 月 29 日

南 華 大 學 碩 士 班 研 究 生

論 文 指 導 教 授 推 薦 函

資訊管理學系 碩士班　陳威志　君所提之論文

使用快速移動歷史影像方法之視訊物件行為識別之研究

Video Object Behavior Recognition using Fast MHI Approach

係由本人指導撰述，同意提付審查。

指導教授 廖松欽

99年7月15日

誌　　謝

對研究生來說，能夠寫誌謝是一件多麼令人感動的事情。在研究生生涯的日子裡，壓力、勞累、掙扎、沮喪都是不可避免，能夠將論文完成，大多要歸功於周遭許多人在學業及生活上的大力支持與鼓勵。

要感謝的人很多，但總不能只簡單的用謝天來帶過。首先要感謝的是我的執導教授廖怡欽老師，在研究所期間，對於沒有任何論文寫作相關的學習與經驗的我，不辭辛勞地督導、要求及完善我的研究及論文，給予我嚴謹正確的研究態度，讓我獲益頗豐，最終能順利完成碩士論文。

再來要感謝的是 BI 實驗室的學長：毛利、花君、錦忠、明偉，在我研究過程中，給予我寶貴的建議與激勵；感謝學弟們的支持：柑仔、峰熙、品佑、蕭章、老三、沒品山田，讓煩悶的研究生活中，能有許多調劑。此外，還有一群寶貴的同學：燕將軍、月光猴、瀞仙、魚老闆、淑樺、美秀、米德，除了研究所課程與自己的研究領域之外，我們也一起走過研討會的舉辦、自我評鑑、系所評鑑等大事件，不管什麼事情，都可以相互勉勵、相互陪伴來完成，認識你們是我這兩年最大的收穫。

當然，還有許多關心我、給予我幫助的人。感謝蘇暉凱老師及珍玉姊的期許、感謝楊美蓮老師及尤國任老師的賞識與稱讚、感謝依汝姊的照顧、感謝雅樂團的鼓勵。回首兩年的碩士生涯，很慶幸我從資工背景轉換到資管領域來，雖然一開始頗有些無法適應，但是資管所開拓了我的視野，淺移默化我許多做事情的觀感及態度，讓我能更從容且積極的去面對每件事。

隨著論文的完成，學生的生涯也即將畫上句點，在口考完成後的階段，一直在思考「莫違本心、莫忘初衷」這句話，讓我有機會好好反思這兩年裡的所得所失。雖然研究生的生涯總是艱辛、挫折居多，但痛會過去美會留下，感謝每一個人，讓我能帶著不虛此行的感覺離開學校。

威志 于 204 研究室
2010 年 七月

使用快速移動歷史影像方法之視訊物件行為識別之研究

學生：陳威志　　指導教授：廖怡欽

南　華　大　學　資訊管理學系碩士班

摘　　要

隨著科技的進步，監視系統的安裝也愈來愈廣泛，要檢查擷取下來的畫面中是否發生特定事件，通常需要很長的時間及大量的人力介入。為了解決這個問題，目前已有許多可自動判斷視訊物件行為的方法被提出來，比起其他方法，以移動歷史影像為基礎的方法通常具有計算複雜度低與容易實作等優點，因此也比較受大家的歡迎。本篇論文提出一個快速移動歷史影像方法，針對每一已知行為事先建立多組特徵值、運用部分距離計算法、以及變換區塊計算順序，來降低移動歷史影像方法的運算時間。為了驗證所提方法的有效性，本篇論文採用 Chen 方法中九個區域的平均像素移動方向作為特徵值，並以歐基里德距離作為相似度比對。依據實驗結果顯示，所提方法確實能夠有效降低運算時間。

關鍵字：移動歷史影像法、視訊物件行為識別

Video Object Behavior Recognition using Fast MHI Approach

Student：Wei-Chih Chen Advisor：Dr. Yi-Ching Liaw

Department of Information Management
The Graduated Program
Nan-Hua University

ABSTRACT

As the progress of technology, the surveillance system is installed more and more widely. It usually requires a lot of time and human efforts to check if a specified event occurs in the captured video. To solve this problem, there are many approaches were proposed to recognize behaviors of a video object automatically. Among available behavior recognition methods, the MHI-based approaches are more popular for they have less computational complexity and are easier than another. In this thesis, a fast MHI approach is proposed to reduce the computation time of the MHI approach by storing the multiple sets of features for a predefined behavior, using the partial distance computation method, and changing the calculated order. Nine local information proposed by Chen and squared Euclidean distance are used in the behavior matching process in this thesis to manifest the performance of the proposed approach. Experiment results show that the proposed method can effectively reduce the computation time of MHI approach.

Keyword*: Motion History Image, Video Object Behavior Recognition*

List of Contents

List of Tables

List of Figures

Chapter 1 Introduction

The surveillance system is a system to obtain a series of images (frames) from real world by using a video camera and the captured images are transferred to a remote monitor for displaying. Through the displayed images, one can realize what is happened in front of the camera. That is, a person can sit in front of a monitor to see the images transferred from the remote cameras to check if there is something happened. Moreover, the captured images can be saved in a storage device, such as a hard disk, to make the captured images can be replayed in any time and at any places. The stored images can also be used as evidences for used in the criminal cases. Nowadays, the surveillance system is a very useful and popular technology and is widely deployed everywhere.

The surveillance system can be set up for various purposes [1], such as the traffic monitoring [2-3], living environment monitoring [4], and remote monitoring [5]. In the application of traffic monitoring, the surveillance system is installed to take frames of roads. The captured images are then transferred to the traffic monitoring center for showing and storing. It is convenient for people to see whether an accident or a traffic jam occurred or not. In the case that an accident occurred, images recorded in the surveillance system can be used to clarify the reason of the accident. When the surveillance system is used to help increasing the security of living environment, cameras are deployed to take images of the elevator, the entrance of building, or the corners in a community. In such a case, the administrator of the surveillance system can sit in the security operation center and easily monitor the surrounding environment of

the community. For the remote monitoring application, the surveillance system is applied to capture images at where humans cannot be there all the time or unreachable. For instance, NASA launched a space ship to take pictures of Mars [6] and the Spirit rover [7] to analyze the distribution and composition of minerals, rocks, and soils.

In above applications of the surveillance system, it requires a lot of time and human efforts to check the content of video and is very easy to make mistakes if the person who is watching the monitor was tired or not attentive. Besides, the task of searching a specific event from a lengthy video stream is often required and difficult. To solve these problems, many approaches were proposed [8-12] for analyzing the video content automatically. Such approaches can be categorized into two categories that are the still image analyzing approaches [8-9] and the motion recognition approaches [10-12].

One famous example of the still image analyzing approaches is the car license plate recognition system [9]. The application of car license plate recognition is to automatically extract characters on the car license plate from a car image. To extract characters from a car image, a series of image processing techniques must be applied. These techniques include: color space conversion, license plate localization, character segmentation, and pattern recognition. The extracted characters can then be used as a key to search the car license database for checking if the car in the image is legal or illegal. If the license is illegal, this information will be passed to the parking office or the police office for taking proper actions. Comparing to the motion recognition approaches, the still image analyzing technology is maturer and easier, as well as has less limitation on available computation time.

The motion recognition technology [10-12] is applied to recognize the behavior of a video object from a series of motions which are recorded in a set of successive video frames. A classical motion recognition approach consists of a training phase and a testing phase. In the training phase, the surveillance system is installed to capture the motion frames of pre-defined behaviors for a video object. For a pre-defined behavior, a set of features is obtained from the motion frames of the behavior. That is, in the end of the training phase, a batch of feature sets is stored in a behavior database for a set of predefined behavior. In the testing phase, a set of features is first extracted from the input frames, which are usually captured in real time and from the surveillance system, of an unknown behavior. The same process of feature extraction to the training phase is then progressed to generate the feature set for the motion frames of an unknown behavior. Thus, we can deploy some algorithms to match the generated feature set of the unknown behavior and feature sets in the behavior database to recognize the unknown behavior.

It is not an easy task to extract a feature set from the motion frames. To conquer this problem, many methods have been proposed [13-18], such as hidden Markov models (HMM) [16], kinematic modes [17], and motion history image (MHI) approach [18]. Among available approaches, the MHI approach usually takes less computational time and requires less prior knowledge. The MHI approach is a template-based [19-20] video object behavior recognition method. Two types of templates (motion energy image and motion history image) are used as patterns of a behavior in the MHI approach. Given a set of successive motion frames, to produce the motion energy image (MEI) for the motion frames, the object masks, which record the position information

for a video object, of motion frames are generated and overlapped in time sequence. In the MEI of the motion frames, the object mask of a frame with earlier time is placed in a lower layer. Otherwise, it will be placed in an upper layer. The motion history image (MHI) is similar to the MEI, and the only difference is that the intensity of an object pixel in an MHI is related to its appearing time. A pixel in an MHI with earlier appearing time will be assigned to a smaller intensity. Otherwise, it will have a greater intensity.

Figure 1-1 (a) gives a set of successive frames of a video object and Figure 1-1 (b) shows the object masks of frames in Figure 1-1 (a). In Figures 1-1 (a) and 1-1 (b), frames are arranged in time sequence from left to right. Figures 1-1 (c) and 1-1 (d) show the MEI and MHI of frames in Figure 1-1 (a), respectively.

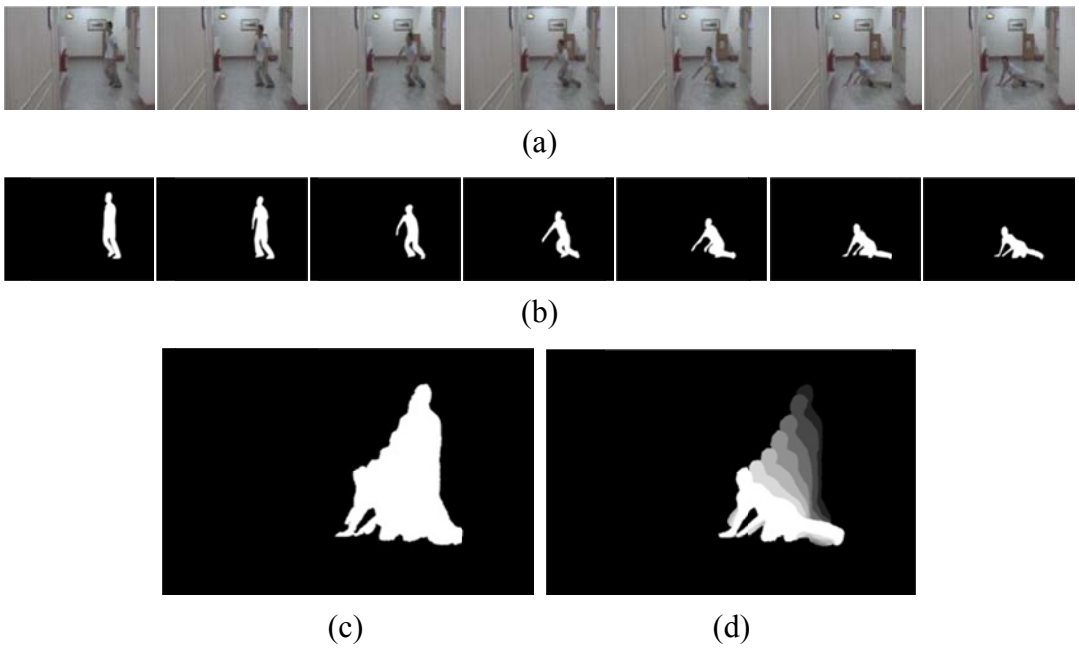


Figure 1-1: An example of MHI approach:
(a) a set of successive frames of a video object, (b) the object masks of the video object, (c) the MEI of the video object, and (d) the MHI of the video object.

From Figure 1-1, we can see that the MHI stores both the orientation information and the temporal history of a video object's movement, while the MEI records the moving region of the video object's motion. These two types of templates are used in MHI approach as patterns for recognizing a video object's behavior.

Although the MHI approach is more appropriate for recognizing a video object's behavior than the others according to the computational complexity, there still are some problems in the MHI approach. The major problem of the MHI approach is that the accuracy of the MHI approach is very sensitive to the time interval between two consecutive frames, the position of the video object in the frame, and the contour of the video object. To overcome these drawbacks of the MHI approach, lots of improving methods were proposed [21-30].

Chen et al. [29-30] proposed an improved MHI approach to increase the accuracy and availability of the MHI method by using a special designed feature extraction scheme. Since the information in MEI can also be obtained from the MHI, Chen's method uses only the MHI to generate two sets of features from the MHI of input frames. These two sets of features are the motion gradient magnitude histogram (MGMH) and the local orientation of MHI. The MGMH is generated by computing the gradient of every pixel in the MHI, and the local information of MHI is obtained by dividing the MHI into nine blocks and calculating the motion orientation for each block. By utilizing the MGMH and the local information of MHI, the accuracy of behavior recognition can be effectively improved.

Among available MHI based methods, Chen's method is very appropriate for recognizing the behavior of a video object because of higher accuracy. However, the

time complexity of Chen's method is still too high. To reduce the computational complexity of Chen's method, a fast MHI approach is proposed in this thesis. The proposed approach adopts a similar feature extraction process as that of Chen's method to remain a similar accuracy as Chen's method. To reduce the time complexity, three processes are presented in this thesis. The first process is to store multiple sets of features for a predefined behavior to decrease the MHI generation time. The second process is to reduce the distance calculation time using a partial distance calculation scheme. The final process is to improve the performance of the partial distance calculation scheme by changing the calculated order of features. Through the above three processes, the time complexity of the MHI approach can be effectively reduced.

The rest of this thesis is organized as follows. In chapter 2, the MHI generation process and Chen's method will be introduced. Our proposed method is presented and described in chapter 3. Experimental result and conclusions are given in chapter 4 and chapter 5, respectively.

Chapter 2 Motion History Image Approach

Motion History Image (MHI) approach was presented by James Davis in 1997 [18] which uses temporal templates to represent and recognize human actions. Figure 2-1 shows the procedures of the MHI approach and the feature extraction for a video object from input frames using the MHI approach. A more detail explanation for the feature extraction process and behavior matching process are provided in the following sections.

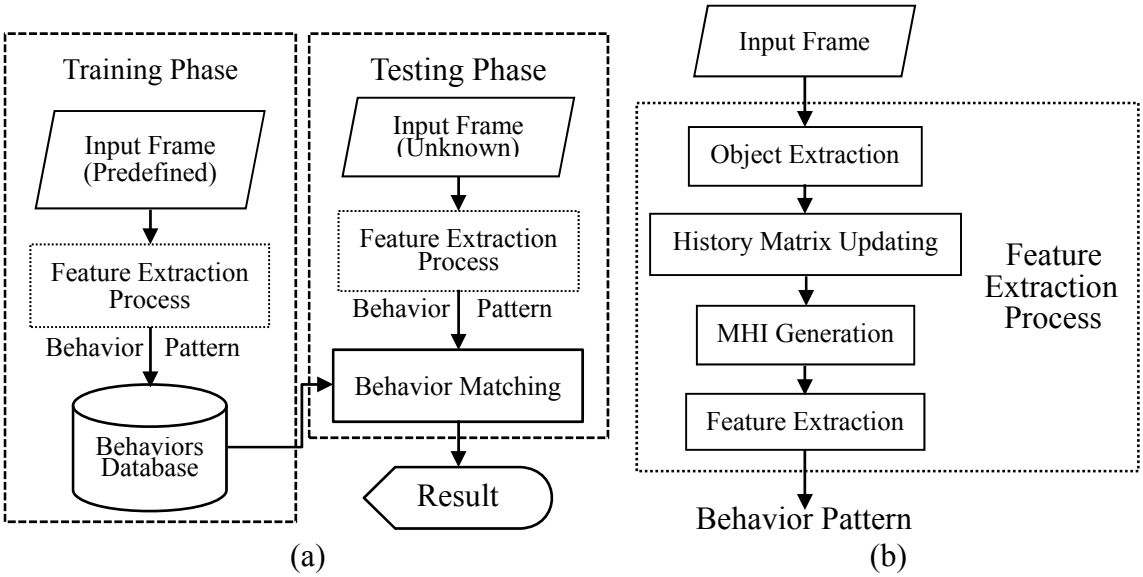


Figure 2-1: (a) The procedure of the MHI approach and (b) the feature extraction process.

2.1 Object extraction process

The object extraction process is used to find where the moving object is located in the input frame and to generate an object mask for the input frame. There are many methods [31-32] developed to deal with this problem. The most common used method is the background suppression segmentation (BSS) method [32].

Using the BSS method, a background frame is initially saved as a reference frame. To generate the object mask for an input frame, the first step of BSS method is to evaluate the difference frame between the reference frame and the input frame. Let $r(x, y)$ and $c(x, y)$ be two pixels in the reference frame and the input frame with coordinate (x, y), respectively. The value in difference frame with coordinate (x, y) is the difference between pixels $r(x, y)$ and $c(x, y)$ and can be computed as below.

$$d(x,y) = |r(x,y) - c(x,y)| \quad (2\text{-}1)$$

Once the difference frame is evaluated, a threshold value is applied to check if a pixel in the input frame is an object pixel or a background pixel. In case of a pixel in the difference frame with a value greater than the threshold value, its corresponding pixel in the input frame is an object pixel. Otherwise it is a background pixel. For a pixel in the input frame is an object pixel, its corresponding pixel in the object mask of the input frame is set as 255. Otherwise, it is set as 0.

After generating the object mask, some post processing methods [33] can be applied to remove noise and small regions from the generated object mask. Figure 2-2 shows an example for the input frame, reference frame, and object mask of the input frame.

(a) (b) (c)

Figure 2-2: An example of BBS method:
(a) the reference frame, (b) the input frame, and (c) the object mask.

2.2 History matrix updating process

As mentioned in the above step, a pixel in an object mask denotes the appearance status of video objects for a particular position. The purpose of the history matrix updating process is to record the appearance history for video objects. To accomplish this goal, a history matrix with the same size as input frames is maintained. Each element in the history matrix is initialized to 0 and updated according to the values of the incoming object masks using a timestamp with an initial value of 1.

Let the current value of timestamp be τ and the value for an element in the object mask with coordinate (x, y) at time-point τ be $M^{\tau}(x, y)$. The updating method for an element in the history matrix with coordinate (x, y) at time-point τ is defined in the following:

$$H^{\tau}(x,y) = \begin{cases} \tau & \text{if } M^{\tau}(x,y) = 255 \\ 0 & \text{if } M^{\tau}(x,y) = 0 \text{ and } H^{\tau-1}(x,y) \leq (\tau - \delta) \end{cases} \quad (2\text{-}2)$$

where δ is the period of time to observe a motion.

Figure 2-3 gives an example to show how the history matrix is updated under a series of the incoming object masks when δ=3.

From Figure 2-3, we can see that, in the beginning, every element in the history matrix is initialized to 0 and is updated using the content of the incoming object masks. As shown in Figure 2-3 (b), we have an incoming object mask which includes an object appears in the left side of the object mask. Since the object mask in Figure 2-3 (b) is the first object mask, we set the value of timestamp to 1. That is, τ is 1 for the first object mask. The updating process of the history matrix using the first object mask is to fill up the value of τ into elements in the history matrix, which have their corresponding pixel

in the object mask with the value of 255, and clear elements in history matrix which have values equal to or smaller than ($\tau - \delta$) to 0. The updated history matrix for the first object mask is given in Figure 2-3 (c). For the successive incoming object masks, the value of timestamp is increased by 1 and the similar process is applied to update the history matrix.

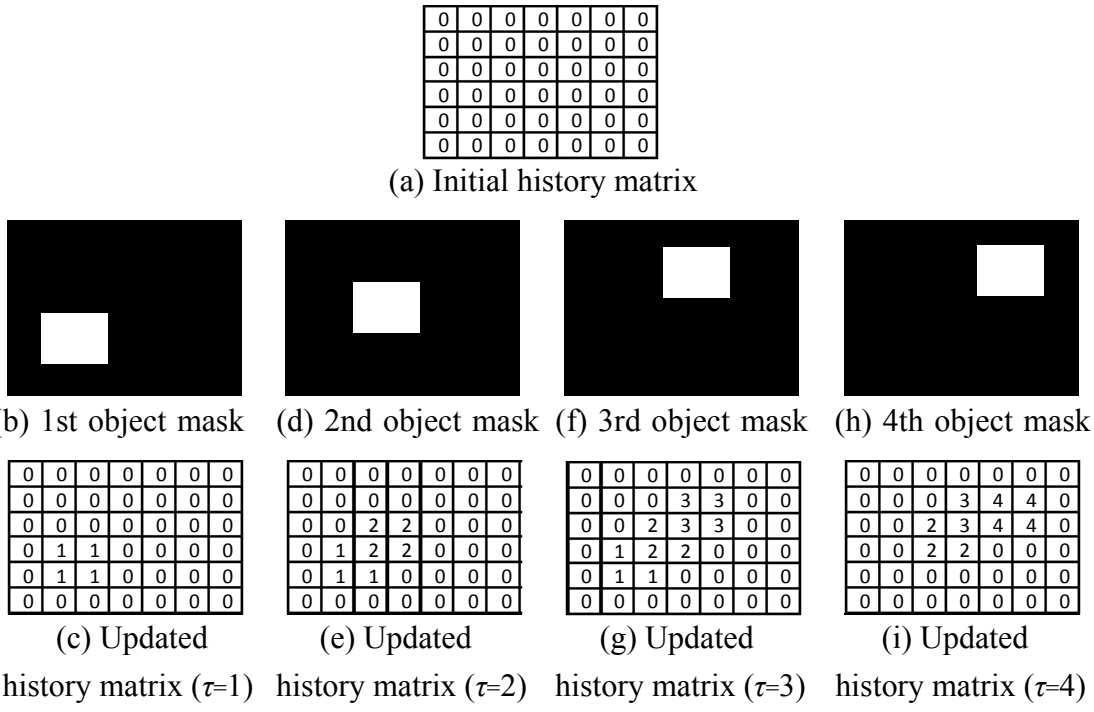


Figure 2-3: An example to show the updating process of the history matrix.

2.3 MHI generation process

Once the history matrix is updated and the value of τ is equal to or greater than δ, that means we have enough frames available for motion recognition and an MHI can be generated using the following equation:

$$MHI^{\tau}(x,y) = \begin{cases} \frac{H^{\tau}(x,y) - (\tau - \delta)}{\delta} \times 255 & \text{if } H^{\tau}(x,y) \neq 0 \\ 0 & \text{if } H^{\tau}(x,y) = 0 \end{cases} \quad (2\text{-}3)$$

where $MHI^{\tau}(x, y)$ is the value of a pixel in the MHI at time-point τ with coordinate (x, y).

Figure 2-4 gives an example to show how an MHI is generated for δ=8. From Figure 2-4, we can find that δ object masks are merged together to form an MHI. In an MHI, object masks with smaller τ are placed in lower layers and have less brightness. Otherwise, object masks are placed in upper layers and assigned with higher intensities. Obviously, we can see that an MHI records motions of an object during a period of time and can be used as a template to recognize an object's behavior.

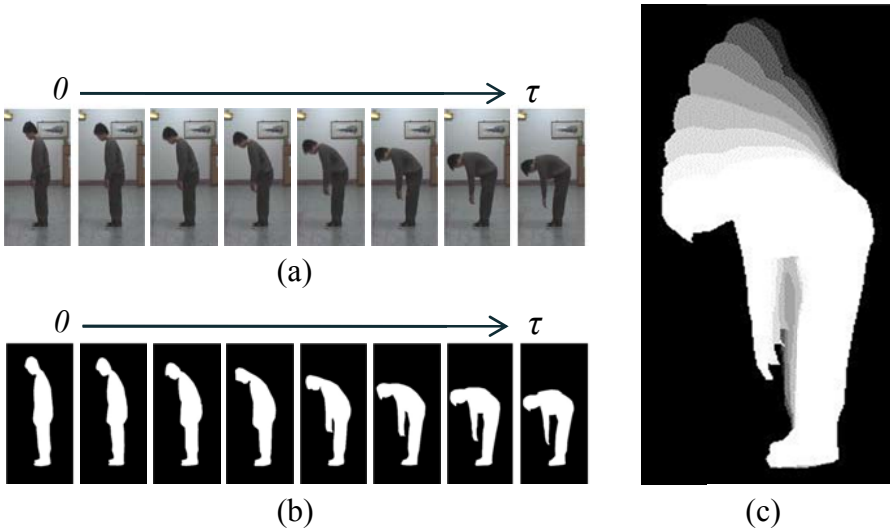
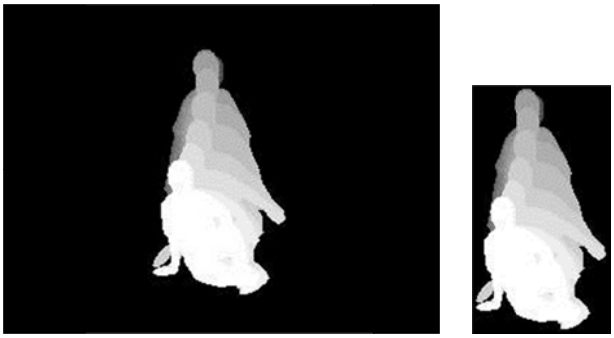


Figure 2-4: An example of MHI approach:
(a) the input frames, (b) the object masks, and (c) the MHI.

Since we focus only on the object, the surrounding black area should be removed for reducing memory space and speeding up the processing time. Figure 2-5 shows an example for an MHI image before and after removing the surrounding black area.



(a) (b)

Figure 2-5: An example of removing the surrounding black area:
(a) before and (b) after removing the surrounding black area from an MHI.

2.4 Feature extraction process

After the MHI is obtained, angle information of each pixel in the MHI is evaluated and used as features for behavior recognition in Chen's method [29].

To evaluate the angle information for an MHI, we first create four blank 2-D arrays: G_x^τ, G_y^τ, θ^τ and E^τ. Arrays G_x^τ, G_y^τ, θ^τ, and E^τ are used to record the horizontal gradients, vertical gradients, angles, and energies for pixels in the MHI of time-point τ, respectively. Here, two sobel masks S_x and S_y as given below are applied to evaluate the horizontal and vertical gradients, respectively.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2-4}$$

Once G_x^τ and G_y^τ are evaluated, the angle and energy of a pixel with coordinate (x, y) in the MHI can be computed using the following equations:

$$\theta^\tau(x,y) = tan^{-1}\left(\frac{G_y^\tau(x,y)}{G_x^\tau(x,y)}\right) \tag{2-5}$$

$$E^\tau(x,y) = \sqrt{G_x^\tau(x,y)^2 + G_y^\tau(x,y)^2} \tag{2-6}$$

After the angles and energies for every pixel in the MHI are obtained, the histogram of angles for the MHI could be computed and used as a feature in the behavior recognition process.

Here, the whole angle range of ω is evenly divided into 72 parts and the energy of each pixel is used as the weighting value for computing the histogram of angles. The computation of histogram with 72 angle parts is given below, where ω is the degree of angle for each pixel and $P^{\tau}_{normal}(\sigma)$ denotes the normalized strength of σth angle part.

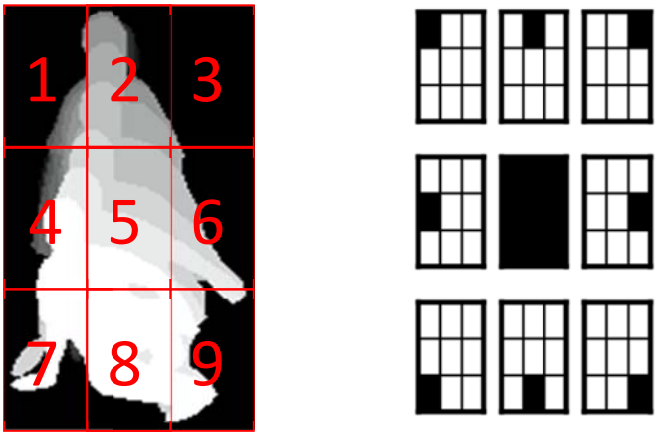
$$\hat{\mathrm{E}}^{\tau}(\omega,x,y)=\begin{cases}E^{\tau}(x,y) & \text{if } \theta^{\tau}(x,y)=\omega \\ & \text{and } MHI^{\tau}(x,y)\neq 0,\ \ 0\leq\omega<360 \\ 0 & \text{otherwise}\end{cases} \tag{2-7}$$

$$P^{\tau}(i)=\sum_{j=i\times5}^{i\times5+4}\sum_{x,y}\hat{\mathrm{E}}^{\tau}(j,x,y)\ ,0\leq i<72 \tag{2-8}$$

$$P^{\tau}_{avg}(i)=\sum_{\mathrm{n}=0}^{71}P^{\tau}(n)/72 \tag{2-9}$$

$$P^{\tau}_{normal}(\sigma)=\frac{P^{\tau}(\sigma)-P^{\tau}_{avg}}{\sqrt{\sum_{m=0}^{71}\left(P^{\tau}(m)-P^{\tau}_{avg}\right)^{2}/_{72}}}\ ,0\leq\sigma<72 \tag{2-10}$$

After the histogram is evaluated, we have the strengths of 72 angle parts. Besides of using the histogram as a feature, the local information of an MHI is also important and can be used as features. To obtain the local information of an MHI, MHI is divided into 9 blocks as depicted in Figure 2-6 (a). Each block covers a specified range of area as shown in Figure 2-6 (b) and the major orientations of blocks are used as features for behavior recognition.



(a) (b)

Figure 2-6: The method of dividing an MHI into blocks and the covering range of each block.

From Figure 2-6 (a), we can see that an MHI is divided into nine blocks and each block is assigned a specified number. Figure 2-6 (b) shows the covering ranges for all blocks. It is noted that the fifth block covers the whole range of MHI and should be computed last to reduce the computation time. Let O_n^τ be the local orientation of *n*th block for an MHI. The definition of O_n^τ is listed as follows:

$$norm(\tau, \delta, H^\tau(x,y)) = 1 - \frac{\tau - H^\tau(x,y)}{\delta} \quad (2\text{-}11)$$

$$O_n^\tau = \theta_{n,ref}^\tau + \frac{\sum_{x,y} angDiff\left(\theta_n^\tau(x,y), \theta_{n,ref}^\tau\right) \times norm(\tau,\delta,H^\tau(x,y))}{\sum_{x,y} norm(\tau,\delta,H^\tau(x,y))}, \quad 1 \le n \le 9 \quad (2\text{-}12)$$

where $norm(\tau, \delta, H^\tau(x,y))$ is the normalization of the history matrix with coordinate (x, y), $\theta_n^\tau(x,y)$ is the angle of a pixel in *n*th block with the coordinate (x, y), $\theta_{n,ref}^\tau$ is the maximum angle in *n*th block, and *angDiff* is a function to evaluate the angle difference between $\theta_{n,ref}^\tau$ and $\theta_n^\tau(x,y)$.

In Chen's method, 72 angle strengths $P_{normal}^\tau(\sigma), 0 \le \sigma < 72$ and 9 block orientations $O_n^\tau, 1 \le n \le 9$ of an MHI are used as a feature set in the behavior matching process.

2.5 Behavior matching process

There are many methods for matching behavior, such as artificial neural network and Euclidean distance. In this thesis, the squared Euclidean distance is used to find the most similar predefined behavior from the behavior database for input frames. The extracted feature set is compared to all of the feature sets in the behavior database to find the most similar one. If the difference between the extracted feature set and the most similar feature set from the behavior database is less than a given threshold value, we can say the behavior corresponding to the most similar feature set appears in the input frames. Otherwise, the behavior is not detected and the object's motion in the input frames is undefined.

Chapter 3 Fast MHI Approach

The MHI approach is an easy method to recognize behaviors for video objects, but it takes too much computing time to extract features as Chapter 2 described. For a behavior database contains K predefined behaviors with various lengths of motion time, to recognize the behavior of input frames, we must generate several MHIs of various deltas. Let the behavior database consists of a delta set $\Delta = \{\delta_1, \delta_2, \delta_3, \ldots, \delta_L\}$. To recognize whether the monitored screen presents a predefined behavior or not, we need to generate L MHIs for different deltas. That is, for every input frame, we have to execute the Object Extraction process and History Matrix Updating processes once, and the MHI generation and Features Extraction processes L times.

To reduce the computational complexity of the MHI approach and Chen's method, a fast MHI approach is proposed in this thesis. The proposed approach uses nine block orientations, which is the same as that used in Chen's method, to simplify the feature extraction process and remain a good recognition performance. To reduce the time complexity in advance, three processes are proposed and presented in the following sections.

3.1 Storing multiple sets of features for a predefined behavior

Since the sustained time of different types of behaviors is diverse, numbers of frames in MHIs for different behaviors are also different. That means different behaviors may have different deltas in constructing their MHIs, and each predefined behavior has its own delta and a set of features must be stored in the behavior database

for the delta. Let the delta of a predefined behavior be δ_b. The set of features stored in the behavior database for the behavior is generated using an MHI containing δ_b frames. To recognize whether the input frame appears a predefined behavior, we must generate multiple MHIs of all possible deltas in Δ, extract multiple sets of features for all MHIs, and compare the extracted features to those of the predefined behaviors. This process is very time consuming. To solve this problem, we choose to store multiple sets of features for a predefined behavior. For a predefined behavior with the smallest delta, say δ_1, only one set of features is stored and for a behavior with larger delta, say δ_i, $2 \le i \le L$, i sets of features must be stored in the behavior database. To meet our requirement, we redesign the behavior database to store multiple sets of features for predefined behaviors. Table 3-1 shows the design of our behavior database.

Table 3-1: The design of Behavior database

Delta / Behavior	δ_1		δ_2		δ_3		…	δ_L	
	Features	LF	Features	LF	Features	LF		Features	LF
B_1	$F_{\delta_1}^{B_1}$	T							
B_2	$F_{\delta_1}^{B_2}$	F	$F_{\delta_2}^{B_2}$	T					
B_3	$F_{\delta_1}^{B_3}$	F	$F_{\delta_2}^{B_3}$	T					
B_4	$F_{\delta_1}^{B_4}$	F	$F_{\delta_2}^{B_4}$	F	$F_{\delta_3}^{B_4}$	T			
⋮									
B_K	$F_{\delta_1}^{B_K}$	F	$F_{\delta_2}^{B_K}$	F	$F_{\delta_3}^{B_K}$	F	…	$F_{\delta_L}^{B_K}$	T

From Table 3-1, we can see that we have multiple behaviors defined in the behavior database and all behaviors are sorted according to their deltas. For a behavior

with smaller delta, fewer sets of features are stored. Otherwise more sets of features must be recorded. Where LF means the last flag and is used to recognize if this is the original delta of a behavior. In other words, the feature set with the value 'T' of LF is the really correct delta in recognizing a behavior. Otherwise the feature set of other deltas are just used to reduce unrequired calculation and LF is set as 'F'. Here, $F_{\delta_i}^{B_j}$ for $1 \le i \le L$ and $1 \le j \le K$ is the feature set for behavior B_j and delta=δ_i. In our approach, only the local information of MHI is used. That is, we use nine block orientations O_n, $1 \le n \le 9$ for an MHI as features.

Let $F_{\delta_i}^{\tau}$ be the feature set extracted from the input frames at time point with delta=δ_i. The procedure of behavior recognition using multiple sets of features for the input frames is described in Figure 3-1.

```
Set   dist[j] = 0,   1 ≤ j ≤ K
Set   done[j] = false,   1 ≤ j ≤ K
for   i = 1 to L {
      obtain F_δi^τ
      for   j = 1 to K {
            if ( done[j] = false )
                  dist[j] = D(F_δi^τ, F_δi^Bj)
                  if ( (dist[j] > THR) or (LF_δi^Bj = true) )
                        done[j] = true
      }
}
Set   id = 1
for i = 2 to K
      if ( dist[i] < dist[id] )
            id = i
if ( dist[id] > THR )
      id = 0
```

Figure 3-1: The algorithm of behavior recognition using multiple sets of features for a predefined behavior.

Where $D(F_{\delta_i}^{\tau}, F_{\delta_i}^{B_j})$ is the squared Euclidean distance function, and *THR* is a threshold value which should be determined experimentally. As shown in Figure 3-1, the procedure will find the *id* of behavior which has the less distance to features extracted from input frames. If no behavior is recognized, the value of *id* will be 0.

3.2 Partial distance calculation

As mentioned above, the squared Euclidean distance is used to calculate the distance between two feature sets. For two sets of features F^1 and F^2 which are generated by the same delta from behavior 1 and behavior 2 respectively, the definition of squared Euclidean distance $D(F^1, F^2)$ is given below. Where the *Blk* is denoted the block number in an MHI.

$$D(F^1, F^2) = \sum_{Blk=1}^{9}(O_{Blk}^1, O_{Blk}^2)^2 \quad (3\text{-}1)$$

Using the Euclidean distance to evaluate the distance between two features, all features of the input MHI must be determined. In many cases, we don't have to fully compute the squared Euclidean distance to know if a predefined behavior is not the behavior of the input frames. For example, if the behavior of input motion frames is not predefined in the behavior database, the generated feature set of input motion frames must be different from any one feature set in the behavior database. In other words, the feature in a feature set which is generated from the input motion frames may not be the same to any one feature in the behavior database. According to this observation, a partial distance calculation method is proposed. The partial distance calculation method is to divide squared Euclidean distance calculation process into nine separated steps in terms of blocks calculation order. Let $D^i(F^1, F^2)$ is the ith partial distance for feature

sets F^1 and F^2 . The definition of $D^i(F^1, F^2)$ is given below.

$$D^i(F^1,\ F^2)\ =\ (O_i^1,\ O_i^2)^2 \tag{3-2}$$

From equation (3-2), we can see that the $D(F^1, F^2)$ can be computed using the following equation.

$$D(F^1,\ F^2)\ =\ \sum_{i=1}^{9} D^i(O_i^1,\ O_i^2)^2 \tag{3-3}$$

By dividing the squared Euclidean distance into nine steps, we can check the accumulated distance after each partial distance is computed to see whether current accumulated distance is already excess *THR* or not. If current accumulated distance is excess *THR*, the distance computation process can be terminated earlier.

```
Set   dist[j] = 0,   1 ≤ j ≤ K
Set   done[j] = false,   1 ≤ j ≤ K
for   i = 1 to L {
    for   Blk = 1 to 9 {
        obtain  O^τ_Blk  from  MHI^τ_δi
        Set terminate = true
        for   j = 1 to K
            if ( (done[j] = false) and (terminate=false) ){
                dist[j] += D^Blk(O^Bj_Blk,  O^τ_Blk)
                if ( (dist[j] > THR) or (LF^Bj_δi = true) ){
                    done[j] = true
                } else terminate = false
            }
        if (terminate = true ){
            id = 0
            return
        }
    }
}
Set   id = 1
for i = 2 to K
    if ( dist[i] < dist[id] )
        id = i
if ( dist[id] > THR )
    id = 0
```

Figure 3-2: The algorithm of behavior recognition using partial distance calculation and multiple sets of features.

To further reduce the computational complexity, the feature set for the input frames should not be evaluated at a time. That is, a feature of the input frames must be evaluated when we need it. In such a case, if the feature set of the input frames is quite different from those stored in the behavior database, a lot of computations can be avoided. Figure 3-2 gives the procedure for applying the partial distance calculation method to behavior recognition.

3.3 Changing distance calculated order

In the process of distance calculation, we need to compute square sum for nine blocks. The calculating order of nine blocks is {1, 2, 3, 4, 6, 7, 8, 9, 5}. The purpose of partial distance calculation is to early terminate the computation of a distance. Empirically, we found that the distance contributed by every block are quite difference. That is, the calculating order of nine blocks can be changed according to the degree of distance contribution from all blocks.

Equation 3-3 gives the definition of distance contribution calculation, where AD^{Blk} is the average distance of block, *Un* is the number of unknown behavior MHIs, *K* is the number of the predefined behaviors, O_{Blk}^{B} is the block feature of someone behavior in the behavior database, and O_{Blk}^{Un} is the block feature of unknown behavior MHIs.

$$\mathrm{A}D^{Blk}(F^1,\ F^2)\ =\ \sum_{Un}\sum_{B=1}^{K}(O_{Blk}^{B},\ O_{Blk}^{Un})^2\ /\ K\ \ ,\ 1 \le Blk \le 9 \qquad (3\text{-}3)$$

Figure 3-3 gives an experimental result to show the distance contribution for every block, where the distance contribution of a block is evaluated by summing the distances between features of the block from predefined behaviors and our testing MHIs (Figure 4-43).

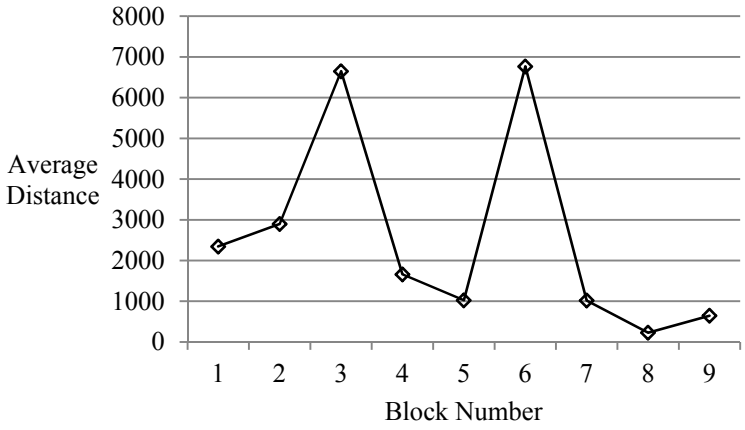


Figure 3-3: Average distances for nine blocks.

From Figure 3-3, we may find that blocks 3 and 6 always contribute more distance and blocks 7, 8, 9 contribute less distances. Thus, the distance calculated order is changed to {6, 3, 2, 1, 4, 7, 9, 8, 5} for improving the efficiency of partial distance calculation.

Chapter 4 Experimental Results

To evaluate the performance of the proposed fast MHI method, in the training phase, four behaviors (falling, hunkering, sitting, and standing) were acted by two persons and captured by a camera and from four directions (front, back, left, and right). That is, eight video clips were captured for each behavior. The size of captured frames is 704x480 pixels.

Figures 4-1 to 4-16 give the object masks of four behaviors performed by the first person and Figures 4-17 to 4-32 show the object masks of four behaviors performed by the second person. The corresponding MHIs for Figure 4-1 to Figure 4-32 are given in Figure 4-33 to Figure 4-42, respectively. All the following shown figures are sorted from the left to the right and from the top to the bottom in terms of time passing. The following figures of delta set for falling, hunkering, and sitting behaviors is 12, but the delta of standing behavior contains both 12 and 16.

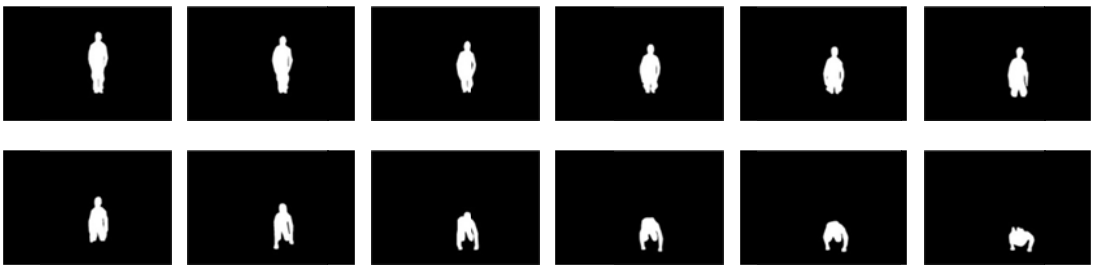


Figure 4-1: The object masks of falling behavior performed by person 1 and captured from the front side.

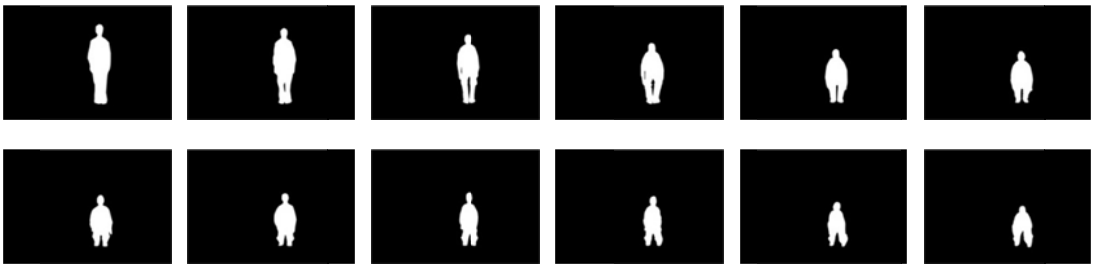


Figure 4-2: The object masks of falling behavior performed by person 1 and captured from the back side.

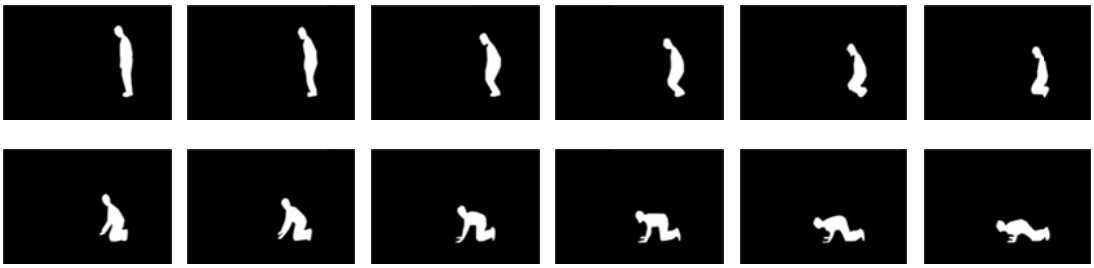


Figure 4-3: The object masks of falling behavior performed by person 1 and captured from the left side.

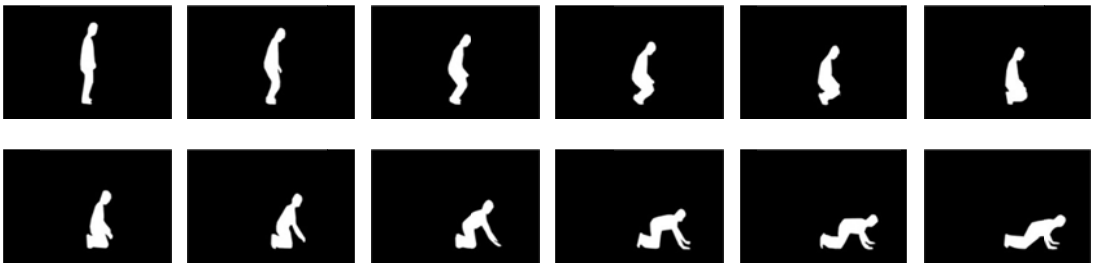


Figure 4-4: The object masks of falling behavior performed by person 1 and captured from the right side.

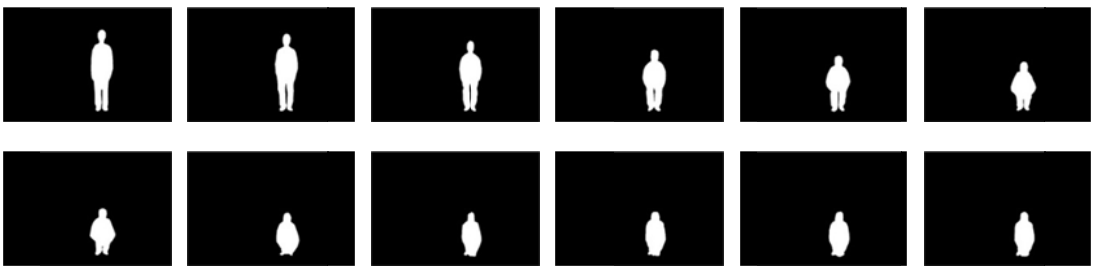


Figure 4-5: The object masks of hunkering behavior performed by person 1 and captured from the front side.

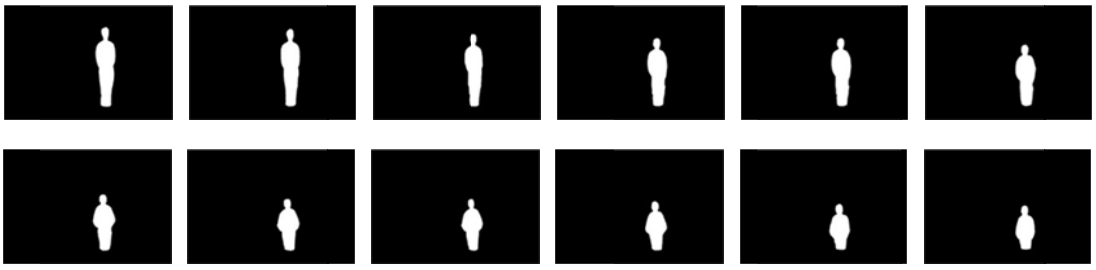


Figure 4-6: The object masks of hunkering behavior performed by person 1 and captured from the back side.

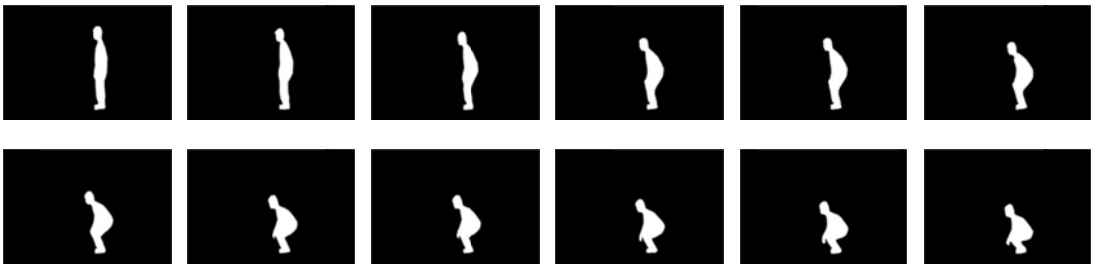


Figure 4-7: The object masks of hunkering behavior performed by person 1 and captured from the left side.

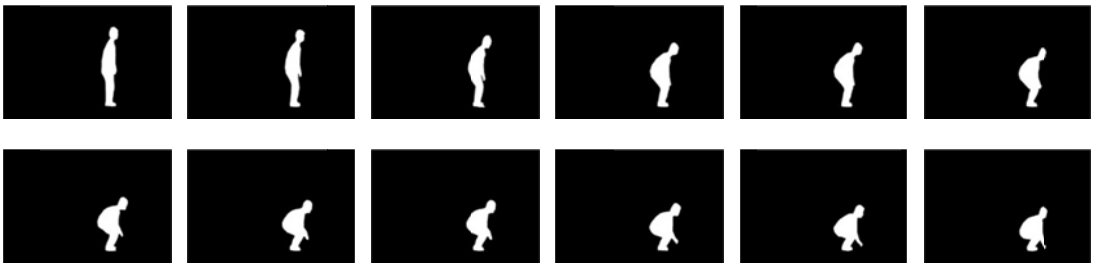


Figure 4-8: The object masks of hunkering behavior performed by person 1 and captured from the right side.

Figure 4-9: The object masks of sitting behavior performed by person 1 and captured from the front side.

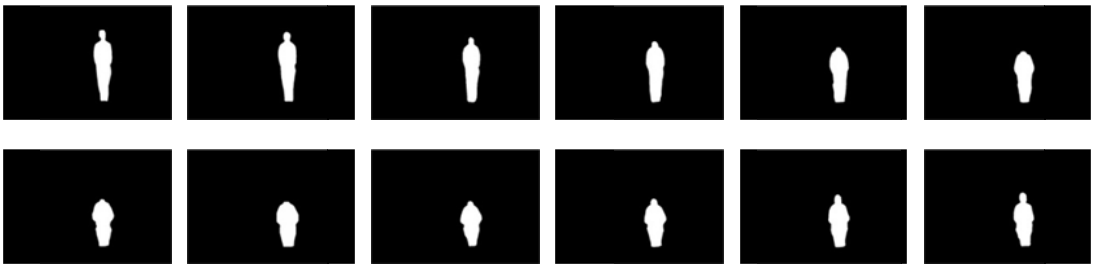


Figure 4-10: The object masks of sitting behavior performed by person 1 and captured from the back side.

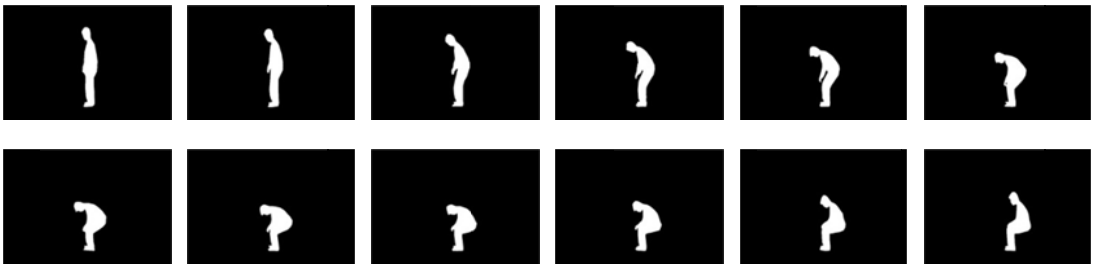


Figure 4-11: The object masks of sitting behavior performed by person 1 and captured from the left side.

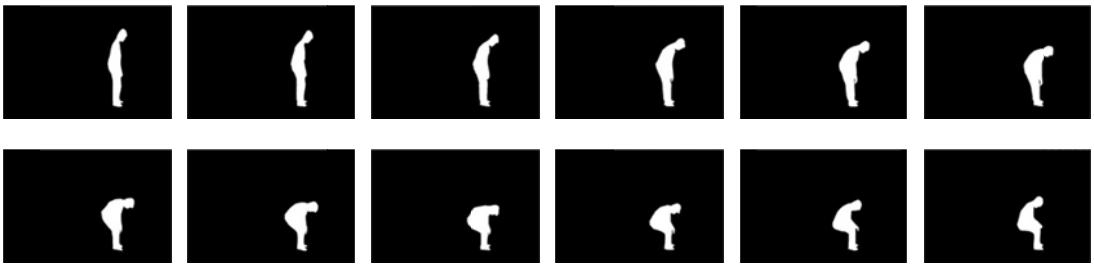


Figure 4-12: The object masks of sitting behavior performed by person 1 and captured from the right side.

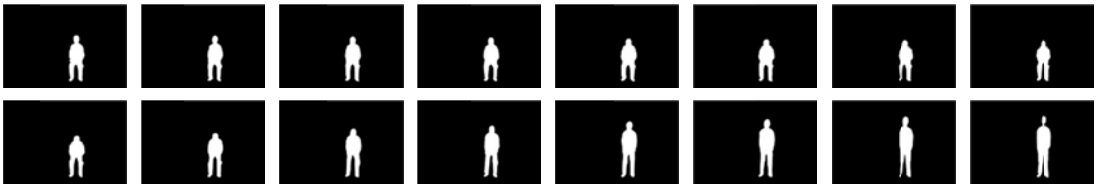


Figure 4-13: The object masks of standing behavior performed by person 1 and captured from the front side.

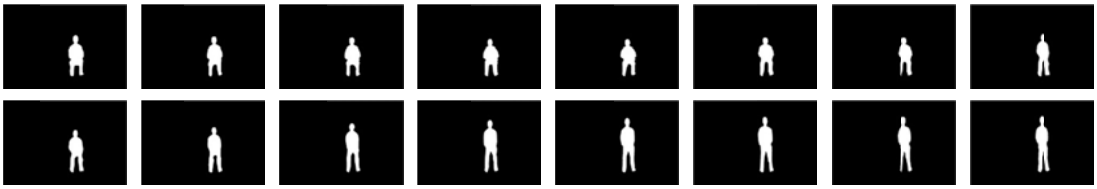


Figure 4-14: The object masks of standing behavior performed by person 1 and captured from the back side.

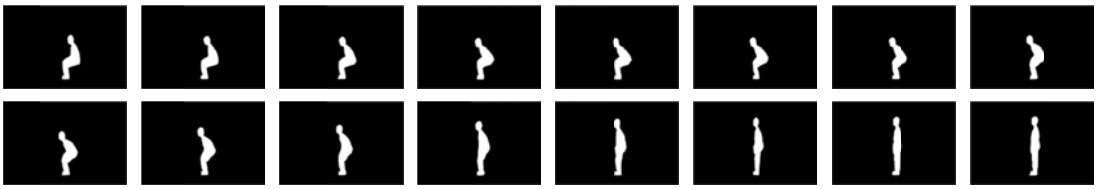


Figure 4-15: The object masks of standing behavior performed by person 1 and captured from the left side.

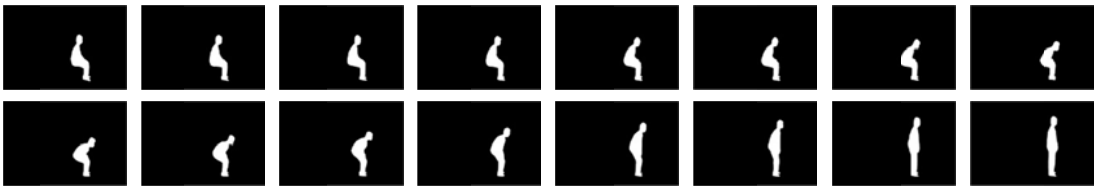


Figure 4-16: The object masks of standing behavior performed by person 1 and captured from the right side.

Figure 4-17: The object masks of falling behavior performed by person 2 and captured from the front side.

Figure 4-18: The object masks of falling behavior performed by person 2 and captured from the back side.

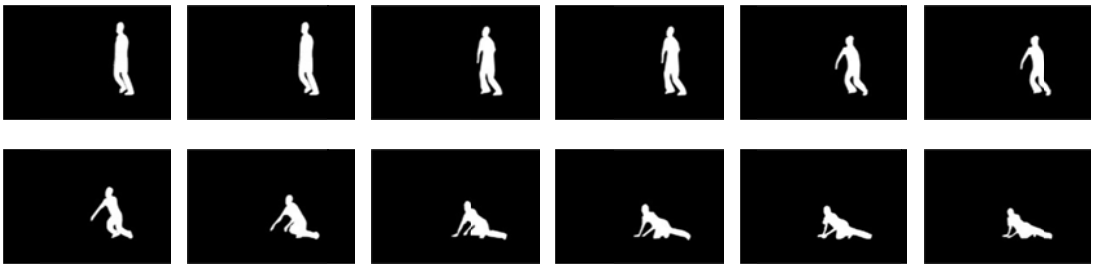


Figure 4-19: The object masks of falling behavior performed by person 2 and captured from the left side.

Figure 4-20: The object masks of falling behavior performed by person 2 and captured from the right side.

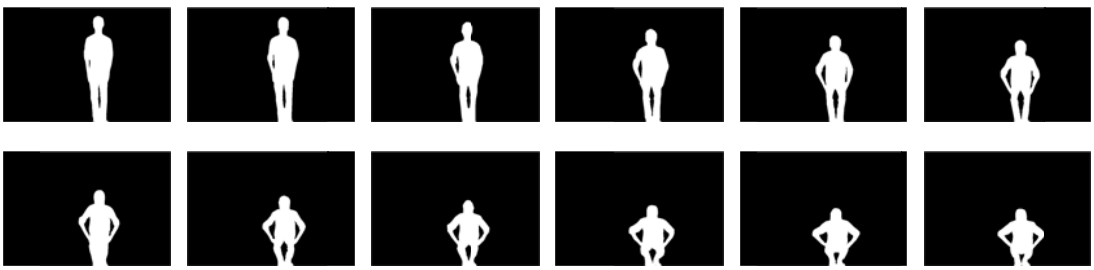


Figure 4-21: The object masks of hunkering behavior performed by person 2 and captured from the front side.

Figure 4-22: The object masks of hunkering behavior performed by person 2 and captured from the back side.

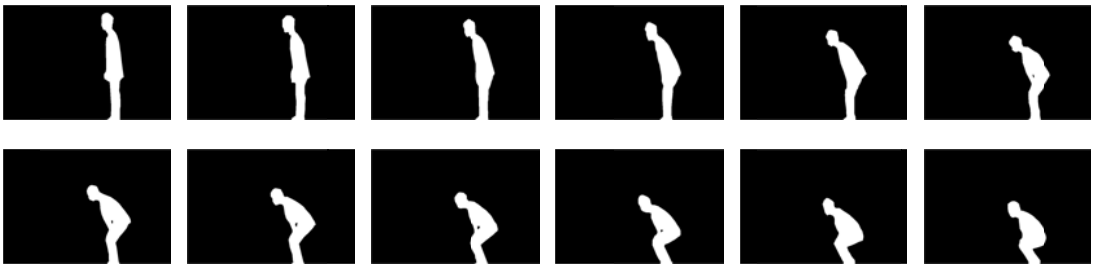


Figure 4-23: The object masks of hunkering behavior performed by person 2 and captured from the left side.

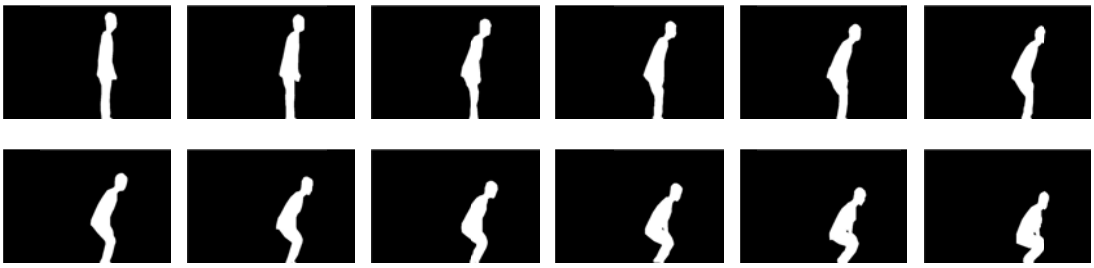


Figure 4-24: The object masks of hunkering behavior performed by person 2 and captured from the right side.

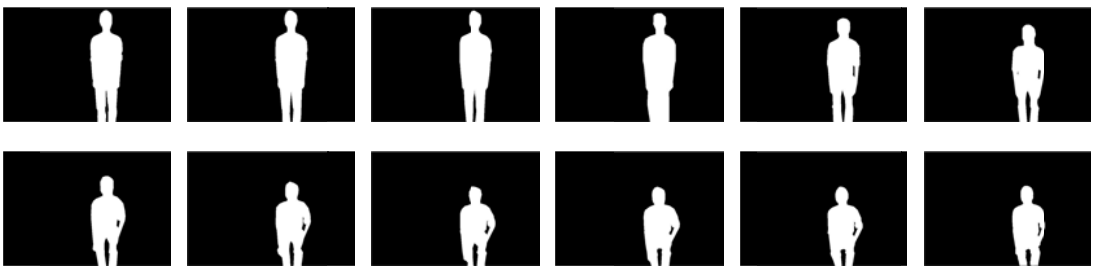


Figure 4-25: The object masks of sitting behavior performed by person 2 and captured from the front side.

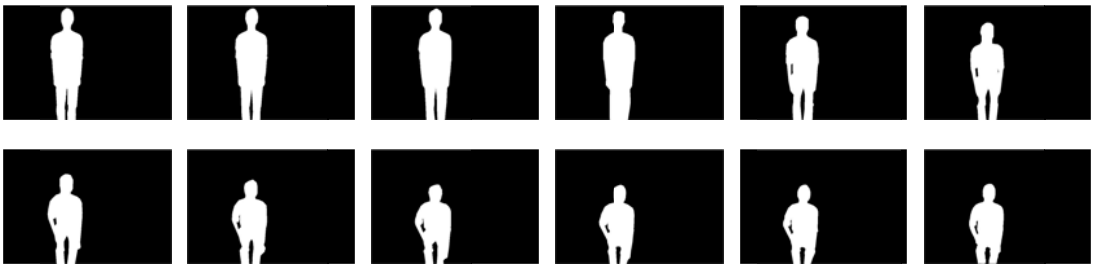


Figure 4-26: The object masks of sitting behavior performed by person 2 and captured from the back side.

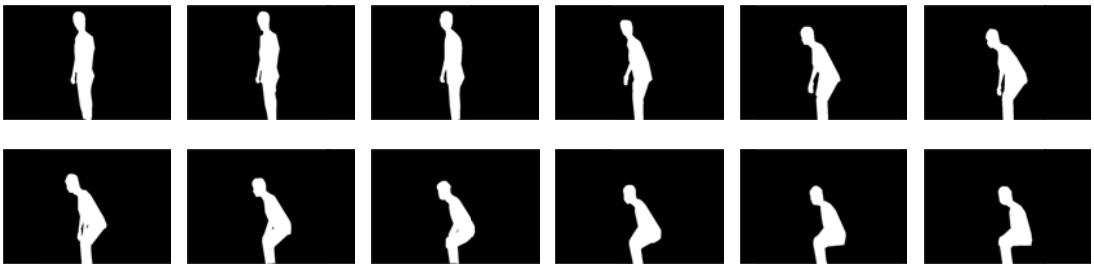


Figure 4-27: The object masks of sitting behavior performed by person 2 and captured from the left side.

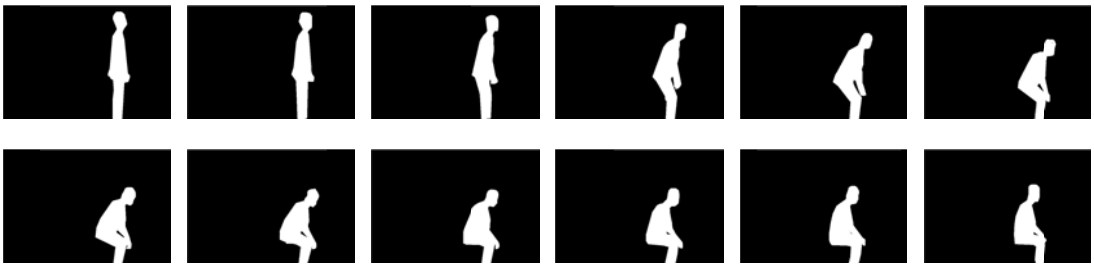


Figure 4-28: The object masks of sitting behavior performed by person 2 and captured from the right side.

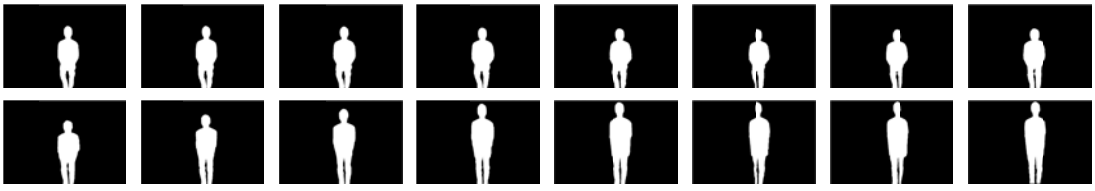


Figure 4-29: The object masks of standing behavior performed by person 2 and captured from the front side.

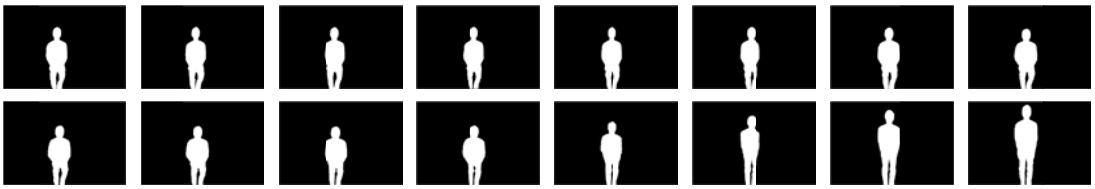


Figure 4-30: The object masks of standing behavior performed by person 2 and captured from the front side.

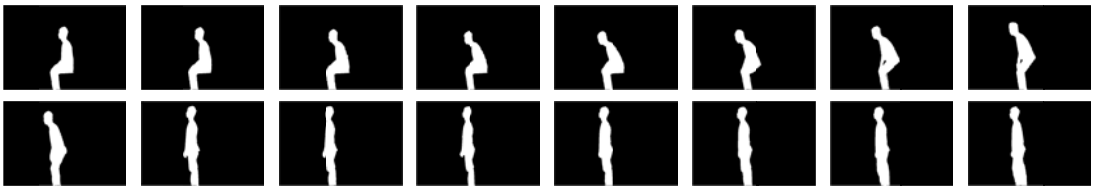


Figure 4-31: The object masks of standing behavior performed by person 2 and captured from the left side.

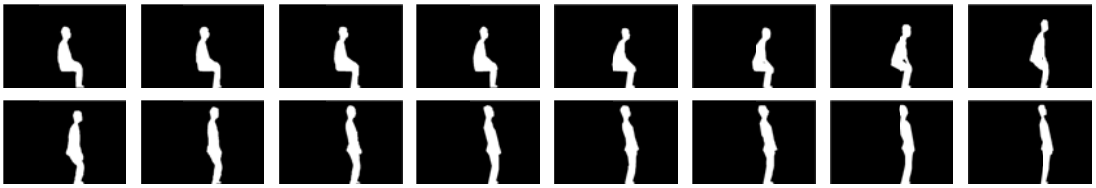


Figure 4-32: The object masks of standing behavior performed by person 2 and captured from the right side.

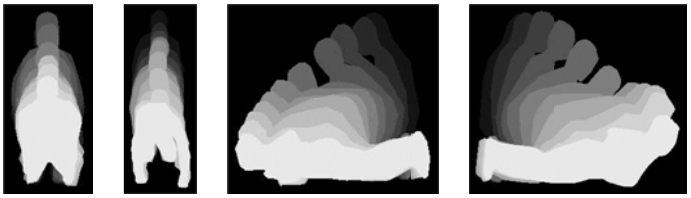


Figure 4-33: The MHIs of falling behavior performed by person 1 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.

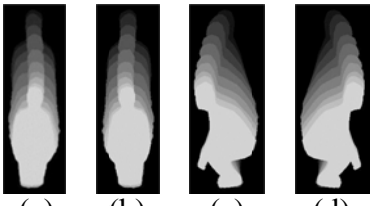
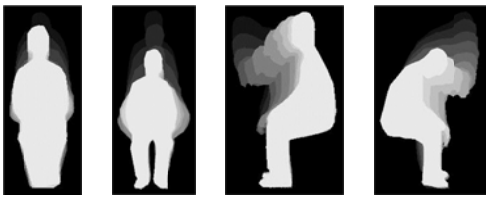
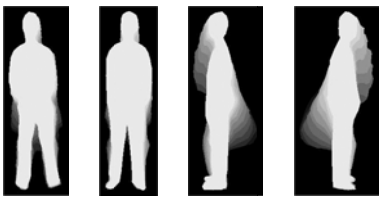


Figure 4-34: The MHIs of hunkering behavior performed by person 1 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



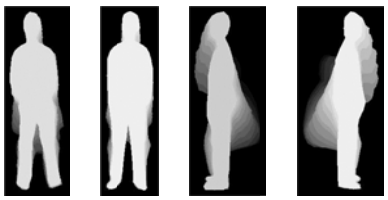
(a) (b) (c) (d)

Figure 4-35: The MHIs of sitting behavior performed by person 1 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



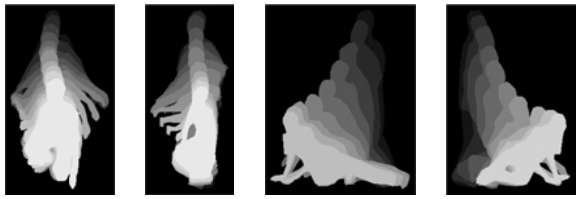
(a) (b) (c) (d)

Figure 4-36: The MHIs of standing behavior performed by person 1 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



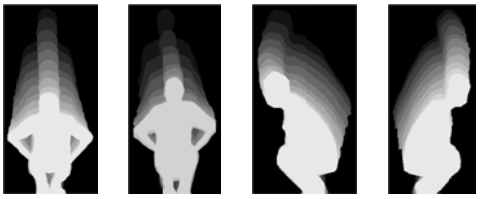
(a) (b) (c) (d)

Figure 4-37: The MHIs of standing behavior performed by person 1 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=16.



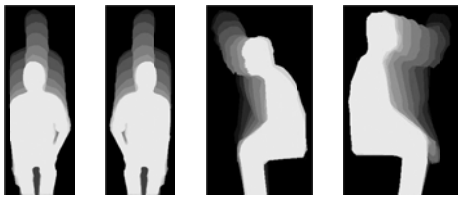
(a) (b) (c) (d)

Figure 4-38: The MHIs of falling behavior performed by person 2 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



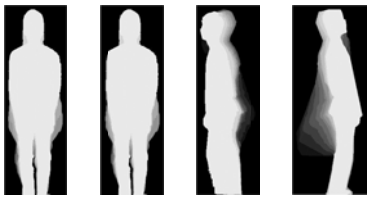
(a) (b) (c) (d)

Figure 4-39: The MHIs of hunkering behavior performed by person 2 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



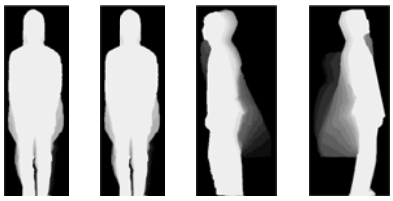
(a) (b) (c) (d)

Figure 4-40: The MHIs of sitting behavior performed by person 2 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



(a) (b) (c) (d)

Figure 4-41: The MHIs of standing behavior performed by person 2 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=12.



(a) (b) (c) (d)

Figure 4-42: The MHIs of standing behavior performed by person 2 captured from the (a) front, (b) back, (c) left, and (d) right sides with delta=16.

After the MHIs are generated, the feature sets can be extracted from the generated MHIs as patterns and the extracted feature sets are stored in the behavior database. In the experiment, there are 40 sets of features calculated by the equations from 2-2 to 2-5, 2-11, and 2-12, and these features should be stored in the behavior database. Table 4-1 and Table 4-2 list these orientation features for $\Delta=\{12, 16\}$ and four types behaviors from four directions performed by the person 1 and person 2, respectively.

Table 4-1: The feature sets for behaviors performed by the person 1.

Delta	Orientation (degree) / Behavior / Block		1	2	3	4	5	6	7	8	9
12	Falling	Front	60.5	70.2	82.0	71.6	52.9	101.2	22.6	13.0	35.8
		Back	83.6	71.3	158.2	87.5	52.4	79.8	41.9	16.2	21.3
		Lift	141.0	32.0	44.8	59.9	38.9	41.7	24.3	26.2	30.3
		Right	32.2	21.2	0.0	34.4	29.1	41.5	43.4	20.3	4.8
	Hunker	Front	148.5	149.2	155.5	159.2	160.6	164.8	164.3	163.5	155.8
		Back	87.0	56.3	89.3	74.3	47.9	83.1	35.4	6.6	9.7
		Lift	52.7	54.6	182.4	43.3	50.2	108.9	34.8	22.3	29.4
		Right	35.0	18.9	18.3	51.0	27.6	19.1	93.5	32.7	16.7
	Sitting	Front	49.7	38.2	82.5	46.3	32.8	64.7	27.8	17.7	27.6
		Back	102.5	40.5	107.8	34.1	32.4	48.4	67.2	7.7	16.1
		Lift	44.3	45.5	17.4	51.9	32.5	4.8	94.7	22.8	32.2
		Right	155.6	58.6	54.3	7.8	33.7	30.8	24.0	35.6	92.6
	Stand	Front	14.8	13.6	22.8	22.9	15.6	5.4	49.0	14.2	16.4
		Back	22.1	14.5	74.4	34.8	25.9	29.4	41.1	34.9	45.6
		Lift	58.3	18.6	0.0	32.6	31.8	61.0	103.7	5.8	17.3
		Right	0.0	13.5	27.5	77.9	28.6	19.1	47.1	16.5	70.9
16	Stand	Front	14.8	13.6	22.5	23.8	15.9	5.3	50.2	14.4	16.4
		Back	22.1	14.5	74.4	36.6	27.4	31.0	50.5	36.0	45.8
		Lift	57.6	22.3	0.0	31.1	36.4	85.0	97.9	6.9	21.8
		Right	96.8	21.6	26.9	99.3	36.8	16.6	59.3	16.6	57.1

Table 4-2: The feature sets for behaviors performed by the person 2.

Delta	Orientation (degree) / Block / Behavior		1	2	3	4	5	6	7	8	9
12	Falling	Front	95.1	77.1	140.2	83.2	56.7	79.0	34.9	21.4	24.8
		Back	0.0	67.7	105.0	59.0	50.2	60.8	124.7	23.3	18.9
		Lift	0.0	27.3	33.4	49.0	35.8	19.7	26.5	37.5	62.9
		Right	43.2	51.3	0.0	34.6	37.0	37.1	47.3	36.4	34.6
	Hunker	Front	43.9	42.5	88.8	58.0	43.6	97.7	62.0	11.8	31.6
		Back	30.8	33.8	68.2	50.8	38.2	71.8	58.9	11.2	47.4
		Lift	59.5	53.9	133.4	39.6	47.9	99.7	43.9	4.4	32.1
		Right	129.2	67.0	64.1	80.5	42.6	29.0	39.0	3.7	19.7
	Sitting	Front	68.9	43.9	86.3	32.5	30.3	53.3	19.4	14.6	22.5
		Back	125.5	21.7	107.2	27.1	16.1	8.9	28.1	2.1	9.8
		Lift	53.7	41.3	107.5	52.2	34.6	23.3	103.3	18.5	6.0
		Right	12.0	25.0	35.9	4.7	26.6	52.2	10.1	5.3	60.3
	Stand	Front	19.9	7.4	14.1	20.1	15.3	4.1	65.2	22.3	11.5
		Back	53.3	19.1	34.0	43.7	20.0	2.3	81.1	11.4	4.8
		Lift	14.9	85.4	85.3	6.6	31.0	66.1	10.1	13.0	50.8
		Right	145.1	33.0	32.5	54.2	29.9	4.4	21.2	35.7	8.6
16	Stand	Front	19.5	7.3	14.1	19.6	17.1	8.3	63.7	26.6	19.3
		Back	33.2	20.4	28.9	65.1	24.8	1.8	90.4	23.0	11.6
		Lift	21.4	99.4	0.0	8.1	31.8	64.3	7.5	19.3	10.9
		Right	45.5	35.2	28.0	56.2	34.1	3.9	43.1	38.9	7.3

In the testing phase, 146 consecutive frames with the size of 704x480 pixels are tested in both Chen's method and the proposed fast MHI approach to evaluate the behavior recognition result and the computing time. The testing frames contain four predefined behaviors and some undefined behaviors which are performed by the first person are taken from the front side. Figure 4-43 lists the object masks of the testing frames.

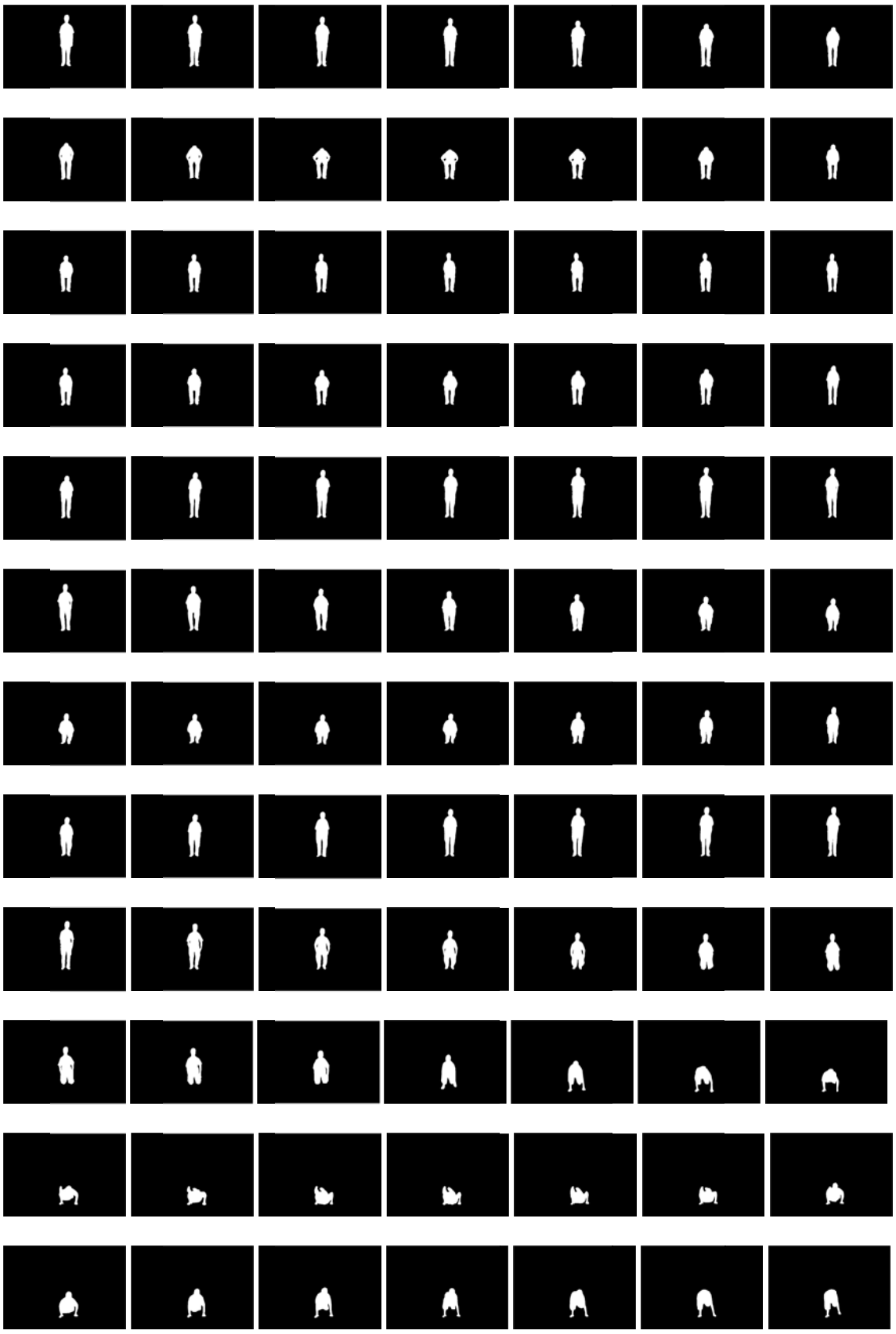


Figure 4-43: The object masks of the testing frames.

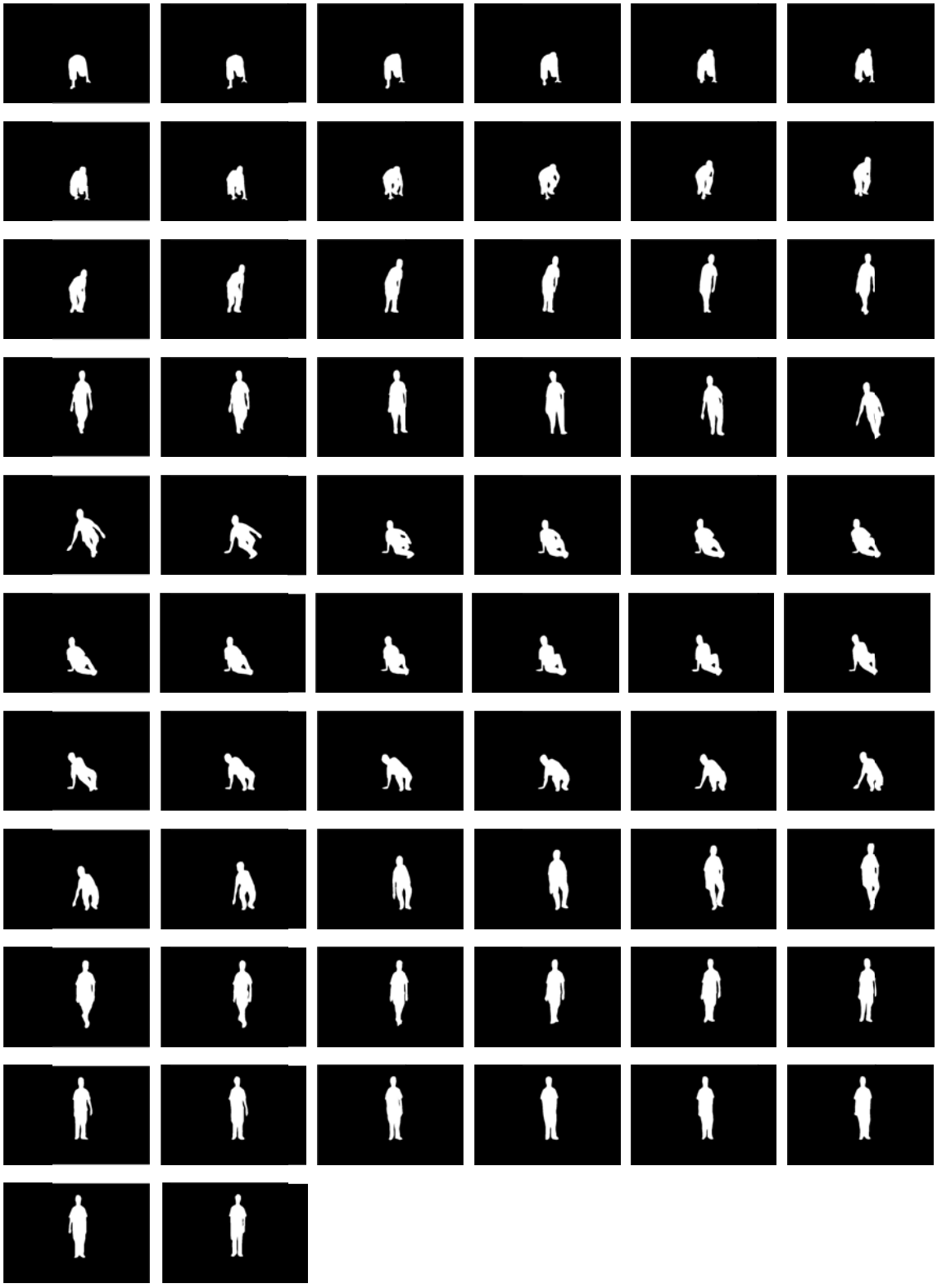


Figure 4-43 (continued): The object masks of the testing frames.

In the recognition process, we define that a behavior is recognized only if the behavior is detected from three continuous generated MHIs. Moreover, if one behavior of the object is recognized, the next behavior of the object must be any behaviors but this current recognized one. For reducing the complexity of computation more, the recognition process will not detect the behavior which is the same to the previous one for a period of $\delta_1/2$ frames, where the δ_1 is the smallest delta in the behavior database. In the original Chen's approach, 72 strengths and nine local orientations are used as a pattern and an error back-propagation neural network is used to recognize the behavior of a video object. In our experiment, the same behavior recognition method as used in our proposed method is also used in the modified Chen's approach for keeping the experimental result independent of the behavior recognition method. In our proposed method, only nine orientation features are adopted for reducing the higher time complexity, and the method described in Chapter 3 is used to recognize behaviors of the video object. The threshold value *THR* is set as 49 degrees in the experiment and which value is determined by the best accuracy of our experiment. The experimental environment for this experiment is listed in the following Table.

Table 4-3: The experimental equipment.

Computer Type	Desk-top computer
CPU	AMD Phenom 9550 Quad-Core 2.20 GHz
RAM	2 GB
Capture Device	Panasonic SDR-H250GT
Development Language	Microsoft Visual C++ 2008 Express

Table 4-4 lists the result of behavior recognition in using the modified Chen's method and our proposed method, where the Meaningless is denoted the behaviors which are not predefined in the behavior database or some petty actions between previous behavior and next behavior. From Table 4-4, we can find that both methods have the same recognition result. That is, in this experiment, nine local information perform very well. However, if a large amount of behaviors were defined in the behavior database, the result may be different.

Table 4-4: The result of behavior recognition
in using the modified Chen's method and our proposed method.

Behavior	Occurrence times	Times for behaviors are recognized	
		The modified Chen's Method	Fast MHI Method
Sitting down	1	1	1
Standing up	4	3	3
Hunkering	1	0	0
Falling	2	1	1
Meaningless	4	0	0

Table 4-5 gives the average execution time of 146 input motion frames for two level deltas (Δ_1) and three level deltas (Δ_2). From Table 4-5, it is very easy to notice that the proposed method can effective reduce the computation time than Chen's method. It is obviously to see that the improvement is mainly from using the multiple sets of features for a predefined behavior and the partial distance calculation. The principle of

storing multiple sets of features is to reduce the unnecessary computations on generating MHIs of higher delta. The more different values of deltas appear in the behavior database, the more computations could be reduced.

Table 4-5: The average execution time for
the modified Chen's method and our proposed method.

Approach	Average execution time (ms)	
	Δ_1={12,16}	Δ_2={8,12,16}
The modified Chen's method	71.8	142.7
The modified Chen's method without using MGMH features	63.6	84.2
The proposed method with multiple sets of features	41.8	58.4
The proposed method with multiple sets of features and partial distance calculation	28.0	49.1
The proposed method with full function	27.7	48.8

In the experiment, there are only two kinds of delta sets (Δ_1 andΔ_2) presented in the behavior database, and the comparison between feature sets from the behavior database and the input frames will get more opportunity to be reduced if there are more elements in the delta set in the behavior database. Comparing to the modified Chen's method, the proposed method can reduce about 61% of computation time for Δ_1, and 65.8% of computation time for Δ_2.

Chapter 5 Conclusions

In this thesis, a new fast MHI approach is proposed to reduce the time complexity of MHI approach by storing multiple sets of features for a predefined behavior to decrease the MHI generation time, using the partial distance calculation skill to reduce the distance calculation time, and changing the calculated order of features to enhance the efficiency of partial distance calculation method. Through the proposed method, the time complexity of the MHI approach can be effectively reduced. To evaluate the proposed method, a set of predefined behaviors are first captured and the features for these behaviors are extracted and stored in the behavior database. Then, a set of testing frames containing defined behaviors and undefined behaviors are used to test our proposed method and the modified Chen's method in terms of the recognition result and the computing time. Comparing to the modified Chen's method, our proposed method can reduce about 61% of computing time for two levels delta set and 65.8% for three levels delta set. From the experimental results, we can also find that the proposed method can get better performance in terms of the computing time when more deltas appears in the behavior database.

References

[1] M. Valera and S. A. Velastin, “Intelligent distributed surveillance systems: a review,” IEE Proc.-Vis, Image Signal Process, Vol. 152, No. 2, pp. 192-204, April 2005.

[2] S. Khan, “Automated versus human traffic control for Dhaka and cities of developing nations,” IEEE 2007 10th international conference on Computer and information technology (ICCIT 2007), Dhaka, Bangladesh, Dec. 2007.

[3] F. F. Meng, Z. S. Qu, Q. S. Zeng, and L. Li, “Traffic object tracking based on increased-step motion history image,” IEEE International Conference on Automation and Logistics, Jinan, China, August 2007, pp. 345-349.

[4] Z. K. Zhang, “The Design and Implementation of Home Security Surveillance System,” Da-Yeh university, Master of Science, Feb. 2008.

[5] C. C. Chang, “The Development of remote monitoring with embedded system,” National Yunlin University of Science & Technology, Master of Science, Jan. 2007.

[6] http://marsprogram.jpl.nasa.gov/mgs/mission/orbiterupdate.html

[7] http://www.nasa.gov/mission_pages/mer/index.html

[8] A. R. Jimenez, A. K. Jain, R. Ceres, and J. L. Pons, “Automatic fruit recognition: a survey and new results using Range/Attenuation images,” ScienceDirect, Pattern Recognition, Vol. 32, pp. 1719-1736, 1999.

[9] F. Lv, S. Y. Zhang, and L. J. Hu, “Image extraction and segment arithmetic of license plate recognition,” IEEE 2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), Vol. 3, Shenzhen, China, Dec. 2009, pp. 378-381.

[10] R. Poppe, “A survey on vision-based human action recognition,” ScienceDirect, Image and Vision Computing, Vol. 28, pp. 976-990, 2010.

[11] S. Antani, R. Kasturi, and R. Jain, “A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video,” ScienceDirect, Pattern Recognition, Vol. 35, pp. 945-965, 2002.

[12] C. Cédras and M. Shah “Motion-based recognition: A survey,” ScienceDirect, Image and Vision Computing, Vol. 13, Issue 2, pp. 129-155, March 1995.

[13] X. Y. Tan, S. C. Chena, Z. H. Zhoub, and F. Zhang, “Face recognition from a single image per person: A survey,” ScienceDirect, Pattern Recognition, Vol. 39, pp. 1725-1745, 2006.

[14] Øivind D. Trier, A. K. Jain and T. Taxt, “Feature extraction methods for character recognition - A survey,” ScienceDirect, Pattern Recognition, Vol. 29, No. 4, pp. 641-662, 1996.

[15] H. Zhou, A. H. Sadka, M. R. Swash, J. Azizi, and U. A. Sadiq, “Feature extraction and clustering for dynamic video summarization,” ScienceDirect, Neurocomputing, Vol. 73, pp. 1718-1729, 2010.

[16] F. Martinez-Contreras, C. Orrite-Urunuela, E. Herrero-Jaraba, H. Ragheb, and S. A. Velastin, “Recognizing human actions using silhouette-based HMM,” IEEE 2009 Sixth International Conference on Advanced Video and Signal Based Surveillance (AVSS), Genova, Italian, Sept. 2009, pp. 43-48.

[17] S. Ali and M. Shah, “Human action recognition in videos using kinematic features and multiple instance learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, Issue 2, pp. 288-303, Feb. 2010.

[18] J. W. Davis and A. F. Bobick, “The representation and recognition of action using temporal templates,” IEEE 1997 Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997, pp. 928-934.

[19] M. C. Roh, H. K. Shin, and S. W. Lee, “View-independent human action recognition with Vol. Motion Template on single stereo camera,” Pattern Recognition Letters, Vol. 31, pp. 639-647, 2010.

[20] M. Allan and C. K. I. Williams, “Object localization using the Generative Template of Features,” ScienceDirect, Computer Vision and Image Understanding Vol. 113 pp. 824-838, 2009.

[21] J. W. Davis, “Hierarchical motion history images for recognizing human motion,” IEEE Workshop on Detection and Recognition of Events in Video, Vancouver, British Columbia, Canada, BC, August 2001, pp. 39-46.

[22] G.R. Bradski and J. Davis, “Motion segmentation and pose recognition with motion history gradients,” IEEE 2000 Fifth Workshop on Applications of Computer Vision, Palm Springs, California, USA, pp. 238-244, August 2000.

[23] M.A.R. Ahad, J.K. Tan, H.S. Kim, and S. Ishikawa, “Solutions to motion self-occlusion problem in human activity analysis,” IEEE 2008 11th International Conference on Computer and Information Technology (ICCIT), Khulna, Bangladesh, 2008, pp. 201-206.

[24] M.A.R. Ahad, J.K. Tan, H. Kim, and S. Ishikawa, “Human activity analysis: concentrating on motion history image and its variants,” IEEE ICROS-SICE International Joint Conference, Fukuoka, Japan, August 2009, pp. 5401-5406.

[25] S. Koelstra and M. Pantic, “A dynamic texture based approach to recognition of facial actions and their temporal models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Issue 99, Feb. 2010.

[26] M. Valstar, M. Pantic and I. Patras, “Motion history for facial action detection in video,” IEEE 2004 International Conference on Systems, Man, and Cybernetics, Vol. 1, pp. 635-640, Oct. 2004.

[27] M. A. R. Ahad, T. Ogata, J. K. Tan, H. S. Kim and S. Ishikawa, “Comparative analysis between two view-based methods: MHI and DMHI,” IEEE 2007 10th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, Dec. 2007.

[28] J. W. Davis, “Recognizing movement using motion histograms,” M.I.T Media Laboratory Perceptual Computing Section Technical Report, No.487, April 1998.

[29] F. T. Chen, “Human behavior identification based on motion history image,” Chung-Hua University, Master of Science, December, 2005.

[30] F. H. Cheng and F. T. Chen, “An efficient method for human behavior recognition,” MVA2007 IAPR Conference on Machine Vision Applications, Tokyo, JAPAN, May, 2007.

[31] Z. Kalafatic, S. Ribaric, and V. Stanisavljevic, “Real-Time object tracking based on optical flow and active rays,” in proceedings of Melecon 2000, Cyprus, May 2000.

[32] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, “Efficient moving object segmentation algorithm using background registration technique,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 7, pp. 577-586, 2002.

[33] James T. Enns, “Object substitution and its relation to other forms of visual masking,” ScienceDirects, Vision Research, Vol. 44, pp. 1321-1331, 2004.