國立交通大學

資訊科學與工程研究所

博士論文

電腦視覺為基礎之多部位人體追蹤系統設計

Design of a Computer Vision-Based Multi-Part Human Tracking System

研 究 生：趙善隆

指導教授：李錫堅　教授

中 華 民 國 九 十 八 年 六 月

電腦視覺為基礎之多部位人體追蹤系統設計

Design of a Computer Vision-Based Multi-Part Human Tracking System

研究生： 趙善隆 Student: San-Lung Zhao

指導教授： 李錫堅 Advisor: Hsi-Jian Lee

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

博 士 論 文

A Dissertation

Submitted to Department of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

電腦視覺為基礎之多部位人體追蹤系統設計

研究生： 趙善隆　　指導教授： 李錫堅

國 立 交 通 大 學 資 訊 科 學 與 工 程 研 究 所

摘　要

本研究我們提出一個在錄影序列中之多部位的人體追蹤系統，首先我們利用一個背景模型偵測並切割出錄影資料中之人體，由於背景影像通常成區塊變化，空間特性可用來表示背景的外觀，為了要建立背景外觀的空間特性的模型，我們為成對的點上之組合顏色建立混合高斯分布的模型。當人體被偵測及切割出來後，接著我們使用人體部位外觀當作特徵並且使用粒子濾波器作為核心來追蹤此人體，我們採等化之顏色統計表當作粒子濾波器中使用的外觀特徵以強化不同物體的鑑別率，為了建立可穩健區別目標物及背景物體的追蹤器，我們同時使用目標的模型和背景模型來計算目標物的相似度，為了對抗背景及目標物的外觀變化，背景模型及目標物模型都是可適應變化的。在一個粒子濾波器中，當粒子數量很多時，特徵抽取的過程會有多餘的重複計算而沒有效率，為了加速特徵抽取，我們為每張影像建立了數張累加的統計圖，每一個粒子的顏色統計表可因此在常數時間中被計算出來。當追蹤人體的時候，我們會把人體切割為三個部位：頭、軀幹、臀腳，這三個部位會分別被表示成為內縮的矩形並用粒子濾波器來追蹤，因為這樣的處理，我們可以檢查這三個部位的一致性以減少可能的追蹤失敗，當追蹤過程中追蹤狀態被更新後，我們會用支持向量機(SVM)來偵測追蹤錯誤並且判斷不正常的部位，假如只有一個部位不正常，我們會校正這個不正常的部位並利用系統動態模型來追蹤此部位，假如兩到三個部位不正常，我們就會從出此三個部位的預估位置重新初始化這三個部位的追蹤。實驗結果顯示，我們提出的背景模型可以有效的偵測背景有改變時及物體在原地移動時的移動物體區域，跟高斯背景模型和混合高斯背景模型比較，我們提出

的方法可以抽取出更完整的移動物體區域。人體追蹤的實驗顯示出，我們提出的三部位人體追蹤及錯誤校正可以正確持續追蹤 95%的人高達 105 個畫面，考慮人體部位追蹤，我們提出的系統可以持續追蹤頭部、軀幹、臀腳這三個部位 105 個畫面的正確率分別高達 95%、83%、91%，和整個人體視為一個部位的追蹤作比較，正確率提升 20%，這個結果顯示出這個系統是一個有效的追蹤系統。

Abstract

The study presents a multi-part human tracking system in video sequences. First, we detect and extract humans in a video according to a background model. Since background images usually change in blobs, spatial relations are used to represent background appearances. To model the spatial relations of background appearances, the joint colors of each pixel-pair are modeled as a mixture of Gaussian (MoG) distributions. After the human is detected and extracted, we then track body parts of the human by using appearances of these parts as the features and using particle filters as the tracking kernel. In the particle filter, we adopt color histograms as the appearance features and use a specific histogram mapping to enhance the discriminability between different objects. To form a robust tracker that can distinguish target objects from background objects that have color distribution similar to those of target objects, we calculate the target similarity from both the target object model and the background model. To handle the appearance variations of background and target objects, both the models of the background scene and the target object are adaptable. In a particle filter, when the number of particles is large, the feature extraction is repeated redundantly and inefficiently. To speed up feature extraction, we create a cumulative histogram map from each image. The color histograms of each particle can then be extracted in constant time. When tracking a human, we decompose the human body into three parts: head, torso, and hip-leg, represent them by three shrunk rectangles, and track them by particle filters. In this way we can reduce possible tracking failures by checking the consistency of states among these three parts. After the tracking states are updated, we use support vector machines (SVM) to detect tracking failures and abnormal body parts. If a single part is abnormal, we adjust its position and use the system dynamic model to track the abnormal one. If two or three parts are abnormal, we re-initialize the tracking process of the three parts around their predicted

positions. Experimental results show that the proposed background model can be used to efficiently detect the moving object regions when the background scene changes or the object moves around a region. By comparing with the Gaussian background model and the MoG-based model, the proposed method can extract object regions more completely. The experimental results of human tracking showed that the proposed three-part tracking system with failure detection and correction can track correctly about 95% persons until the 105th frame. With respect to the body parts, our system has about 95%, 83%, and 91% tracking rates for the head, torso, and hip-leg parts respectively until the 105th frame. The tracking rate of a human increases 20% comparing with that of the whole-body tracker. These rates show the effectiveness of the proposed system.

誌謝

本論文的完成，首先必須感謝我的指導教授李錫堅老師，由於他長期以來的教誨及指導，才能完成我的研究。其次我要感謝的是所有口試委員：陳稔教授、蔡文祥教授、范國清教授、廖弘源教授、黃仲陵教授及余孝先老師，由於諸位老師給予的寶貴建議與指正，使的本論文的內容更加完整。

感謝對我的研究方向及研究態度有著重要啟發的實驗室學長們：曾逸鴻教授、蔡俊明教授、盧文祥教授、陳俊霖博士、楊希明學長、鄭紹余學長。另外還要感謝在實驗室跟我共同研究的同學及學弟們：王舜正、游以正、陳映舟、黃仁贊、吳杭芠、呂偉成、張倍魁、林欣頡、林俊隆、劉一葦、陳思源、曹育誠、陳威甫、陳大任、郭文傑、林鍵彰，在研究的內容及日常生活上，實驗室同伴們都對我有著莫大的幫助，由於大家長期的陪伴，讓我能夠繼續我的研究。

感謝在我到花蓮時提供我許多協助的朋友及學弟妹們，尤其要感謝許展榮先生在我剛到花蓮時，熱心陪我找住所及帶我了解花蓮的環境，感謝邱奕廉和林柔依對我多次到花蓮時交通和居住上的幫助，感謝歐陽儒及陳文祥對我花蓮生活上的幫助，在花蓮幫助過我的朋友太多了，難以一一列舉，由於花蓮朋友們的幫助，我才能順利度過研究的最後難關，因此由衷感謝這些朋友們。

當然還要特別感謝一路支持我的家人們，由於父母親及岳父岳母一直以來給我的支持，讓我更堅定要完成研究的信念。當然更要感謝我的妻子菱芝一直以來對我的包容與鼓勵，在我許多次要放棄研究時，都是因為妻子的鼓勵我才能繼續堅持下去。

最後誠摯的以此研究成果獻給所有幫助過我的人們。

趙善隆

九十八年鳳凰花開時於新竹

目錄

List of Figures

List of Tables

1. Introduction

Human tracking is a fundamental and important step for many visual surveillance applications, such as security guard, patient care, and human-computer interaction. A human tracking system can be divided into two modules, human segmentation and human tracking. The human segmentation module detects and segments a person in a frame. After the person is detected and segmented, the human tracking module then locates the positions of the person in the following frames.

1.1 Background Subtraction

In an indoor environment, people are usually considered to be the only foreground objects, which are defined as ego-motion objects. If the images with only background objects can be captured in advance, the positions of a human can be detected by comparing the current image with the background images. However, background images vary when camera positions, background object positions, and illuminations change. Tracking objects in general environments will become very complicated.

In many surveillance applications especially in indoor environments, camera positions are generally fixed. Illumination variations and background object motion may change the captured images significantly. Examples of the motions include placing a book on a desk and moving a chair to another position. The positions of the objects are usually changed by people or other external forces. After the motion stops, these objects remain in the same position for a certain period; these motions are usually not repeated. In an indoor environment, the illumination of objects is not affected by continuous light changes, such as sun rise, sun set, or weather changes. Ignoring these continuous changes, brightness variations such as turning lights on or off, as shown in Fig. 1.1, and opening a window are assumed to be abrupt. Several

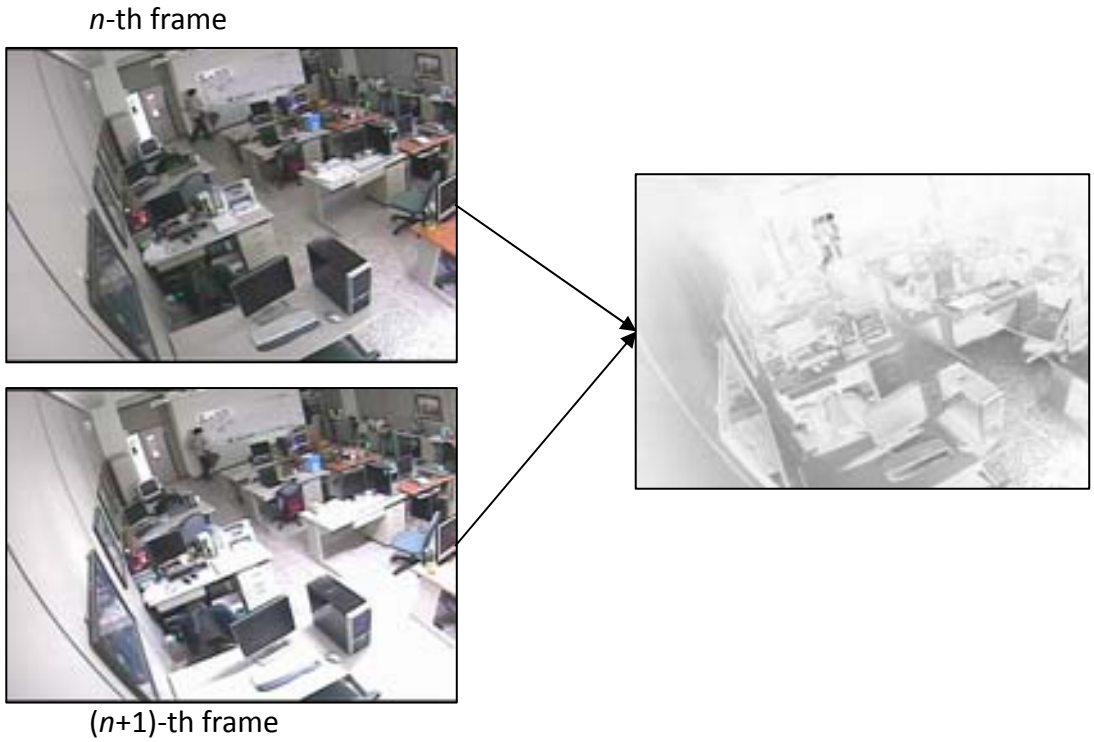


Fig. 1.1 Left column: two consecutive images in different illumination conditions. Right image: intensity differences between the left two images.

researchers [1] assumed that brightness variations due to illumination changes are uniform. The right-hand side image in Fig. 1.1 shows the intensity differences after the lights were turned on. We observe that brightness variations in different pixels are not uniform. It is difficult to process this kind of variations. Several other researchers [1-4] assumed that illumination changes are not repeated like the motion of freely movable objects. However, light sources can be repeatedly turned on or off several times over a period of time. The appearance changes on the illuminated regions will also be repeated.

To model the non-repeated background changes, we can use an online updating scheme to adapt to the background appearances in recently captured images [1-17]. When the appearances of a pixel repeatedly change, they can be modeled as a Mixture of Guassians (MoG) [9]. The online updating MoG model is useful for modeling rapidly repeated

background appearances such as waves on water surface, but does not works well in long-term repeated appearances such as door opening and closing. In consecutive images, the repeated appearances of background objects usually appear in blobs in fixed places, while the appearances of foreground objects usually change their places and do not form fixed blobs. In this study, we extend the MoG model by using the spatial relations among pixels to model the background appearances.

The objective of our system is to extract moving object from a sequence of images. The system is divided into two modules: background modeling and foreground detection. The first module creates a background model to represent possible background appearances. The parameters of the model are learned and updated automatically from recently captured images. In the background model, the distributions of background features are assumed to be mixtures of Gaussians [9]. Since background appearances are changed in blobs, the features used in the MoG should be able to represent spatial relations in the blobs. To represent the spatial relations, we estimate the joint color distributions of pixel-pairs in a short distance. Since estimating the distributions of all pixel-pairs is costly and not all pixel-pairs provide enough information to model background, we first calculate the dependence of colors in each pixel-pair. A pixel-pair with a higher color dependency implies that the two pixels provide more information to represent the appearance changes in blobs. Highly dependent pixel-pairs are then selected to model the spatial relation of background. In the second module, the background model that has already been updated from recent images is used to calculate the background probability of each pixel of the current image. The probability is then used to decide whether the pixel belongs to the foreground or background. Connected foreground pixels are extracted to form foreground regions.

1.2 Human Tracking

Fig. 1.2 Example of the three body parts used on tracking a person. The three body parts are shrunk and the limbs are excluded to reduce the affection from human motion.

To track an object in a sequence of frames, we can model appearances of the object and then use the model to predict its position in the sequence. However, in a complex environment, detecting a target object using the appearance model in video sequences is not easy since the appearances of the object are variable due to occlusion, illumination variations, or orientation changes. In general, the movements of an object in consecutive frames are assumed smooth. Therefore, if we can locate the target object in several frames, the appearance model and movement model of the target object obtained from these frames can be used to track the object in the following frames.

In this study, we aim to create the trajectory of a human and predict his positions for safeguarding, that is, to detect an intruder approaching a building or a designated place. Since a human is not a rigid object, his appearance might be greatly affected by his motion. We decomposed the human body into three parts: head, torso, and hip-leg, since the three parts usually have different appearances and can be distinguished as shown in Fig. 1.2. The images show that the colors of the head part contain mostly skin colors and hair colors, which are

usually different from the colors of the other two parts. The colors of the torso and hip-leg parts consist mainly of those of the clothes, which may be similar, as shown in the fourth and fifth images of Fig. 1.2. To separate the three parts, we have to use other features such as height ratios.

With respect to the features used, we adopted color histograms proposed by Perez et al. [18] and Nummiaro et al. [19] to model the appearances of the three body parts. In the initialization phase, we adopted the background subtraction method according to a Gaussian background model to extract a human and then extract the histograms of the body parts from the human region. We then tracked the humans by using their appearances as the features and tracked the three parts by particle filters to reduce possible failures due to appearance changes by checking the consistency of states among these three parts. Since the appearance model of each person in recent frames was usually unique and temporally context-dependent, the model can be used to distinguish different persons and track them independently. However, when modeling the color histogram in the whole color space, histogram matching was time-consuming due to the high dimensional features used. The method proposed by Nummiaro et al. [19] quantized the color histogram into an $8 \times 8 \times 8$ or $8 \times 8 \times 4$ three-dimensional one. The method proposed by Perez et al. [18] modeled colors in HSV color space by two histograms. The intensity channel was modeled as a histogram and the other two channels as another two-dimensional histogram. The histograms were quantized into several bins to improve the speed and reduce the effect of noise. However, in these models, two objects with very few dissimilarities were not easily distinguished. In our research, we propose a specific histogram mapping for histogram feature extraction to improve the ability of discriminating the objects with similar color distributions. Since the camera in our system is fixed, the background scene can be assumed less changed in consecutive frames. To improve the discriminability between the target object and background

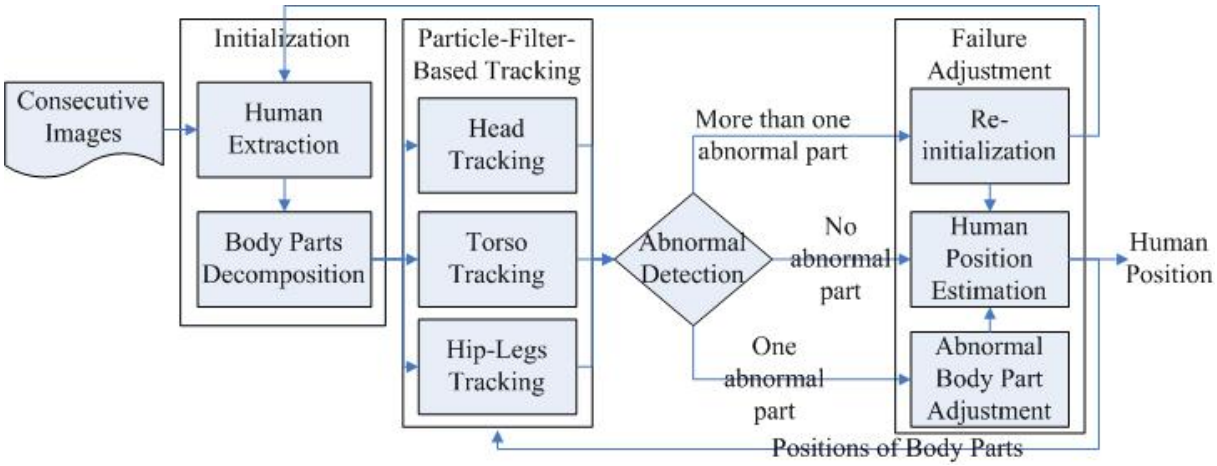


Fig. 1.3 The system flow diagram of the three-part human tracker.

objects, we combined the adaptive background model with the adaptive color histogram model of the target object.

When adopting the color histograms as the features used in a particle filter, we need to extract the histogram feature for each particle. It is generally very inefficient to extract the features for a large number of particles. In this research, we will create a cumulative histogram map (CHM) for each image to improve the efficiency of feature extraction. The cumulative histogram map is similar to the integral map that is popularly used for extracting Haar-Like features [20]. They will be modified to cumulate the histogram features of each sample state in constant time.

For failure detection and adjustment, we will use a support vector machine (SVM) [21,22] to distinguish abnormally and normally tracked body parts. The position of an abnormal body part will be adjusted according to its relative positions with the other body parts. If a single part was abnormal, we adjusted its position and used the system dynamic model to track the abnormal one. If two or three parts were abnormal, we re-initialized the tracking process of the three parts around their predicted positions. Next, we detect whether the failure is caused by occlusion or similar appearances. For the latter case, we will estimate

the appearance model from the adjusted rectangle of the body parts; else, the appearance model is kept unmodified. The flow diagram of our tracking system is depicted in Fig. 1.3. It includes four major modules: initialization, particle-filter-based tracking, abnormal body part detection, and state correction.

1.3 Organization of This Dissertation

The rest of this dissertation is organized as follows. Chapter 2 is a review of related research. Chapter 3 describes the proposed spatially-extended background model. Chapter 4 describes the particle weight measurement and the cumulative histogram map used to improve the calculation speed. Chapter 5 describes the three-part human tracking and consistency checking for failure adjustment. Chapter 6 gives experimental results and their analysis. Finally, Chapter 7 presents the conclusions and future works.

2. Related Works

2.1 Background Subtraction

A background model in a surveillance system represents background objects. The method that compares the current processed image with the background representation to determine foreground regions is called background subtraction. If the background is unchanged but affected by Gaussian noise, the colors of the background pixels can be modeled as a Gaussian distribution with mean vector (μ) and covariance matrix (Σ) [1-8]. Background subtraction is then performed by calculating the probability of each pixel in the current image belonging to the Gaussian model.

Since background appearances may be affected by external forces, modeling a pixel with a Gaussian distribution may misclassify some background pixels as foreground ones. In many cases, the background may change repeatedly. A background pixel with repeated changes can be divided into several background constituents and modeled as an MoG distribution [9-13]. For each background constituent in a pixel, the means (μ_i), covariances (Σ_i), and weights (w_i) of the *i*-th constituent (b_i) have to be estimated. If there are K background constituents, the parameters of the background model can be represented as $\{\mu_i, \Sigma_i, w_i | 1 \leq i \leq K\}$ In order to decide whether a sample point X belongs to the background B, the conditional probability $P(B|X)$ is calculated as follows:

$$P(B|X) = \sum_{i=1}^{K} w_i P(b_i|X) \propto \sum_{i=1}^{K} w_i \eta(X; \mu_i, \Sigma_i). \quad (2.1)$$

where η represents a Gaussian probability density function,

$$\eta(X;\mu_i,\Sigma_i) = \frac{1}{2\pi^{1/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(X-\mu_i)^T\Sigma_i(X-\mu_i)}. \quad (2.2)$$

The motion of some background objects may not be repeated. After the motion, the objects remain in the same position for a period. To model the background changes, researchers have proposed methods for online updating of the parameters of background models [1-17]. The mean vector and covariance matrix in time *t* are represented as μ_t and Σ_t, respectively. The updating rules are formulated as follows:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t, \quad (2.3)$$

$$\Sigma_t = (1-\rho)\Sigma_{t-1} + \rho(X_t-\mu_t)(X_t-\mu_t)^T, \quad (2.4)$$

where ρ is used to control the updating rate. To integrate the updating method into an MoG model, Stauffer and Grimson [9] proposed a method to update the mean vector and covariance matrix of a background constituent to match those of X_t in Eqs.(2.3) and (2.4). The weight $w_{i,t}$ of the *i*-th background constituent is updated as follows:

$$w_{i,t} = (1-\alpha)w_{i,t-1} + \alpha I_{i,t}, \quad (2.5)$$

where $I_{i,t}$ is an indicator function, whose value is one if the *i*-th background constituent matches X_t and zero otherwise, and α is a constant used to control the updating rate of the weights. In Stauffer and Grimson's method [9], the updating rate ρ for the parameters of the *i*-th constituent (Gaussian distribution) is calculated according to α and $\eta(X;\mu_i,\Sigma_i)$.

To make background models more robust, researchers tried to modify updating rules or adopt different features [10-13]. In adaptive background models, background objects are assumed to appear more frequently than foreground ones. However, the

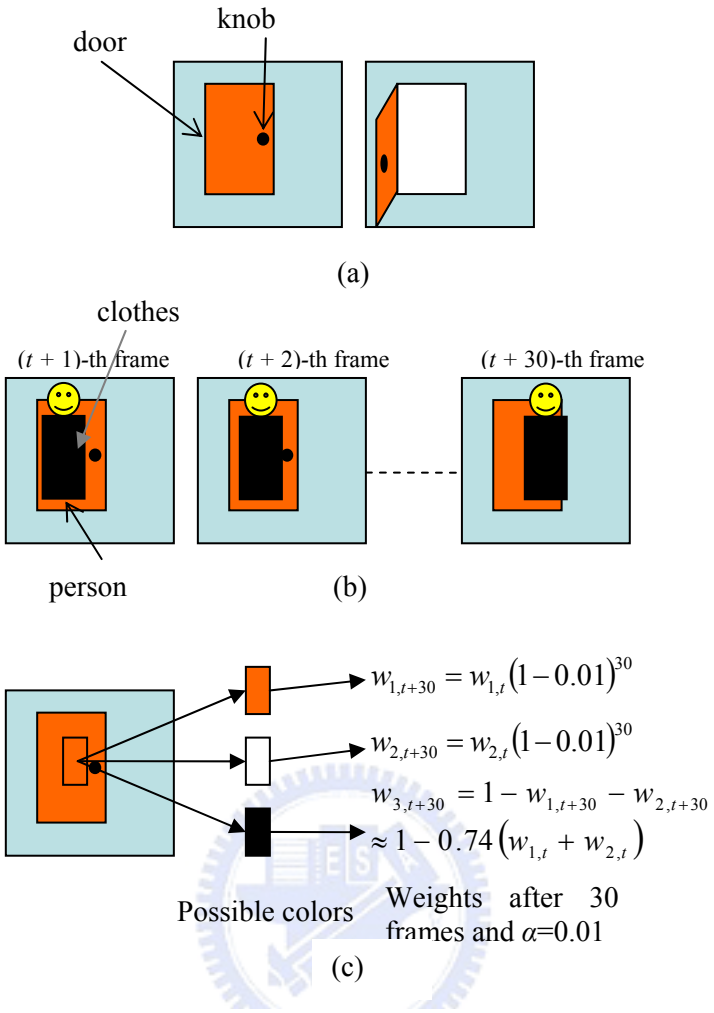
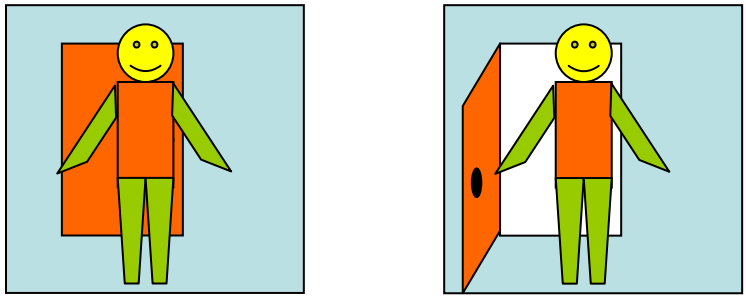


Fig. 2.1 An example of a slowly moving person. (a) Sketches of two possible background scenes. Left: door closed; Right: door opened. (b) Consecutive frames of a person moving from left to right. (c) Possible colors and their weights of the rectangle region shown in the left image.

assumption is not always satisfied. If the appearances of a background pixel appear less frequently than those of foreground objects, the background pixel is probably misclassified as a foreground object. Taking the following case as an example, assume a room is monitored by a fixed camera and the background objects in the room include a door and a wall as shown in Fig. 2.1(a). People may enter the room, and close or open the door. If a person wears a suit of clothes of single color and walks



(a) (b)

Fig. 2.2 Sketches of a person whose clothes colors are similar to the door color in front of different background scenes: (a) scene when the door is closed and (b) scene when the door is opened.

slowly across the room as shown in Fig. 2.1(b), the major color of the clothes may be captured repeatedly in a certain position among several consecutive images. Assume that the person moves from left to right in 30 frames. If the updating rate α in Eq.(2.5) is set as 0.01, the weight of the repeatedly captured color of the clothes in the images will increase from 0 to 0.26, as shown in Fig. 2.1(c). This large weight may cause the clothes to be labeled as the background, when the MoG model in Eq.(6.2) is used. Using a small updating rate can overcome this problem; however, the background model will be updated very slowly and may fail to learn background changes.

In another situation, the color of the person's clothes is assumed to be the same as that of the door, as illustrated in Fig. 2.2(a). If the person enters the room and passes through the door, the region of clothes may be labeled as background due to the similarity of colors. However, after the door is opened, the current background color is not similar to the clothes color as shown in Fig. 2.2(b). The clothes may still be labeled as the background, since they are very similar to possible background colors been estimated.

In these two situations, we observe that modeling each pixel independently

cannot sufficiently represent the similarity among different object appearances caused by either object motion or illumination changes. Most researchers regarded the variations caused by object motion as foreground changes and attempted to eliminate the effect caused by illumination. In consecutive images, when the illumination changes, the pixels of an object are usually changed simultaneously. In order to model background objects, the pixels at different positions should be considered together. To represent the relation among the pixels, Durucan and Ebrahimi [14] proposed to model the colors of a region as vectors. They segmented the foreground regions by calculating the linear dependence between the vectors of the current image and those of the background model. However, it is expensive to represent the dependence between vectors in terms of storage and speed. To reduce the cost, the vector of a region should be reduced to a lower dimensional feature. Li et al. [15,16] used two-dimensional gradient vectors as the features of local spatial relations among neighboring pixels. In their proposed method, the appearance variations caused by illumination changes can be distinguished from object motion. However, the gradient features cannot be used to extract the foreground region that has a uniform color. To model the relation among pixels, we need to use the relations among near pixels to reduce time and storage consumption, and then extend the relations into a more global form.

The methods based on the Markov random field (MRF) are well known for extending the neighboring relations among pixels into a more global form. Image segmentation methods based on MRF [17,23] assume that most pixels belonging to the same object have the same label and these pixels form a group in an image. The MRF combines colors among a clique of pixels in a neighboring system and uses an energy function to measure the color consistency. Then, the maximum a posterior

estimation method is used to minimize the energy for all the cliques to find the optimal labels. In the MRF-based methods, the final segmentation results are strongly dependent on the energy functions of the labels in different cliques. If a high energy is assigned to the clique with unique labels, the extracted foreground regions will become more complete than those of the pixel-wise background models. An additional noise removal process is not required. However, when several pixels are mis-labeled, these errors will propagate into neighboring pixels. The error propagation will cause more pixels to be mis-labeled. In this research, we directly estimate the relations among pixels instead of the labels, and therefore the errors will not easily propagate.

2.2 Human Tracking

In the last few decades, tracking objects or humans in video sequences has received much attention. Much research about the topic has been proposed and been reviewed in several survey papers [24-28]. Moeslund et al. [24] divided a general human tracking algorithm into two main phases: *figure-ground segmentation* and *temporal correspondences*. The former finds the target human in an image, and the latter associates the detected humans in consecutive frames to create temporal trajectories. In the following, related work about these two phases will first be addressed. The methods for segmenting human bodies and correcting tracking failures will then be described.

The methods of *figure-ground segmentation* can be classified into five categories according to the used features. These categories include *background subtraction* [6,9], *motion-based segmentation* [29], *depth-based segmentation* [30], *appearance-based segmentation* [18,19,21,31,32], and *shape-based segmentation* [2,33]. Background

subtraction and motion-based segmentation methods find the differences between images to extract the target. The two approaches assume that only one target object moves in a specific region and the appearances of background objects in consecutive images do not change. However, the assumptions usually cannot be met in general environments. To achieve better segmentation results for tracking, further checks are needed. The depth-based segmentation approach uses the positions of the target in three-dimensional space or in the ground plane to segment the target. However, to locate such kind of positions, specific hardware (such as multiple cameras) or additional calculations (such as inverse perspective transform) are needed. The appearance-based segmentation approach became popular recently, since the approach is usually simple and fast. The approaches of shape-based and appearance-based segmentation are similar except that the former does not use the color content inside the object. Since the appearances of a tracking target may change with time, several researchers proposed methods to model and update the appearance model of the target person dynamically in consecutive images [19,32]. Since the target is a moving object, some researchers tried to segment the target by a background subtraction method [6,9,34]. Shan et al. [34] modeled the colors of the target object as the appearance feature and then use the color cue to calculate a color probability distribution map from the current image. The color probability distribution map was combined with the background subtracted image using a logical AND operation to detect the target position. Other researchers used classifiers such as SVM [21] and Adaboost [31] to model the appearance of target objects.

In the tracking phase, *temporal correspondence* aims to predict and update the states of the target person from the measurement and predicted state, where the measurement is detected by figure-ground segmentation and the predicted state is

calculated using the system dynamic model. To find the temporal correspondences, Polat et al. [35] used MHT (Multiple Hypothesis Tracker) to construct hypotheses representing all the predictions and measurements. The most likely hypothesis is chosen as the target. To combine the predictions and measurements, Kalman filtering is another well-known method and has already been applied in many studies [3,9,36]. The Kalman-filter-based approaches are commonly used for tracking a target whose system dynamic model can be represented as a linear function and the noise as a Gaussian. In non-linear systems, extended Kalman filters that approximate the non-linear dynamic model by Taylor series have been applied [37]. Recently, particle filters have been proposed to construct a robust tracking framework that are neither limited to linear dynamic model nor Gaussian distributed noise [19,38,39]. The method represents the state of a target object by a set of samples (particles) with weights. The weight of a sample is calculated by the figure-ground segmentation and the samples are generated by the importance sampling method so that the samples can represent the probability distributions of the target object's appearances. We adopt the particle filter in our system, since they can be applied in an appearance-based tracking system very effectively.

A human is not a rigid object and his appearance changes irregularly. Segmentation of human body parts in an image has already been proposed in several papers [40-43]. Forsyth and Fleck [40] introduced the notion of 'body plans' to represent a human or an animal as a structured assembly of body parts learnt from images. Shashua et al. [41] divided a human body into nine regions, for each of which a classifier was learnt based on features of orientation histograms. Mikolajczyk et al. [42] divided a human body into seven parts. For each part, a detector was learnt by following the Viola-Jones approach applied to scale invariant orientation-based

features. Ioffe and Forsyth [43] decomposed the human body into nine distinctive segments. The method finds a person by constructing assemblies of body segments. The segments were consistent with the constraints on the appearance of a person that result from kinematic properties. These body-parts-based human segmentation methods usually focused on detecting humans in a static image. Recently, body-parts-based human tracking in consecutive images has been proposed [21,31,44-47]. Parts of these studies focused on precise decomposition of body parts for motion type or pose analysis. However, in general environments, it is difficult to decompose precisely body parts due to self occlusions and complex background scenes. The studies in [21,47] proposed a detection-based tracking model to solve the occlusion problem. They detected body parts by a pretrained model, and then tried to associate the detected body parts to a target person by smoothing his trajectory. However, when multiple humans appeared in a frame, the detection model could not differentiate the body parts of the different persons. The spatial positions and velocities were the only cues, which can be used to find the temporal correspondences of different persons.

When a person is tracked in consecutive frames, the figure-ground segmentation may fail, since the person may be occluded or other objects may have similar appearances with the target person. The first problem can be classified into occluded by other persons and occluded by background objects. To cope with the problem of inter-person occlusion, several researchers proposed to detect occlusion events and then used the system dynamics to estimate the position of the occluded person [48,49]. Using similar methods to predict the occlusion of background objects, one needs to create background object models. However, it is difficult to model all background objects in a complex environment. Several researchers tried to cope with the

occlusion problems by tracking different body parts of a person simultaneously [21,47,48]. When a body part is occluded, the position of the person can still be tracked based on the other parts. Mohan et al. [21] extracted a human body by detecting four parts: the head, legs, left arm, and right arm, by four distinct quadratic support vector machines. After geometric constraints among these parts are confirmed, another support vector machine is used to classify the combination of the four parts as either a human or a non-human. Wu and Nevatia [47] used four detectors to detect head-shoulder, torso, legs, and full-body. The detectors were learnt by a boosting approach using edgelet features. They used a strong classifier to classify the body parts in images. When we track multiple humans, the classifier cannot be used to distinguish different persons, and their trajectories will easily be confused if no other approaches are adopted. Lerdsudwichai et al. [48] proposed a method to model the face and clothes in the initialization phase. To identify different persons, they used clothes colors. However, the appearance of a person may change when the person presents different poses or is affected by different illuminations. The appearances captured in the initialization phase cannot be applied to track the person in other frames. In our research, we use an adaptive appearance model to track the body parts, even when multiple persons are tracked.

Apart from the occlusion events, a tracker may lose the tracking target when other objects have similar appearances. In general, a robust appearance model can be used to reduce the tracking failures, or the system dynamic model of the target person can be used to predict his position. However, the robust appearance model may be too complex to maintain efficiently. We will use the system dynamic model of the target person to track him, when a tracking failure is detected. To detect the tracking failure, Dockstader and Imennov [36] proposed a method that uses a structural model to

represent a person and a hidden Markov model (HMM) to describe the temporal characteristics of the tracking failure. In the tracking phase, the HMM was used to predict the tracking failures. Since a person may change his velocity, the predicted position of the person using the system dynamic model is too rough. A fine tune for the target person is needed.

3. A Spatially-Extended Background Model

In the initialization step of a tracking system, the human region in a image needs to be segmented for tracking. To segment the human region, we create a background model and update it using recent background variations. Since background images are usually changed in blobs, spatial relations are used to represent background appearances, which may be affected drastically by illumination changes and background object motion. To model the spatial relations, the joint colors of each pixel-pair are modeled as a mixture of Gaussian (MoG) distributions. Since modeling the colors of all pixel-pairs is expensive, the colors of pixel-pairs in a short distance are modeled. The pixel-pairs with higher mutual information are selected to represent the spatial relations in the background model. Experimental results show that the proposed method can efficiently detect the moving object regions when the background scene changes or the object moves around a region. By comparing with Gaussian background model and the MoG-based model, the proposed method can extract object regions more completely.

3.1 Joint Background Model

In a sequence of images, colors will change in blobs instead of individual pixels due to illumination changes or object motion. This paper proposes to utilize the relations among pixels to represent the changes in blobs. The relations are formulated as a spatially-extended background model, which is then used to classify the pixels into either foreground or background.

3.1.1 Spatial Relation in Images

Using pixel-wise features, if the color of a foreground pixel is similar to those of the background, the pixel may be misclassified as background. If we can estimate the distributions of color combinations for the pixels in blobs, the foreground objects can be classified more precisely. Suppose there is a red door in a room and the appearance outside the door is white. When a person wearing a suit of interlaced red and white stripes passes through the door, parts of the suit may be misclassified as background when the colors of the pixels are modeled independently. Nevertheless, if we model the background appearances among pixels using joint multi-variate color distributions, the interlaced red and white stripes can be classified as foreground using the method introduced later. However, estimating the multi-variate distributions for all pixel-pairs is still costly since the number of pixel combinations may be very large. In this research, we will estimate the color distributions of joint random vectors in closed pixel-pairs.

As stated in Sec. 2.1, illumination changes and background object motions may change background appearances. Since the changes are complex, it is difficult to collect enough training samples for all the possible changes. In this paper, we modify Eqs. (2.3) and (2.4) for updating the color distributions of pixel-pairs to adapt to the appearances that have not been trained, to be described in Sec. 3.1.3.

3.1.2 Calculation of Background Probabilities

Assume that we have already estimated the color distributions of all background pixel-pairs. In this research, we decide whether pixel a_1 belongs to foreground according to its color and the color combinations of pixel-pairs (a_1, A), where A denotes a set of pixels associated with a_1.

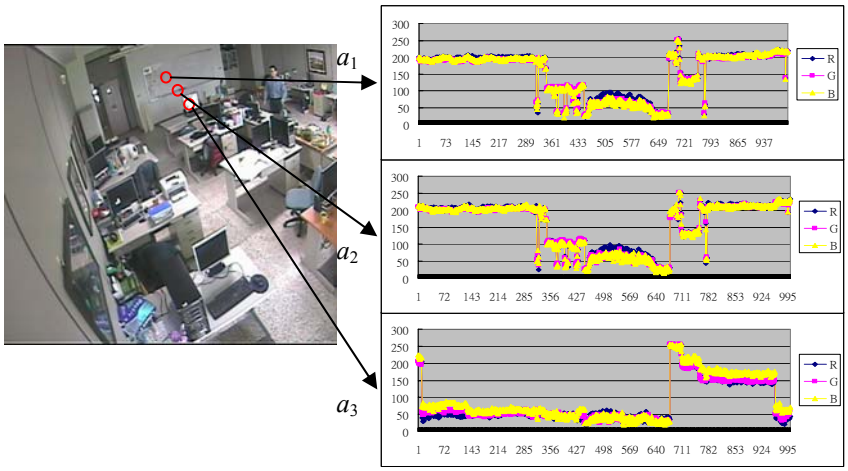
Suppose that a sequence of pixels $(a_0, a_1, \dots, a_n)$ has a corresponding color sequence $(c_0, c_1, \dots, c_n)$. The probability of pixel a_0 belonging to background can be

represented as $P(B_0|x_0 = c_0, x_1 = c_1, \dots, x_n = c_n)$, where the sequence $(x_0, x_1, \dots, x_n)$ denotes the joint random variable of the colors for the sequence $(a_0, a_1, \dots, a_n)$, and B_0 represents the event that pixel a_0 belongs to the background. Assuming that $x_1, x_2, \dots, x_n$ are conditionally independent, based on the naive Bayes' rule, the probability $P(B_0|x_0 = c_0, x_1 = c_1, \dots, x_n = c_n)$ can be computed as the product of n pair-wise probabilities:

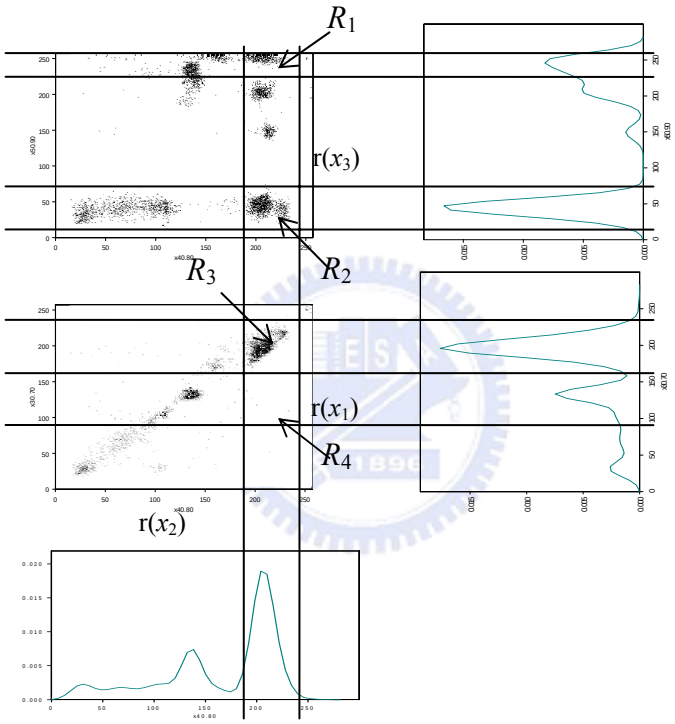
$$P(B_0|x_0 = c_0, x_1 = c_1, \dots, x_n = c_n)$$

$$\approx P(B_0|x_0 = c_0) \prod_{i=1}^{n} P(x_0 = c_0, x_i = c_i)\,. \tag{3.1}$$

When estimating the background probabilities from above equation, we face two problems. The first one is the estimation and updating of the probability distributions $P(B_0|x_0)$, $P(x_0, x_i)$, and $P(x_i)$. The distribution $P(B_0|x_0)$, a pixel-wise background color distribution, is regarded as an MoG and can be calculated from Eq.(6.2), whose parameters are estimated and updated by using Eqs. (2.3), (2.4) and(2.5). It is tedious to estimate and update the bivariate probability distribution $P(x_0, x_i)$, since the number of possible color combinations in x_0 and x_i is large. We will simplify the estimation and updating by combining the MoGs of pixels to form the joint random vector distributions of pixel-pairs. The second problem is the cost of modeling pixel-pairs. To model all pixel-pairs, the number of pixel-pairs is $O(W^2 \times H^2)$, where W and H are the width and height of the images, respectively. We reduce the complexity by only modeling the pixel-pairs that can provide sufficient information to represent spatial relations as described in Sec. 3.2.



(a)



(b)

Fig. 3.1 Color samples of three pixels in 1000 frames. (a) A sample image and the colors in three pixels in a time period. (b) Scatter plots of the pixels in the spaces ($r(x_2)$, $r(x_3)$) and ($r(x_2)$, $r(x_1)$), and probability distributions of $r(x_1)$, $r(x_2)$, and $r(x_3)$.

3.1.3 Estimation of Bivariate Color Distributions

As mentioned before, the color distributions of pixel-pairs should be updated to adapt to the background changes. If we assume the color distributions in a pixel-pair

(a_1, a_2) to be independent, the joint probability $P(x_1, x_2)$ can be regarded as $P(x_1) \cdot P(x_2)$. Assuming the color distributions are a mixture of Gaussians, the background colors of the two pixels a_1 and a_2 form several *background constituents*, which can be represented as Gaussian distributions $G_1 = \{\eta(\mu_{k_1}, \Sigma_{k_1}) | 1 \le k_1 \le K_1\}$ and $G_2 = \{\eta(\mu_{k_2}, \Sigma_{k_2}) | 1 \le k_2 \le K_2\}$, respectively. The weights in both distributions are denoted as $W_1 = \{w_{k_1} | 1 \le k_1 \le K_1\}$ and $W_2 = \{w_{k_2} | 1 \le k_2 \le K_2\}$. When the independence is satisfied, the joint color of the pixel pair (a_1, a_2) forms $K_1 \times K_2$ *background joint constituents*, and the joint color distributions of the constituents are combinations of G_1 and G_2, denoted as G=$\{ \eta(\mu_{k_1,k_2}, \Sigma_{k_1,k_2}) | 1 \le k_1 \le K_1, 1 \le k_2 \le K_2, \ \mu_{k_1,k_2} = (\mu_{k_1}, \mu_{k_2}), \ \Sigma_{k_1,k_2}$ is the covatiance matrix$\}$. The weights of the joint constituents are $W = \{w_{k_1} \cdot w_{k_2} | 1 \le k_1 \le K_1, 1 \le k_2 \le K_2\}$. Since the parameters of G_1 and G_2 can be estimated from Eqs. (2.3), (2.4) and (2.5), the parameters (G, W) of the bi-variate MoG $P(x_1, x_2)$ can be calculated easily.

In our background model, since the dependence between the colors of two pixels is used to model the spatial relations, the colors cannot be assumed independent. To estimate the parameter of a bi-variate MoG $P(x_1, x_2)$, we first examine the example depicted in Fig. 3.1. This figure shows the colors of three pixels a_1, a_2, and a_3 collected from 1000 consecutive images, where a_1 and a_2 belong to the same object but a_3 does not. The right-hand side image of Fig. 3.1 (a) shows the histograms of the colors in a_1, a_2, and a_3 in a time period of the sample image in the left. From the histograms, we observe that the colors of a_1 and a_2 usually change simultaneously and their values are dependent. The two scatter plots of Fig. 3.1 (b) from top to bottom are the scatters of $(r(x_2), r(x_3))$ and $(r(x_2), r(x_1))$, where $r(x_i)$ denote the red values of the color random variable x_i. The projection profiles from the top to

bottom are the probability distributions of $r(x_3)$, $r(x_1)$, and $r(x_2)$. Each probability distribution forms several clusters and each cluster is regarded as a background constituent. As shown in the scatter plots, several combinations of background constituents in a pixel-pair form joint background constituents. When the probability distributions of a pixel-pair (a_1, a_2) are regarded as bi-variate MoGs, the probability function $P(x_1, x_2)$ is formulated as follows:

$$P(x_1 = c_1, x_2 = c_2) = \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} w_{k_1,k_2} \cdot \eta\left(c_{1,2}, \mu_{k_1,k_2}, \Sigma_{k_1,k_2}\right). \quad (3.2)$$

In the equation, the color vector $c_{1,2}$ is the joint color vector of colors c_1 and c_2, and the mean μ_{k_1,k_2} is the vector $[\mu_{k_1}, \mu_{k_2}]$, where μ_{k_1} and μ_{k_2} can be estimated from the background updating in Eq.(2.3). The covariance matrix Σ_{k_1,k_2} is estimated with respect to the mean μ_{k_1,k_2} as

$$\Sigma_{k_1,k_2}^{t}\left(1 - \alpha_{k_1,k_2}\left(c_{1,2}^{t}\right)\right)\Sigma_{k_1,k_2}^{(t-1)} + \alpha_{k_1,k_2}\left(c_{1,2}^{t}\right)\left(c_{1,2}^{t} - \mu_{k_1,k_2}^{t}\right)\left(c_{1,2}^{t} - \mu_{k_1,k_2}^{t}\right)^{T}, \quad (3.3)$$

where the $c_{1,2}^{t}$ and μ_{k_1,k_2}^{t} are the joint color vector and joint mean vector in the pixel-pair (x_1, x_2), respectively. In the equation, if a joint vector of colors is matched with a joint constituent, the covariance matrix of the joint constituent should be updated as follows:

$$\alpha_{k_1,k_2}\left(c_{1,2}^{t}\right) = \begin{cases} \alpha_c, & \text{if } dist\left(c_{1,2}^{t}, \mu_{k_1,k_2}^{t}, \Sigma_{k_1,k_2}^{t}\right) < Th \text{ and} \\ & (k_1, k_2) = \arg\min\left(dist\left(c_{1,2}^{t}, \mu_{k_1,k_2}^{t}, \Sigma_{k_1,k_2}^{t}\right)\right), \\ 0, & \text{otherwise} \end{cases} \quad (3.4)$$

where α_c is a constant to control the updating rate, and $dist(c_{1,2}^{t}, \mu_{k_1,k_2}^{t}, \Sigma_{k_1,k_2}^{t})$ is a distance function between the joint color vector $c_{1,2}^{t}$ and joint mean vector μ_{k_1,k_2}^{t}. The process of determining the minimal distance $dist\left(c_{1,2}^{t}, \mu_{k_1,k_2}^{t}, \Sigma_{k_1,k_2}^{t}\right)$ for all

pairs of (k_1, k_2) is termed a *matching process*. In our experiments, the Mahalanobis distance is selected as the distance function. If a joint color vector $c_{1,2}^t$ does not match with any Gaussian distribution, a new Gaussian distribution is created and its mean is set as $c_{1,2}^t$. The weight of the new bi-variate distribution is initialized to zero.

The weight of a joint constituent in a pixel-pair is measured as the frequency of colors in the pixel-pair in past frames matched with the joint Gaussian distribution of the constituent, similar to Eq.(2.5). The updating rule of the weights is defined as

$$w_{k_1,k_2}^t = \left(1 - \beta_{k_1,k_2}\left(c_{1,2}^t\right)\right) w_{k_1,k_2}^{(t-1)} + \beta_{k_1,k_2}(c_{1,2}^t), \tag{3.5}$$

$$\beta_{k_1,k_2}\left(c_{1,2}^t\right) = \beta_c \eta\left(c_1, \mu_{k_1}, \Sigma_{k_1}\right)\eta\left(c_2, \mu_{k_2}, \Sigma_{k_2}\right), \tag{3.6}$$

where β_c is a constant used to control the updating speed. Thus far, all the parameters used for estimating the joint color probability in Eq. (3.2) are ready and the background probabilities of Eq. (3.1) can be estimated from a set of color joint probabilities in a set of pixel-pairs.

Note that, during the background model estimate on, the weight w_{k_1,k_2} is not set as the product of w_{k_1} and w_{k_2}. In other words, the constituents in the two pixels are not independent, and their relations are represented by the weights of the joint constituents. The relations in our model can be used to improve the accuracy of foreground detection. For example, in Fig. 3.1(b), the weight of joint constituents in R_4 is approximately zero, since no pixels match with the constituents; that is, the two constituents belonging to pixels x_1 and x_2 in R_4 usually do not appear simultaneously. However, when the joint colors in pixel-pair (x_1, x_2) match one of the joint constituents in R_4, the pixel-pair (x_1, x_2) is classified as foreground.

3.2 Saptially-Dependent Pixel-Pairs Selection

The joint colors in pixel-pairs are used to represent the spatial relations of a background model. In a scene, not all pixel-pairs contain sufficient spatial relations. Modeling the unrelated pixel-pairs is useless for foreground detection. To reduce the computation cost, we will find the pixel-pairs with higher dependence.

The colors of two pixels with high dependence will form compact clusters in the scatter plots as shown in Fig. 3.1(b). The compactness of a bi-variate distribution is measured from mutual information [50]. The *mutual information* $I(x_i, x_j)$ for colors c_i and c_j is defined as

$$I(x_i, x_j) = \sum_{\substack{(\text{all } c_i \text{ in } x_i) \\ (\text{all } c_j \text{ in } x_j)}} P(c_i, c_j) \log \left(\frac{P(c_i, c_j)}{P(c_i) P(c_j)} \right). \tag{3.7}$$

Here, $P(c_i)$,$P(c_j)$, and $P(c_i, c_j)$ can be computed from Eqs. (2.1) and (3.2). To reduce the cost of calculating the probabilities for all possible colors, the probability $P(c_i, c_j)$ can be replaced by the weights estimated from Eq. (3.5). The mutual information $I(x_i, x_j)$ in Eq. (6.2) can thus be reformulated as follows:

$$I(x_i, x_j) \approx \sum_{m=1}^{K_i} \sum_{n=1}^{K_j} \left(w_{m,n} \log \left(\frac{w_{m,n} \sum_{m''=1}^{K_i} \sum_{n''=1}^{K_j} w_{m'',n''}}{\sum_{n'=1}^{K_j} w_{m,n'} \sum_{m'=1}^{K_i} w_{m',n}} \right) \right). \tag{3.8}$$

The pixel pair (x_i, x_j) with higher mutual information $I(x_i, x_j)$ is selected to model spatial relations of the background model.

4. A Particle Filter with Discriminability Improved Histogram Model

A tracking algorithm is usually composed of two procedures: prediction and update. In the prediction procedure, the system dynamic model of the target is used to predict the current state of the target from previous states. In the update procedure, current observations are used to adjust the predicted state of the target. Particle filters are used to track the state of a target object approximated by a set of discrete samples with associated weights. In our research, we adopt the color-based particle filter proposed by Nummiar et al. [19] to track the targets. In the following, we will explain the algorithm of the particle filter and our modifications.

4.1 Particle Filter

In a particle filter, a target object is tracked by a set of weighted sample states (particles). In the prediction procedure, the samples are propagated into the next step according to the system dynamic model. The update procedure can be divided into two steps: particle weighting and particle selection. In the first step, the weight of a sample is calculated according to the target model, which models the observations of a target object and can be used to calculate the probability of a sample belonging to the target. In the second step, the Monte-Carlo method is used to re-sample the particles.

4.1.1 Prediction

When a target in consecutive images is tracked, the state parameters are usually

defined by the position, size, and motion of the target. In this application, the target objects are represented by several bounding rectangles. Each state is described as a vector $S = [X, Y, W, H, \dot{X}, \dot{Y}]^T$, where (X, Y) represents the center of the rectangle, (W, H) the size of the rectangle, and $(\dot{X}, \dot{Y})$ the velocity of the center.

In the initialization stage, all particles tracking the same body part are assigned the same rectangle position and size, but different velocities. The position and size of a target are determined by the human segmentation process, to be described in Sec. 5.1 and Sec. 5.2. The initial velocities of each particle are randomly selected, because we do not know where the target is moving toward and how fast it moves in the human segmentation step.

If the target moves smoothly in a scene, the system dynamic model can be defined as a motion with a constant velocity in a short time period. The model is defined as:

$$S_t = AS_{t-1} + W, \tag{4.1}$$

where A defines the deterministic component of the model, and W the noise. In general, the velocity and size of a target object do not vary greatly between two consecutive images. Therefore, in the system dynamic model, the size and velocity of the target object modeled by the deterministic component A are fixed. In the tracker, the slowly changed velocity and size can be adjusted by the noise part W, which is defined as a Gaussian vector. In this study, we define formally the matrix A and vector W as:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2}$$

$$W = [\xi_X \quad \xi_Y \quad \xi_W \quad \xi_H \quad \xi_{\dot{X}} \quad \xi_{\dot{Y}}]^T. \tag{4.3}$$

The component ξ_- of the noise vector W is a zero-mean Gaussian random variable $N(0, \sigma_-^2)$. The variances $\{\sigma_X^2, \sigma_Y^2, \sigma_W^2, \sigma_H^2, \sigma_{\dot{X}}^2, \sigma_{\dot{Y}}^2\}$ of the components are set to {10, 10, 2, 2, 2, 2}, according to our experimental results.

4.1.2 Particle Weighting

In the update procedure, we will convert the state of each particle into feature values. Then the feature values will be compared with those of the target object to calculate the similarity π between them as the weight of the particle.

Each particle is composed of a state vector and a weight. The set of particles is defined as:

$$S = \{S^{(n)}, \pi^{(n)} | n = 1 \dots N\}. \tag{4.4}$$

According to these weights $\{\pi^{(n)}\}$, the estimated state of the target object can be determined from the expectation of $\{\mathrm{S}^{(\mathrm{n})}\}$ at each time step, that is,

$$E(S) = \sum_{n=1}^{N} \pi^{(n)} S^{(n)}. \tag{4.5}$$

The weight of a particle in state $S_t^{(n)}$ is computed as:

$$\pi^{(n)} = w \cdot p\left(z_t^{(n)} \middle| X_t = S_t^{(n)}\right). \tag{4.6}$$

where $z_t^{(n)}$ denotes the appearance feature vector of the *n*-th particle at time *t*, and *w* a normalization factor,

$$w = \frac{1}{\sum_{n=1}^{N} p\left(z_t \middle| X_t = S_t^{(n)}\right)}, \tag{4.7}$$

which ensures that $\sum_{n=1}^{N} \pi^{(n)} = 1$. The detail of the feature extraction and similarity measurement will be described in Sec. 4.2.

4.1.3 Particle Selection

After we track the target for several frames, the weights may concentrate on a small number of particles. In the extreme case, the weight of a single particle may approximate to one and the others to zero. In that case, the particle filter will only be related to the system dynamic model. The particles should be resampled when the weights are concentrated on a small number of particles. The sequential importance sampling (SIS) algorithm draws new particles $S_t^{(n)}$ at time *t* from the particles $S_{t-1}^{(n)}$ at time *t*-1 according to $S_t^{(n)} = S_{t-1}^{(h(n))}$. The function $h(n)$ maps the selected particles. To create the mapping, we first create an accumulated histogram of the weights of old particles as follows:

$$\begin{cases} c^{(n)} = c^{(n-1)} + \pi^{(n)} & 1 < n \le N+1 \\ c^{(1)} = 0 \end{cases}. \tag{4.8}$$

Then we generate *N* uniformly distributed random numbers $\{u^{(n)} | 1 \le n \le N\}$. The mapping $h(i)$ is then defined as

$$h(n) = m \qquad \text{when } c^{(m)} \le u^{(n)} < c^{(m+1)}. \tag{4.9}$$

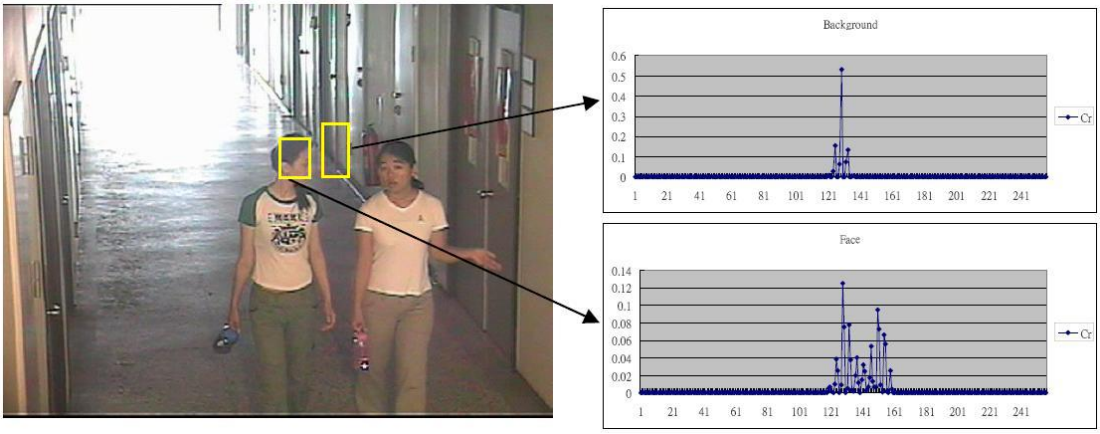


Fig. 4.2 A sample image and the histogram of the Cr channel in the two rectangles (head and background).

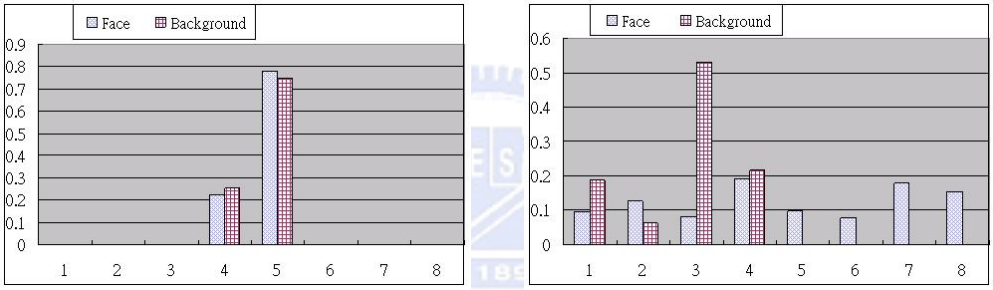


Fig. 4.2 The two histograms of the Cr channel in Fig. 4.2 quantized into eight bins. (a) Uniform mapping. (b) Equalized mapping.

4.2 Target Object Similarity

To calculate the weight of each particle, the probability of a particle belonging to the tracking target should be calculated. Considering the captured video of a fixed camera, the tracking targets are usually the moving objects in the scene. If only a target object moves in the scene, we can model the background image and extract the target object by subtracting the background image from the current frame. However,

when multiple target objects are tracked, background subtraction is not enough to distinguish them. To distinguish different targets, Pérez et al. [18] introduced a target appearance model using color histograms, since the color histogram is robust against non-rigidity. When the target is moving, his appearance may change due to the variations of poses or illumination. If the target appearance changes, the appearance may fail to detect the target, but the background model can be used. Therefore, in our study, we integrate the similarities of a particle from both background model and target appearance model to form a robust tracking system.

4.2.1 Specific Histogram Mapping

In our application, we aim to track a human in consecutive color images. The color histogram model [19] is robust against partial occlusion, non-rigidity, and rotation. However, in our application, the region of a tracking target may be small. To track the object in small regions, the histogram may be sparse and not sufficient to represent the color distribution of the region. For instance, if the number of bins is set as $8 \times 8 \times 8$ and the region in image is 32×32, the expected number of pixels in each bin is only two, which is insufficient to represent the color distribution. To represent the color distribution, we model the histogram in color channel independently. Here, we select YCbCr as the color space, since the three channels are assumed independent. We divide the values in each channel into eight bins respectively. The expected number of pixels in each bin is 128, which can represent the color distribution more sufficiently. Another benefit of the modification is the computational efficiency when we compare the histograms between a particle and the target object, because the total number of bins is reduced to 24.

To represent the color histogram in several bins, another important task is how to map from a range of colors in the histogram to a bin. If the range is equally quantized

for each bin and the histogram is compact, all pixels may fall into a small number of bins. In our cases, two different histograms cannot easily be distinguished. Fig. 4.2 shows two histograms of the Cr channel in the face region of a person and a background region, whose histograms are very different. When the ranges are equally divided into eight bins as shown in Fig. 4.2, the color distributions of the two regions will be very similar. To cope with the problem, we first choose one histogram H as the reference one for histogram equalization. The equalization can be denoted as $z = M(H)$, where $M(.)$ is a function that equalizes the reference histogram H into an equalized histogram z, which is represented as a vector. The function $M(.)$ is then applied to another histogram H' to form a feature vector $z = M(H')$. Based on the mapping, we can prevent the pixels from falling into the same bins for two slightly different color distributions. Fig. 4.2 shows the quantized bins of the face region and background regions by selecting the face region as a reference one. In the figure, we can easily find that the two quantized histograms are different, especially in the third bin.

4.2.2 Target Appearance Model

We model the histogram in each of the three color channels in the color space YCbCr, since the three channels are assumed independent. We divide the values in each channel into eight bins. In the initialization phrase, the color histogram is extracted from the image of the target object. Since the target object is moving, its appearance may change gradually. To adapt to the changes, the histogram model is updated as follows:

$$H_{t+1} = H_t \cdot (1 - \alpha) + Q_t \cdot \alpha, \tag{4.10}$$

where H_{t+1} and H_t are the histogram models at time t+1 and t, Q_t is the histogram

directly extracted from the estimated state at time *t*, and α is a constant used to control the updating speed. In a frame, the region of a particle state whose color histogram is similar to that of the target object should have a higher probability belonging to the target object.

4.2.3 Background Appearance Model

To check whether a particle state is located in the position of a moving object, we also check the differences between the current frame and the background scene. Here, we adopt a Gaussian background model [3] to extract the background image. In general, the background appearances may change due to background object moving or illumination change. To adapt to the change, the background model is updated as

$$B_{t+1}(x,y) = B_t(x,y) \cdot (1-\beta) + I_t(x,y) \cdot \beta, \tag{4.11}$$

where $B_t(x,y)$ and $I_t(x,y)$ are the color vectors in pixel (x,y) of the background image and frame image in time *t* respectively. To detect the foreground object, the currently processed image can subtract with the background image. However, the pixel-wise background subtraction is sensitive to background variations such as variation of illumination or vibration of leaves. Since the background variation is not greatly affect the color distributions in a region, we extract the color histograms in the positions of particle states as the background features. In a frame, the region of a particle state whose color histogram is similar to that of the background image should have a lower probability belonging to the target object.

4.2.4 Similarity Measurement

The appearance of the particle that belonging to target object should be similar to target appearance model but different from the background appearance. As described

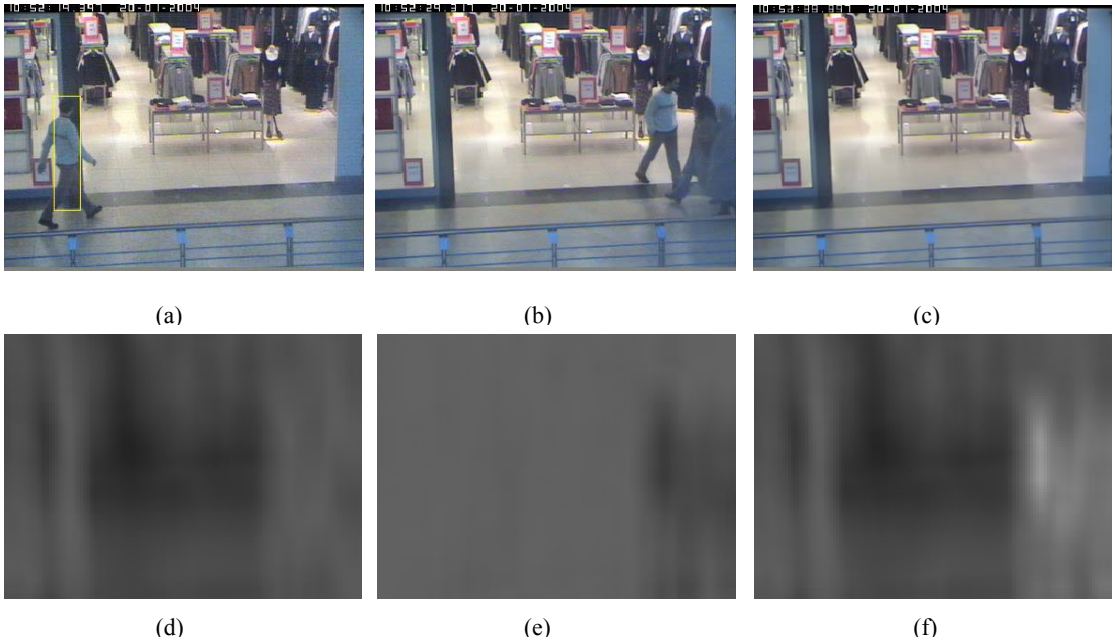


Fig. 4.3 An example of the similarity measurement by using the color histograms of target person and background image. (a) Image of a tracking target person, (b) Image of a tracking frame, (c) Background image, (d) Target appearance similarity map of the tracking frame, (e) Background appearance similarity map of the tracking frame, (f) The similarity map by combining background and target appearance models.

above, we can model the target color histogram and map it into *N* bins according to the mapping function $M(.)$ defined in Sec. 4.2.1, labeled as x_t. We can also extract the background color histogram from the adaptive background model in the region of a particle and map it into *N* bins, labeled as $b_t^{(n)}$. To measure the probability of a particle $S^{(n)}$ belonging to the target object, we extract the color histograms of particle from current processed image and map it into *N* bins, labeled as $z_t^{(n)}$. A particle state $S^{(n)}$ with a higher probability belonging to the target object has the property that the distance between x_t and $z_t^{(n)}$ must be small, and between $b_t^{(n)}$ and $z_t^{(n)}$ must be high. Therefore, the probability used in Eq. (4.6) can be formulated as

$$p\left(z_t^{(n)} \middle| X_t = S_t^{(n)}\right) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{\left(-\frac{1}{2}D\left(z_t^{(n)}, x_t, b_t^{(n)}\right)^2\right)}, \tag{4.12}$$

$$D\left(z_t^{(n)}, x_t, b_t^{(n)}\right) = \frac{\left\| z_t^{(n)} - x_t \right\|}{\left\| z_t^{(n)} - b_t^{(n)} \right\|}. \tag{4.13}$$

Fig. 4.3(d) shows the similarity map of Fig. 4.3(b) comparing with the target appearance model extracted from the person region of Fig. 4.3(a). Fig. 4.3(e) shows the similarity map of Fig. 4.3(b) comparing with the background appearance model extracted from Fig. 4.3(c). Fig. 4.3(f) shows the combines of the two similarity according to Eqs. (4.12) and (6.2). In the similarity map, the gray scale of a pixel represents the similarity of a region with the same size of the person region in Fig. 4.3(a) centered in the pixel. In Fig. 4.3(f), we can observe that the region of the target person in Fig. 4.3(b) has largest similarity.

Note that, in a particle filter, the measurement affects the selection of particles in the resampling step. The motion of the selected particles in the next frame is determined by the system dynamic model defined in Eq. (4.1). The noise *W* in the system dynamic model affects the distribution of the particles. Since we assume that the noise *W* is a Gaussian distribution, the distribution of particles forms a mixture of Gaussians in the state space. The set of Gaussian distributed particles will be placed in each important part of the state space, which will next be resampled according to the weights calculated from the measurements. All the selection and generation of particles are the characteristics of the particle filter.

4.3 Cumulative Histogram Map

When we track a target object in images, a set of particles should be used in the

tracker and the color histograms of the current frame and background image in the positions of these particles must be extracted. It is time-consuming to extract a large number of color histograms in an image. Since most of the regions of these particles are overlapped, many redundant calculations are spent for estimating the colors in the pixels of the overlapped regions. To reduce the redundancy, we create a cumulative histogram map (CHM) for the processing frame and background image. Then we can extract the color histogram feature for each particle in a constant time.

The CHM is similar to the integral map popularly used for extracting Haar-Like feature [20]. The map is created to cumulate the color histograms. In a region $R = \{(x, y)|x_1 < x \leq x_2, y_1 < y \leq y_2\}$, the color histogram of a color channel can be calculated as

$$h_R(i) = \sum_{(x,y)\in R} \sum_{u\in M_i} \delta[I(x, y) - u], \tag{4.14}$$

where $\delta[.]$ is the Kronecker delta function, and M_i is the set of colors that map into i-th bin. According to the equation, we can define the CHM for an image as

$$\mathrm{CHM}(x, y, i) = \sum_{x'\leq x, y'\leq y} \sum_{u\in M_i} \delta[I(x', y') - u]. \tag{4.15}$$

The CHM can be calculated recursively as

$$\begin{aligned} \mathrm{CHM}(x, y, i) = {} & \mathrm{CHM}(x, y - 1, i) + \mathrm{CHM}(x - 1, y, i) \\ & -\mathrm{CHM}(x - 1, y - 1, i) + \sum_{u\in M_i} \delta[I(x, y) - u]. \end{aligned} \tag{4.16}$$

When we obtain the CHMs for an image, the histogram from Eq. (4.14) can be calculated as

$$h_R(i) = \mathrm{CHM}(x_2, y_2, i) + \mathrm{CHM}(x_1, y_1, i) - \mathrm{CHM}(x_1, y_2, i) - \mathrm{CHM}(x_2, y_1, i). \tag{4.17}$$

Based on the modification, the time complexity for creating M CHMs for an image is $O(W \times H \times M)$, where W and H are the width and height of an image respectively and M is the number of histogram bins. The time complexity for extracting color histograms in a region is $O(M)$. The complexity for extracting the color histograms for N particles in an image is thus $O(M \times N + W \times H \times M)$. The time complexity of histograms extraction in Eq. (4.14) for all particles in an image is $O(W' \times H' \times N)$, where W', $H'$are the width and height of the region R. The number of bins is much smaller than $W' \times H'$. If $W \times H \times M$ is smaller than $W' \times H' \times N$, the speed can be improved. According to our test results, when more particles are used, the target can be traced more precisely. In our experiments, we set the number of particles around 1000. Therefore, the speed of the particle filter-based tracker can be improved. The measurement of speed improvement will be described in the experiment result section.

4.4 Dynamic Number of Particles Adjustment

The number of particles will greatly affect the search region of the target object in an image. In general, when more particles are selected, the tracker may become less efficient but more accurate. Recently, Fox [51] proposed a method to dynamically adjust the number of particles by using the KL-distance to reduce the error. The method adjusts the number of particles to minimize the error between the true posterior and the sample-based approximation. However, the computation is costly. In a particle filter, the particles with local maximum weights are much important for the target state estimation and particle resampling. Therefore, we modify the number of particles to control the covering range in state space so that the state with the local maximum weight can be located.

When we track a target with the particle filter, if the appearances of many regions are similar to the target object, the weights of particles will approximate a uniform distribution and have a higher entropy. If the target has been missed, the conditional probability $p(z_t|X_t = S_t^n)$ will be low and the distribution of weights will also approximate a uniform distribution. In these two cases, since we do not know where the target object is, a wide search window should be set. Therefore, a larger number of particles are needed. In another case, if there is only one region whose appearance is similar to the target object, the weights will concentrate on few particles and have a lower entropy. In that case, a small search window is needed and a smaller number of particles is required.

To address the cases described above, we define the number of particles at time *t* (or the *t*-th frame) based on the entropy as

$$N_t = C \cdot N_{t-1} \cdot \frac{-\sum_{n=1}^{N_{t-1}} \pi^{(n)} \log \pi^{(n)}}{-\log(1/N_{t-1})}. \tag{4.18}$$

where *C* is a constant to control the increase rate of the number of particles. For example, if *C* is set as two, the maximum number of particles at time *t* is $2 \cdot N_{t-1}$. In our experiments, the constant *C* is 1.2. To avoid the number of particles increasing or decreasing drastically, we limit the number N_t between (200, 1000) in our experiments.

5. Three-Part Human Tracking and Consistency Checking

To track a human reliably, the three parts, head, torso, and hip-leg, are tracked simultaneously, as shown in Fig. 1.2. To design the tracking system, we first segment a person in a frame via the background subtraction method, and decompose the person into three parts. The positions of each part in the following frames are then predicted and updated using the particle filter described in the previous section. For each frame, after the positions of the three body parts are estimated by particle filters, consistency checking and adjustment of these body parts are performed to correct the abnormal body part. Finally, we perform an inter-person occlusion detection to avoid losing the target person when the person is occluded by other persons.

5.1 Human Extraction

To segment the human from an image, we first apply the method described in chapter 3. Next, we extract the connected-components of foreground pixels as foreground regions. The connected-components smaller than a prespecified threshold are regarded as noise and removed. The extracted foreground regions may include shadows, other background objects being moved or due to illumination changes. A foreground region may also include more than one person. To extract the persons, we restrict the size of a human region by setting two thresholds, region size T_{fg} and width to height aspect ratio T_{wh}. If a foreground region does not satisfy the criteria, we will separate the foreground region into two sub-regions by finding the lowest

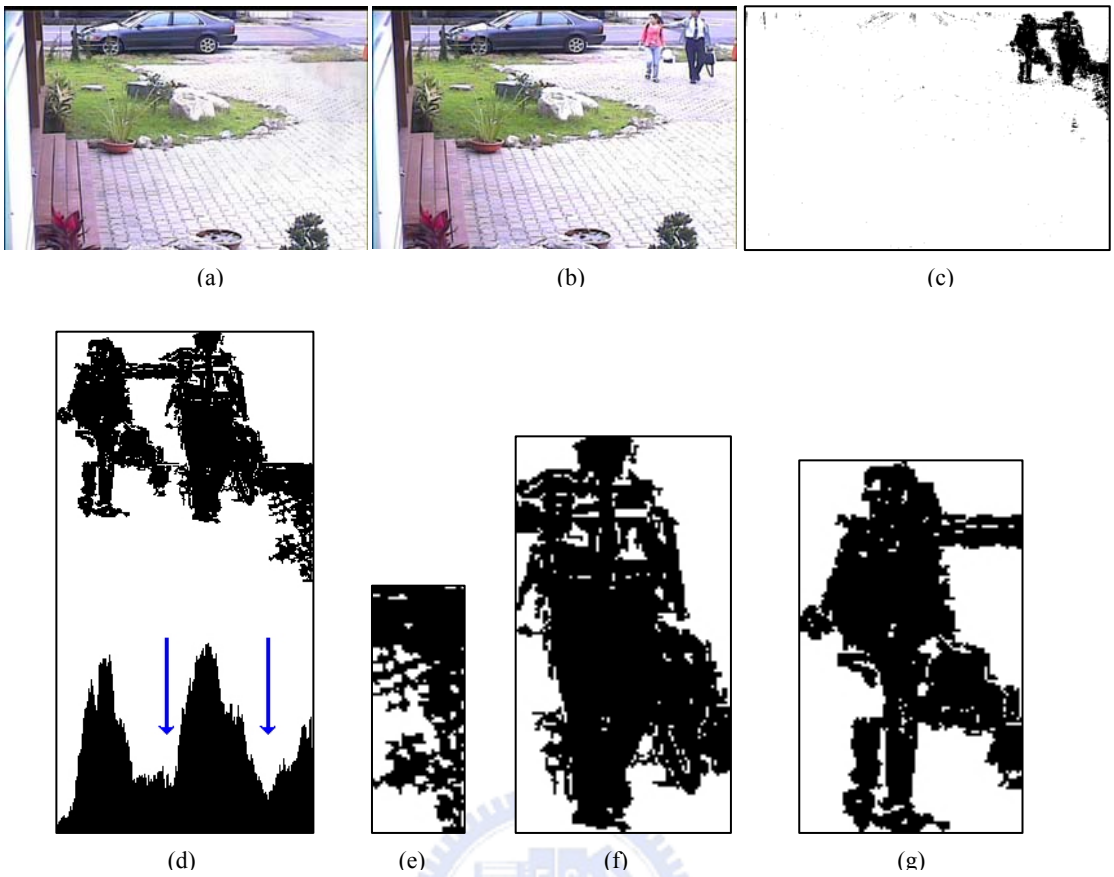


Fig. 5.1 Foreground subtraction images. (a) The current frame. (b) The mean colors of the Gaussian background model. (c) Foreground regions. (d) The foreground connected-component and its vertical projection profile. (e)(f)(g) Sub-regions after recursive separation.

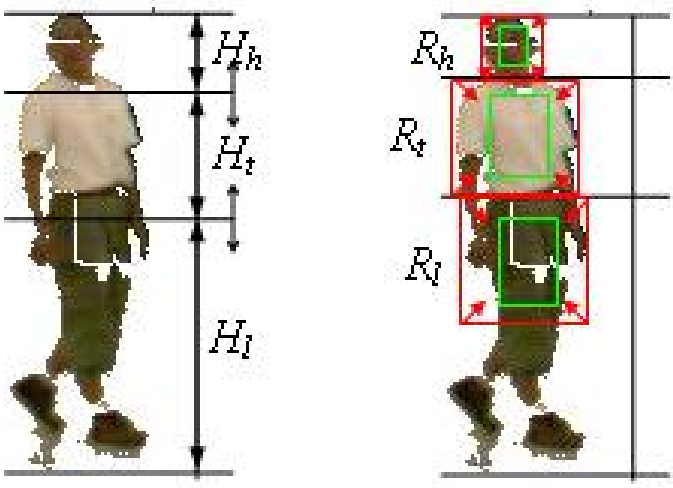
valley in the vertical projection profile [2]. After the separation, if the aspect ratio of a sub-region is still larger than T_{wh}, the separation rule will be performed recursively until all sub-regions satisfy the criterion of aspect ratio. Note that the two thresholds should be trained for different camera settings. In our experiments, the threshold T_{fg} was set as 2000 pixels and T_{wh} as 1.0. Also note that the region of a person may touch with small misclassified background regions, such as shadows. Since we do not use the whole image of the person for tracking, the small misclassified background region will not affect greatly the tracking performance. Even though a detected body

part of the person belongs completely to a misclassified background region, the part can also be adjusted by using the other two parts during the process of tracking failure adjustment described in Sec 5.4.

As an illustration, Fig. 5.1(a) shows the current frame, and Fig. 5.1(b) the mean colors of the Gaussian background model. To perform human segmentation, we extract foreground regions by the background subtraction method. After the image is thresholded, we obtain the foreground image in Fig. 5.1(c). Then we extract connected-components from the foreground images and remove the connected-components smaller than T_{fg} pixels. A large connected-component remains. Since the aspect ratio of the component is greater than the threshold T_{wh}, we separate it into two blocks based on its vertical project profile, as shown in the lower part of Fig. 5.1(d). The smaller block shown in Fig. 5.1(e) is removed because its size is less than T_{fg}. For the larger block, another separation is performed and two sub-blocks shown in Fig. 5.1(f)(g) are obtained. Since the aspect ratio criterion is now satisfied, the recursive separation process stops.

5.2 Human Part Decomposition

Since the three body parts are used to track a person, their appearances should not be confused with each other. In this study, we aim to separate the three parts with a high distinguishability. The distinguishability can be defined as the difference between the color histograms of two regions. Since the difference measurement of color distributions used for our particle filter as depicted in Sec. 4.2.1 is costly, we will compute and compare the mean colors of the three parts instead.



(a) (b)

Fig. 5.2 An example of human parts decomposition of a person. (a) Initial horizontal separation lines of the person. (b) Results of final three parts of the person.

We assume that the size ratios of the three body parts of most people are similar. As shown in Fig. 5.2, we first locate two horizontal lines to separate the region of a person into three sections according to the predefined height ratio of the three parts, denote as H_h, H_t, and H_l. Then we move the two separation lines vertically to find the positions such that the three regions have the highest differences in the mean colors. The foreground human region is accordingly separated by the two horizontal lines into three sections R_h, R_t, and R_l. As described above, the segmented foreground regions may include background regions or noise. Besides, the shapes of the three parts for different persons and different poses are varied. To achieve a higher reliability of the tracked parts, we will shrink the segmented regions according to the spatial distribution of the pixels in the three sections. A section R is shrunk into a smaller rectangle, called inner rectangle hereafter, as the pixel set $\{(x,y)\big|C_x - S_x < x < C_x + S_x, C_y - S_y < y < C_y + S_y, (x,y) \in R\}$, where (C_x, C_y) is the center of the rectangle and (S_x, S_y) the covered range of the rectangle. They are defined as follows:

$$C_x = \frac{1}{|R|} \sum_{(x,y)\in R} x, \tag{5.1}$$

$$C_y = \frac{1}{|R|} \sum_{(x,y)\in R} y, \tag{5.2}$$

$$S_x = K_R \cdot \sqrt{\frac{1}{|R|} \sum_{(x,y)\in R} (x - C_x)^2}\,, \text{and} \tag{5.3}$$

$$S_y = K_R \cdot \sqrt{\frac{1}{|R|} \sum_{(x,y)\in R} \left(y - C_y\right)^2}\,. \tag{5.4}$$

where K_R is used to control the shrinking rate. Our shrunk body parts are less affected by limb motions, and the three parts are usually located in a vertical line.

Fig. 5.2(a) shows the three inner rectangles found for the position in Fig. 5.2(b), in which the colors are more uniform. Note that the height of the inner rectangle of hip-leg is set to the half height of the segmented hip-leg, because the appearances of lower legs may vary significantly for different motions and dresses, which are not stable for tracking.

5.3 Tracking Failure Detection

The relative positions of the three body parts are limited in a certain range and the velocity of each part is also limited. Tracking failure will generate abnormal relative positions of estimated body parts, and the states will change irregularly in recent frames. If we can create a classifier to distinguish normally and abnormally tracked body parts, we can detect the event of tracking failure.

To detect the tracking failure, we also have to detect the failure component. In this study, we use support vector machines (SVM) [22] as classifiers to detect whether

and which body part cannot be tracked properly. The SVM is a well-known classifier that finds a hyperplane in a higher dimensional space to separate data of two categories with the largest margin. The optimal hyperplane is computed as follows:

$$f(x) = sgn\big(g(x)\big), \tag{5.5}$$

where

$$g(x) = (\sum_{i=1}^{l^*} y_i \alpha_i K(x, x_i^*) + b). \tag{5.6}$$

Here, we select the radial basis function as the kernel function *K* to map the feature vectors into the higher dimensional space. The class label $y_i \in \{-1,1\}$ denotes whether the feature vector x_i^* belongs to tracking failure or not. The set $\{x_i^* | 1 \le i \le l^*\}$ is a subset of the training data set, called *support vectors*. The coefficients α_i and *b* are determined by solving a large-scale quadratic programming problem.

To detect which part cannot be tracked, we design three SVMs for detecting the tracking failures of the three body parts. If the tracker fails to track two or three parts, the SVM failure detector for different body parts may become ineffective, since we cannot easily distinguish which part is abnormal by the relative positions. To cope with the problem, we design an additional SVM to determine whether the failure type is a single part failure or a multi-part failure.

The features used in an SVM are the estimated states of the three parts in the current frame and the relative state changes between the current frame at time t_0 and a previous frame at time $t_0 - \Delta t$. Here Δt is selected to make the state changes large enough (In our experiments,$\Delta t = 0.5$ seconds). The feature vector is defined as $[RS_H(t_0), RS_T(t_0), RS_L(t_0), RS_H(t_0) - RS_H(t_0 - \Delta t), RS_T(t_0) - RS_T(t_0 -$

$\Delta t), RS_L(t_0) - RS_L(t_0 - \Delta t)]^T$, where the vectors $\mathrm{RS_H}(t)$, $\mathrm{RS_T}(t)$, and $\mathrm{RS_H}(t)$ denote the relative state vectors of the three body parts in time *t*. The relative state vectors are defined as:

$$RS_{(K)}(t) = S_{(K)}(t) - \frac{\sum_{I=H,T,L} S_{(I)}(t)}{3} \qquad (K = H, T, L), \tag{5.7}$$

where $S_H(t)$, $S_T(t)$ and $S_L(t)$ are the estimated state vectors of head, torso and hip-leg parts.

To collect training samples, we apply particle filters to track the three body parts in several video sequences. We than manually label the training samples from these tracking results for each SVM. In our experiments, the number of training samples for each SVM is 150. The state vectors not covering the target body part are labeled as negative, while those falling inside are labeled as positive. The samples not satisfying these two criteria are eliminated; this ensures that the feature vectors of the two classes are distinguishable. In the tracker, since a misclassified tracking failure may cause error propagation and hard to be adjusted, we prefer a higher true-negative rate. Thus, we adjust the parameters of SVMs to achieve the goal.

5.4 Tracking Failure Adjustment

In case when a tracking failure is detected, we have to adjust the state of the target person. If two or three parts cannot be tracked, we will detect the foreground region around the previous tracked position of the target object, and re-initialize the tracking process. If the state of a single part is abnormal, we will use the other two body parts to adjust the position and size of the abnormal one. To keep the adjusted body part tracked in the following frames, the particle states and the appearance model (color histogram) must also be modified. If the abnormal body part still appears

(a)

(b)

Fig. 5.3 Examples of two types of tracking failure. (a) A person with clothes colors similar to those of background regions. (b) A person occluded by a pillar.

in the image, we can use the adjusted position and size to extract the particle states and the appearance model. However, if the failure is caused by occlusion, the appearances of the target person may not be correctly extracted as shown in Fig. 5.3, and thus the system dynamic model should be used to track the person. In this case, the appearance model is not updated and the process of failure detection and adjustment is not performed either. The method of occlusion detection is discussed below.

5.4.1 Failure from Inter-Person Occlusion

When two persons are both tracked, the occlusion event can easily be detected by checking whether the tracked body parts of the two persons are touching. If the answer is positive, we determine which one of them is occluded. Here, we use the

weighted function in Eq.(4.6) to calculate the feature similarity between the tracking target and the feature kept in the particle filter. When two persons are overlapped, the person with the lower probability is determined as the occluded one.

5.4.2 Failure from Background Object Occlusion

If a person is occluded by a fixed background object such as a pillar or a door, we cannot detect the event since we only track the position of a person but not the background object. To cope with the problem, we can label manually the large and fixed background objects that may occlude moving humans. This is reasonable for a scene monitored by a fixed sensor.

The position of the tracking failure part can be adjusted according to the position of the other two parts. If the tracking failure part is the torso part as shown in Fig. 5.3(a), we adjust the center of the torso to the middle of the other two parts and the size to the average of the other parts as follows:

$$\{X_T, Y_T, W_T, H_T\} = \frac{\{X_H, Y_H, W_H, H_H\} + \{X_L, Y_L, W_L, H_L\}}{2}. \quad (5.8)$$

Since the torso of a person is usually large enough, the adjusted rectangle usually lies inside the torso. Instead, the head of a person is usually smaller than the other two parts. Using similar adjustment method, its inner rectangle may contain background objects. We will use the background model to segment the foreground region above the torso part, and then extract the inner rectangle from the foreground region by the method described in Sec. 5.2. For the hip-leg part, its shape may have great variations. The adjustment method is similar to that of the head part, except that the foreground region is segmented below the torso part.

In our tracker, the moving velocity of the particles will affect the predicted target position in the next frame. If we set the same moving velocity to all particles, the

particle filter may be unable to predict the state in the next frame. For a moving human, the velocities of the three body parts are assumed similar. If the trackers of the other two parts have N_1 and N_2 particles, we set the particle number of the tracking failure part as $N_1 + N_2$. All the particles of the adjusted body part have the same size $\{W, H\}$ and position $\{X, Y\}$, but different velocities $\{\dot{X}, \dot{Y}\}$ from the particles of the other two parts.

6. Experimental Results

In this research, we propose a spatially-extended background model, and a human tracking model. The human tracking model composed an adaptive color-based particle filter using cumulative histogram maps and a three-part consistency checking algorithm. In this chapter, we will verify the performances of these algorithms respectively.

6.1 A Spatially-Extended Background Model

The test video clips used to test the background model are captured in two different sites and by three different cameras. Two cameras are set in the two ends of a corridor (Cam1 and Cam2), and the other one is set in a laboratory (Cam3). The camera Cam1 is a grayscale CCD camera, Cam2 a color CCD camera, and Cam3 a USB-Webcam. The resolution of each video frame is 320 × 240 and the frame rate is 30 fps. The total time of captured video clips is about 97.4 minutes, which include 175394 frames. The clips contain moving humans, moving background objects, and changing illuminations.

In our experiments, we will compare the foreground detection results of three background models: the Gaussian background model (GBM), MoG-based model (MBM) [9], and spatially-extended background model (SBM). In both MBM and SBM, we represent the background color distributions as a mixture of six Gaussians. In our SBM, we model the pixel-pairs with the distance of five pixels, and use the two spatially-dependent pixel-pairs to represent the spatial relations.

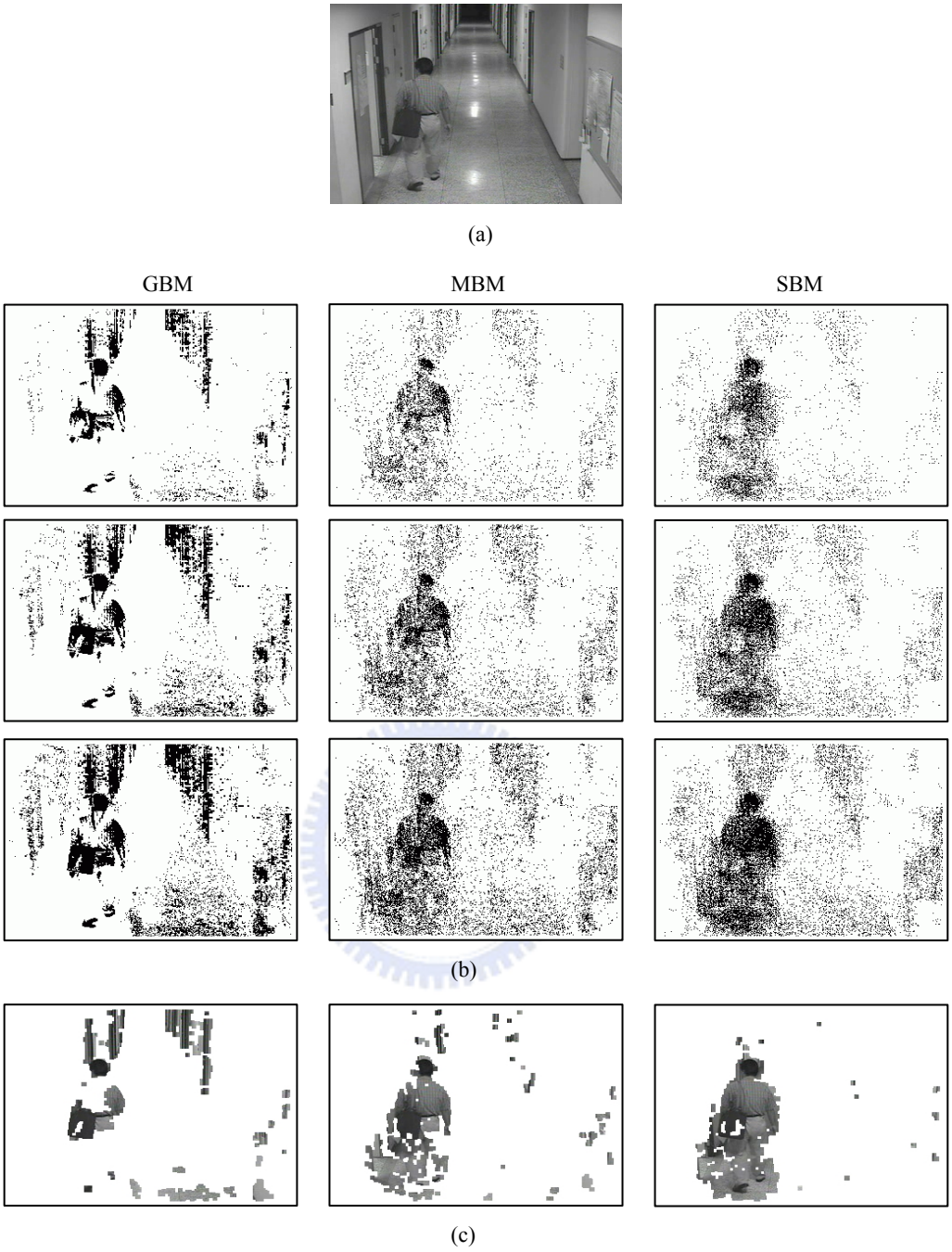


Fig. 6.1 Foreground detection results of an image captured by Cam1. (a) Original image, (b) Detected foreground regions: the images from left to right are the results based on GBM, MBM, and SBM, and from top to bottom are results with 3*N*/40, 5*N*/40, and 7*N*/40 pixels. (*N*=image size), (c) Foreground regions after noise removal.

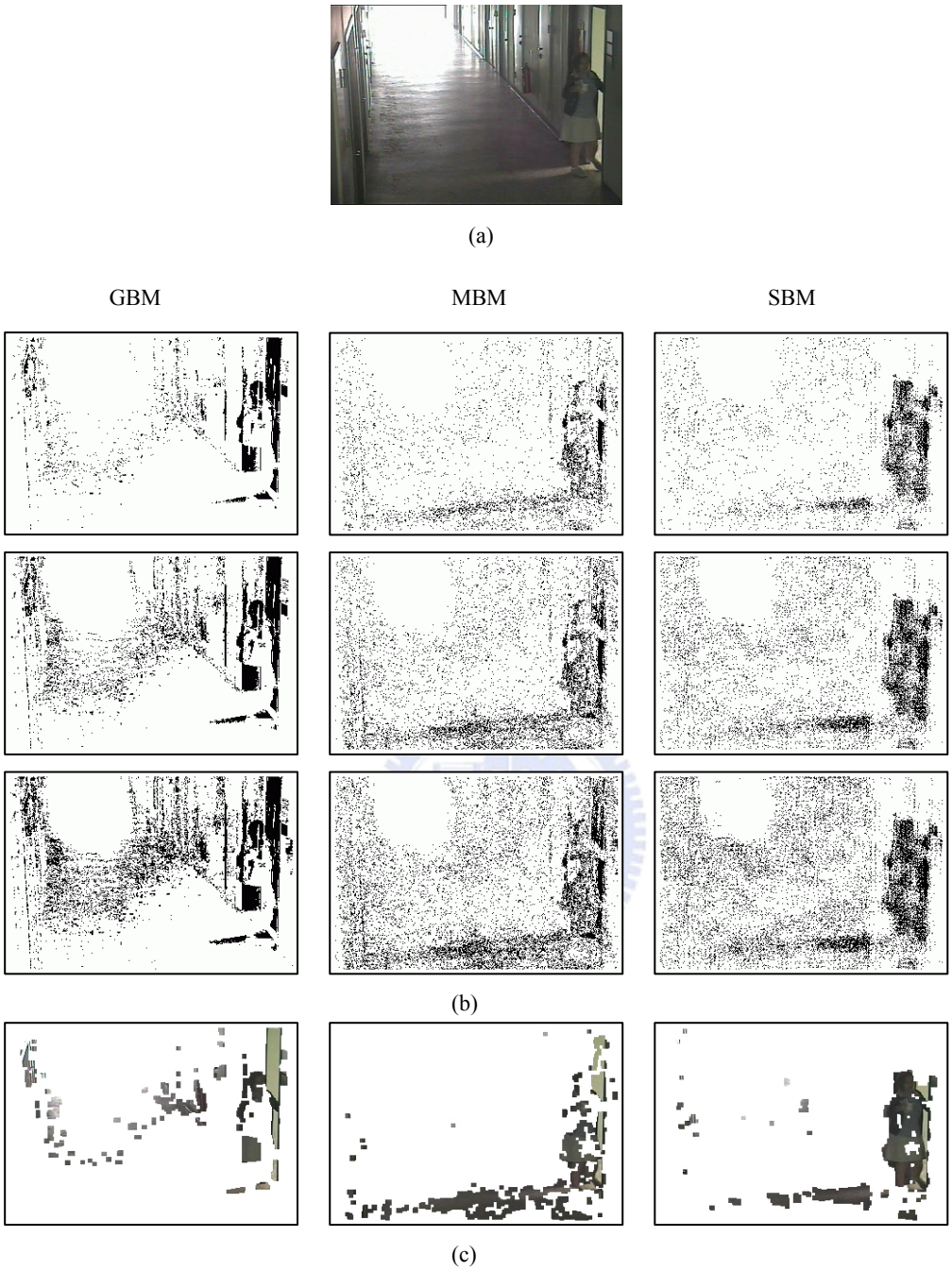


Fig. 6.2 Foreground detection results of an image captured by Cam2. (a) Original image, (b) Detected foreground regions: the images from left to right are the results based on GBM, MBM, and SBM, and from top to bottom are results with 3*N*/40, 5*N*/40, and 7*N*/40 pixels, (c) Foreground regions after noise removal.

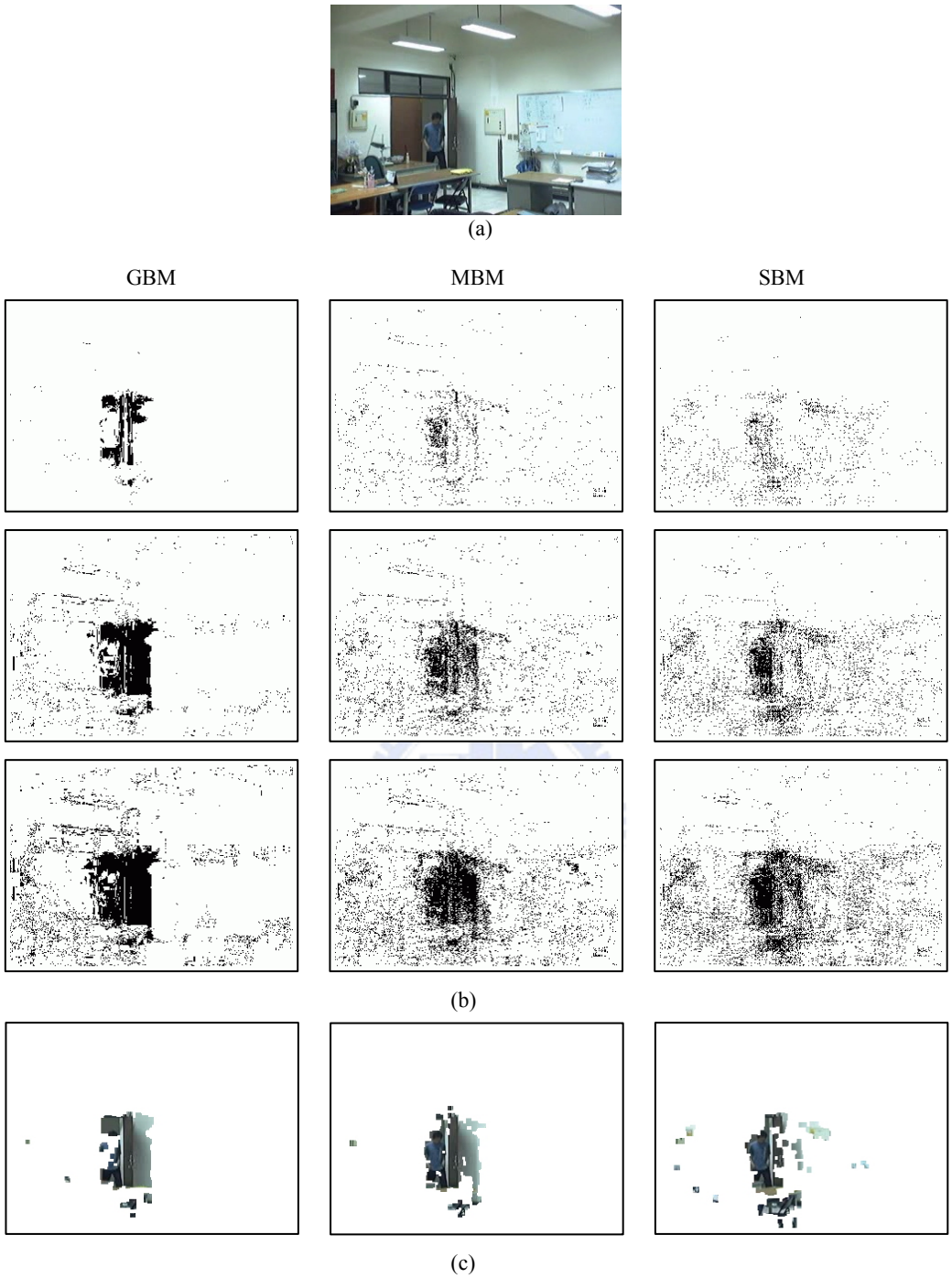


Fig. 6.3 Foreground detection results of an image captured by Cam3. (a) Original image, (b) Detected foreground regions: the images from left to bottom are the results based on GBM, MBM, and SBM, and from top to bottom are results with $N/40$, $3N/40$, and $5N/40$ pixels, (c) Foreground regions after noise removal.

To detect foreground, an effective background model can label most foreground pixels and very few background pixels as the foreground. The number of detected candidate foreground pixels is generally affected by the foreground segmentation threshold used to classify the pixels into foreground or background. Comparing different methods with unsuitable foreground segmentation thresholds cannot reflect the real performances. Here we perform two different kinds of experiments that detect foreground pixels via controlling either detected pixel numbers or foreground segmentation thresholds.

Fig. 6.1 through Fig. 6.3 show result images by controlling detected pixel numbers. Figures Fig. 6.1 through Fig. 6.3(a) show the original images, Fig. 6.1 through Fig. 6.3(b) the detected foreground pixels using different methods, and Fig. 6.1 through Fig. 6.3(c) the foreground regions of the images in the middle row of Fig. 6.1 through Fig. 6.3(b) after morphology-based noise removal. In the noise removal process, if the closing operator is performed before opening, near noises may be merged into a large one and cannot be removed by opening using the same structure element. If the opening is performed before closing, near small holes may not be removed. Therefore we first apply a closing operator with a smaller structure element (3×3) to fill the holes and then apply an opening with a bigger structure element (5×5) to remove noise pixels.

Fig. 6.1(b) shows the results of a sample image captured by Cam1. Since the image is a gray scale one, different objects may easily have similar appearances. The distributions of joint random vectors of pixel-pairs are less efficient to distinguish different objects than those in color images. Thus, the results of SBM and MBM are much similar. After we apply noise removal as shown in Fig. 6.1(c), the regions of the person using SBM are still more complete than those using MBM. The result shows

Fig. 6.4 Foreground detection results of the images captured by the three cameras. The images from top to bottom are original images, the results of GBM, MBM, and SBM.

our proposed SBM is better than the other two methods.

Fig. 6.2(b) shows the results of a sample image captured by Cam2. The captured image is colored, and the colors of many parts of the person are similar to those of the background. The detected foreground regions of GBM and MBM are fragmental. Even though we apply a morphology-based hole filling procedure as shown in Fig. 6.2(c), the foreground regions of the two methods are still fragmental. Thus, we can also conclude that SBM are more efficient than MBM and GBM.

Fig. 6.3(b) shows foreground detection results of a sample image captured by Cam3. Some of the regions of the door and its shadow are misclassified as foreground ones by using GBM and MBM, but not misclassified by using SBM. In the sample, since the door is opened when the person enters the room, the GBM does adapt to the current appearance of the door and its shadow. In MBM, since the color distributions of the person, door and shadow may all be modeled, the regions of these object may be misclassified. As shown in the middle column of Fig. 6.3(b), the regions of the

Fig. 6.5 Test samples and the manually labeled ground truth masks used for estimating the ROC curves.

door and its shadow may also be misclassified or those of the person may be fragmental when an unsuitable segmentation threshold is set. By adopting SBM, the appearances of the person are not taken as background, since the joint colors of a pixel-pair in the person are not captured repeatedly in the same position. The results in Fig. 6.3(c) show that the person regions do not touch with background ones and less background regions are misclassified as foreground.

Fig. 6.4 shows the foreground detection results of the images captured by the three cameras by setting a fixed threshold. The threshold that results in 15% false positive rate in training images is selected to test the performance of the models. The foreground regions depicted are noise removed. The results show that the foreground regions extracted by SBM are more complete, and the false positive regions are less than those of the other two methods. The persons in Fig. 6.4(a), (c) and (f) walk around a place. Since the appearances of the persons are repeated in similar locations, the colors of the persons will be learnt as background by the pixel-wise background models GBM and MBM. In SBM, the colors of pixel-pairs will be modeled and the pixel-pairs without higher spatial dependency will be eliminated. Even though the appearances of a person are similar in a specific location, the joint colors of a pixel-pair in a fixed distance are usually varied and have low probabilities to be

labeled erroneously as background. Note that the illuminations in these scenes are dramatically changed in Fig. 6.4(e) and (f), when the lamplight is turned on, and slowly changed in Fig. 6.4(a) (b) (c) and (d), when the illuminations are affected by the sunlight. In such environments, our proposed method is less affected by the illumination variations than others.

Fig. 6.6-Fig. 6.8 show the receiver operating characteristic (ROC) curves of the video clips by controlling thresholds. The results of each figure are estimated from 20 randomly selected test images. These images all include moving persons. The ground truth data of the test samples are manually labeled as shown in Fig. 6.5. The results show that the curves of SBM and MBM are very similar and the true positive rates of SBM are usually higher than that of MBM. When we fix the true positive rate on 80%, the false positive rates are about 21% and 30% for the test images captured by Cam2 (Fig. 6.7) using SBM and MBM, respectively. The results show that we can eliminate about 30% (9% in 30%) misclassified non-foreground pixels by extending MBM with

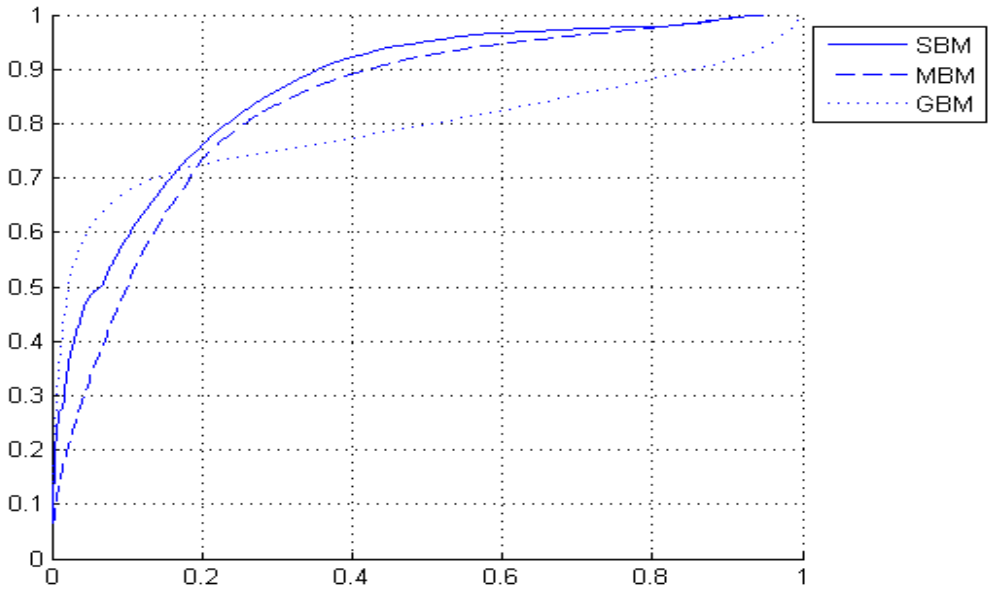


Fig. 6.6 The ROC curve of test images captured by Cam1.

spatial relations.

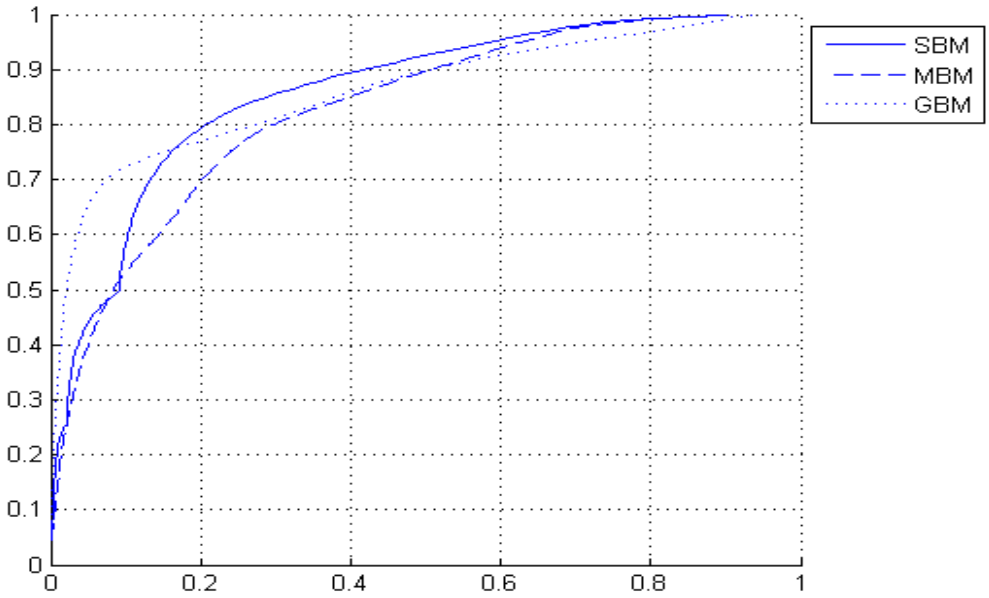


Fig. 6.7 The ROC curve of test images captured by Cam2.

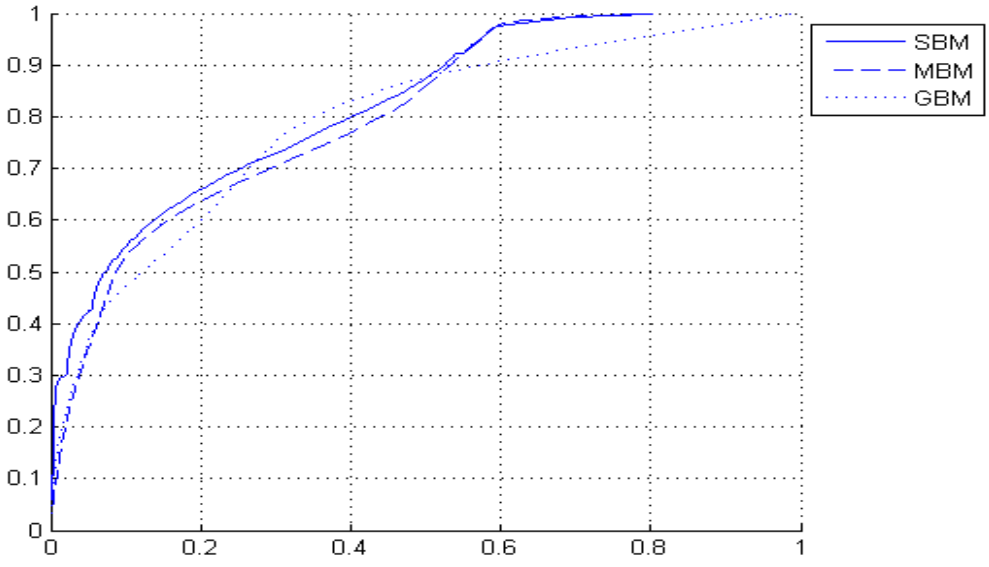


Fig. 6.8 The ROC curve of test images captured by Cam3.

Note that the performances of GBM are usually better than those of MBM and SBM for the samples captured by Cam1 and Cam2 when the false positive rate is lower than 15%. The reason is that the background appearances do not change frequently in the corridor. In the environment with less frequently changed background, a Gaussian distribution can easily model the color distribution. However, when the background changes, the performance of GBM may become unacceptable for a fixed threshold as shown in Fig. 6.4.

Although experimental results show that SBM usually outperforms MBM and GBM, the SBM is slower than the other two methods. The computation complex of calculating the background probability of a pixel of MBM is $O(K)$, where K is the number of background constituents, but SBM is $O(K \times M)$, where M is the number of pixel-pairs used. When we update the model of a pixel, the computation complex of MBM is still $O(K)$, but SBM is $O(K^2 \times M)$ since the computation cost of updating each weighting matrix is $O(K^2)$. On a PC with Pentium4 2 GHz CPU, the SBM can perform about one frame per-second, but the MBM is about 10 frames per-second. In our tests, about 90% CPU time spends on calculating the mutual information (Eq. (3.8)) and updating the matrix w (Eq. (3.5)).

6.2 Adaptive Color-Based Particle Filter

We evaluated the proposed adaptive color based particle filter on two video sets. The first set called the "TCU set", which is captured with four stationary cameras mounted on Tzu Chi University and on a house near Tzu Chi University. The cameras are mounted approximately three meters high and the angle between the camera and floor is smaller than 30°. The video set consists 10 video clips with 48 target sequences in four scenes. Fig. 6.9 shows the images of the four scenes. The captured

(a) (b)

(c) (d)

Fig. 6.9 Sample images of the four scenes in the TCU video set.

image size shown in Fig. 6.9(a) is 720×480, and others are 640×480. The sampling rate is about 15 fps. The test video and ground truth data are released in the ftp site ftp://203.64.84.203. The second set is 12 video clips selected from the CAVIAR video corpus [52] (six are front view and six are corridor view of the Shopping Center in Portugal). The image size is 384×288 pixels and the sampling rate is 25 fps. The details of the database can be found in [52].

To analyze the tracking accuracy and speed, we compared the proposed method with the method proposed by K. Nummiaro et al. [19]. In the K. Nummiaro's method, the target similarity is measured using weighted color histogram represented by $8 \times 8 \times 8$ matrix. The weighted color histogram is extracted as

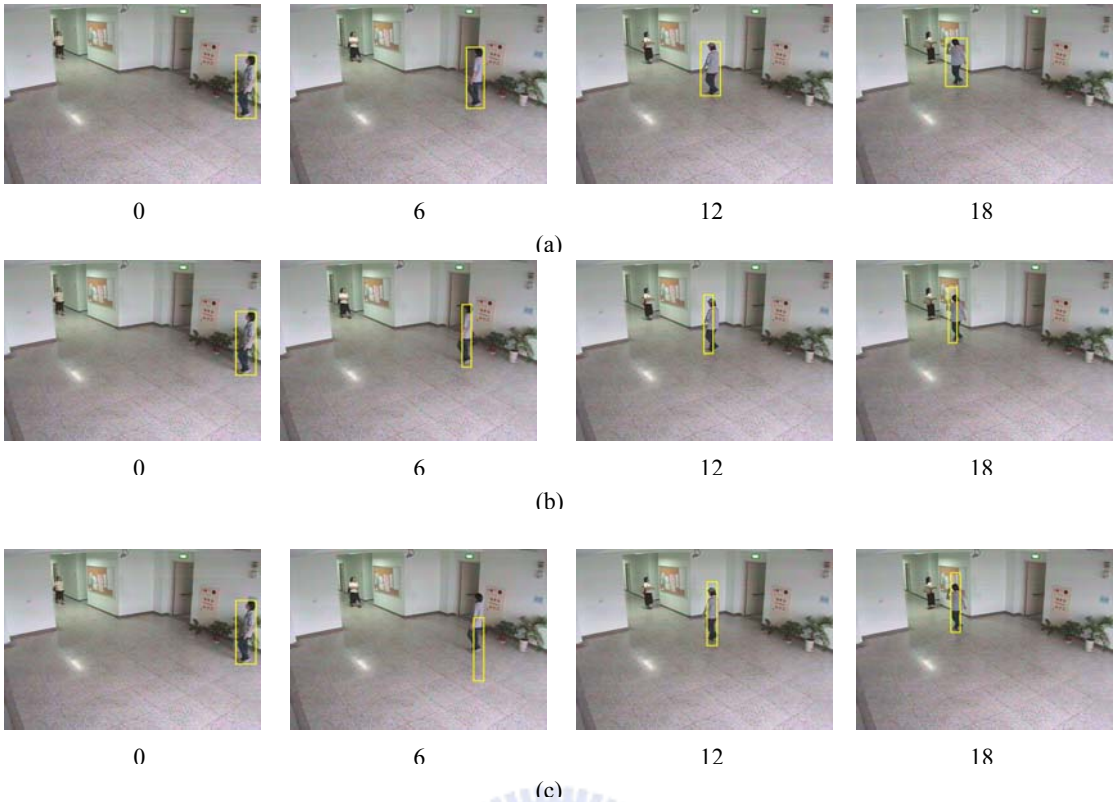
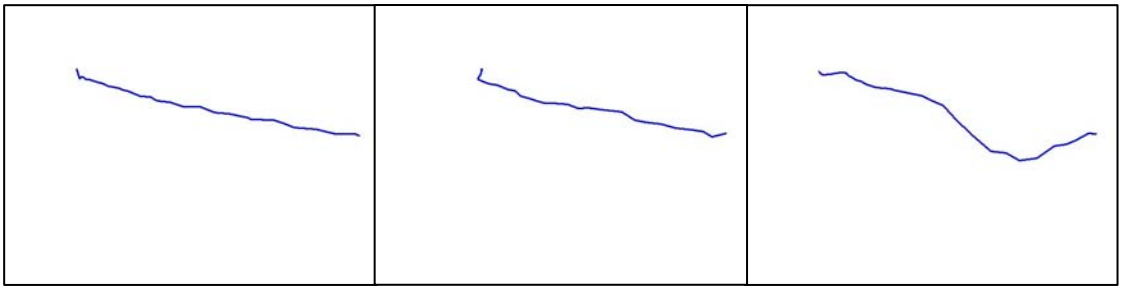


Fig. 6.10 Sample images and the tracking results of a target person. (a) The manually labeled bounding rectangle of the target person. (b) The tracking results created by the proposed method. (c) The tracking results created by K.Nummiaro's method.

$$p_y^{(u)} = f \sum_{i=1}^{I} k\left(\frac{\|y - x_i\|}{a}\right) \delta[h(x_i) - u], \tag{6.1}$$

$$k(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & \text{otherwise} \end{cases}, \tag{6.2}$$

where y is the center, I is the number of pixels in the regions, a is $\sqrt{\|W^2 + H^2\|}$ and f is the normalization factor that ensures $\sum_{u=1}^{m} p_y^{(u)} = 1$. In our proposed methods, we implemented the similarity measurement described in Sec.4.2. The experiments are performed in a personal computer with 3GHz CPU.



(a) (b) (c)

Fig. 6.11 The trajectories of a target person. (a) Trajectory of the target person from manually labeled ground truth. (b) Trajectory of the target person created by the proposed method. (c) Trajectory of the target person created by K. Nummiaro's method.

6.2.1 The TCU Video Set

In this research, the purpose is to create the trajectories for visual surveillance system. The centers of state vectors of a person in consecutive frames represent the trajectory as shown in Fig. 6.11. Fig. 6.11(a) shows the ground truth trajectory, which is created by connecting the centers of the manually labeled bounding rectangles. Several sample images of the manually labeled bounding rectangles are show in Fig. 6.10(a), where the number labeled below is the frame index. Fig. 6.10(b) and Fig. 6.10(c) show the trajectories created by the proposed method and K. Nummiaro's method, respectively, and Fig. 6.10(b) and Fig. 6.10(c) show the sample images of the tracking results of the two methods.

Table 1 Average Center Deviation by Different Similarity Measuement

#of particles	200	400	600	800	1000	2000
Our method	52.2	46.3	41.7	43	41.7	39.1
K. Nummiaro's method	78.4	73	73.5	73	71.3	68.6

Table 2 Tracking Speed by Different Similarity Measuement (fps)

#of particles	200	400	600	800	1000	2000
Our method	28.1	25	23.7	21.9	21.1	15.9
K. Nummiaro's method	12.3	6.1	4.1	3.2	2.5	1.3

Here, we want to measure the difference between trajectories of ground truth data and tracking results as the error measurement. Since the trajectories are composed by a set of center points, we measure the Euclidean distance between the center points of ground truth and those of the tracking result (called center deviation in following descriptions). Table 1 shows that the average center deviation of our proposed method is smaller than K. Nummiaro's method. Comparing with the sample images in Fig. 6.10, the colors of the person's clothes are similar to the colors of the floor. If we just use the color histogram of the target person to measure the particle weight, the particles located in the floor may be assigned high weight. Therefore, in Fig. 6.11(c), the trajectory of K. Nummiaro's method curves to the floor region of the scene. In our method, that kind of tracking failures can be reduced, since our similarity measurement considers both target appearances and background appearances.

Table 2 shows the comparison of tracking speed, where the speed is measured as

Table 3 Average Center Deviation on The Caviar Video Set by Different Similarity Measuement

# of particles	200	400	600	800	1000	2000
Our method	32.9	26.9	22.2	22.4	22.6	19.8
K. Nummiaro's method	44.7	42.1	41.2	39	41.3	40.5

Table 4 Tracking Speed on The Caviar Video Set by Different Similarity Measuement (fps)

# of particles	200	400	600	800	1000	2000
Our method	62.4	56.1	49.7	45	41.1	21.5
K. Nummiaro's method	50.3	26.4	17	12.9	10.3	4.6

the average frames per second (fps). In the table, K. Nummiaro's method is in inverse proportion to the number of the particles, since most computation costs of the method are spent on feature extraction and similarity measurement of particles. In our methods, most cost are spent on creating CHMs for current captured image and background image. When applying large number of particles, our method is significantly faster than the K. Nummiaro's method (over two times faster when apply 200 particles and about four times faster when apply 600 particles). The result shows that the proposed method can easily be applied to a realtime tracking system.

6.2.2 The CAVIAR Video Set

Table 3 shows the average center deviation and Table 4 shows the tracking speed when applying the tracking methods on the CAVIAR video set. The results also show that the proposed method can achieve high accuracy and high speed.

The results also show that the more particles cannot ensure higher accuracy. In Table 3, when we apply 2000 particles, the proposed method can reduce the average center deviation but K. Nummiaro's method cannot. By further analyzing the tracking results, when a large number of particles are used, the particles will widely separated in the state space, and several particles will located in background regions. If the correction process cannot assign distinguishable weights for target object and background regions, the tracker may lost the target. In our proposed method, since the weight assigned to target object and background are significant different, the precision can be improved when we apply 2000 particles. However, the widely separated particles also cause that the minimal bounding box of all particles become larger. Since the CHMs are created for the image in the minimal bounding box, the speed when applying 2000 particles is greatly dropped. Even though the speed is dropped, the proposed method can also achieve 21.5 fps. The speed is higher than that of the K. Nummiaro's method and approximates to the frame rate of the test videos.

6.2.3 Specific Histogram Mapping

We perform the experiment of the tracking model with specific histogram mapping, as described in Sec. 4.2.1, on the TCU video set. To verify the tracking model, we first define three rectangles on the head, torso, and hip-leg parts in 50 images as the target regions. We then compare the numbers of false-alarm target regions with and without specific histogram mapping. A wrongly classified region is defined as the region that does not overlap with the target rectangle but its histogram feature is similar to that of the target one, whose similarity is less than a threshold. The similarity measurement between two histograms is defined as that in [19]. To define the threshold, we select the regions that are overlapped with the target one with more than half of size and then calculate the average histogram similarity. To

(a) (b) (c

Fig. 6.12 he tracked body parts using different color histogram models. (a) Body parts in the initialization frame. (b) The tracked body parts using the equalized color histogram model. (c) The tracked body parts using the color histogram model without equalization. There are 7 frames apart between (a) and (b) and between (a) and (c).

calculate the number (or rate) of false alarms, we randomly define 1000 regions that do not overlap with the target one as the test regions from each test image. Among the 50000 test regions (1000 regions x 50 images), 66 (0.13%), 79 (0.16%), and 45 (0.09%) regions were wrongly classified as belonging to a body part when we use the proposed specific histogram mapping; and 973 (1.95%), 734 (1.47%), and 725 (1.45%) without specific histogram mapping which demonstrates the effectiveness of our proposed method.

Fig. 6.12 shows the tracked results using different color histogram models. Fig. 6.12 (a) is the three body parts extracted in the initialization step, where the upper rectangle denotes the head part, middle the torso part, and lower the hip-leg part. Fig. 6.12 (b) is the tracked three body parts using the equalized color histogram model and Fig. 6.12 (c) using a general color histogram model after several frames. Both the methods model the three channels of YCbCr space separately. In the images, the colors of the target person's hair are similar to those of a dark background region.

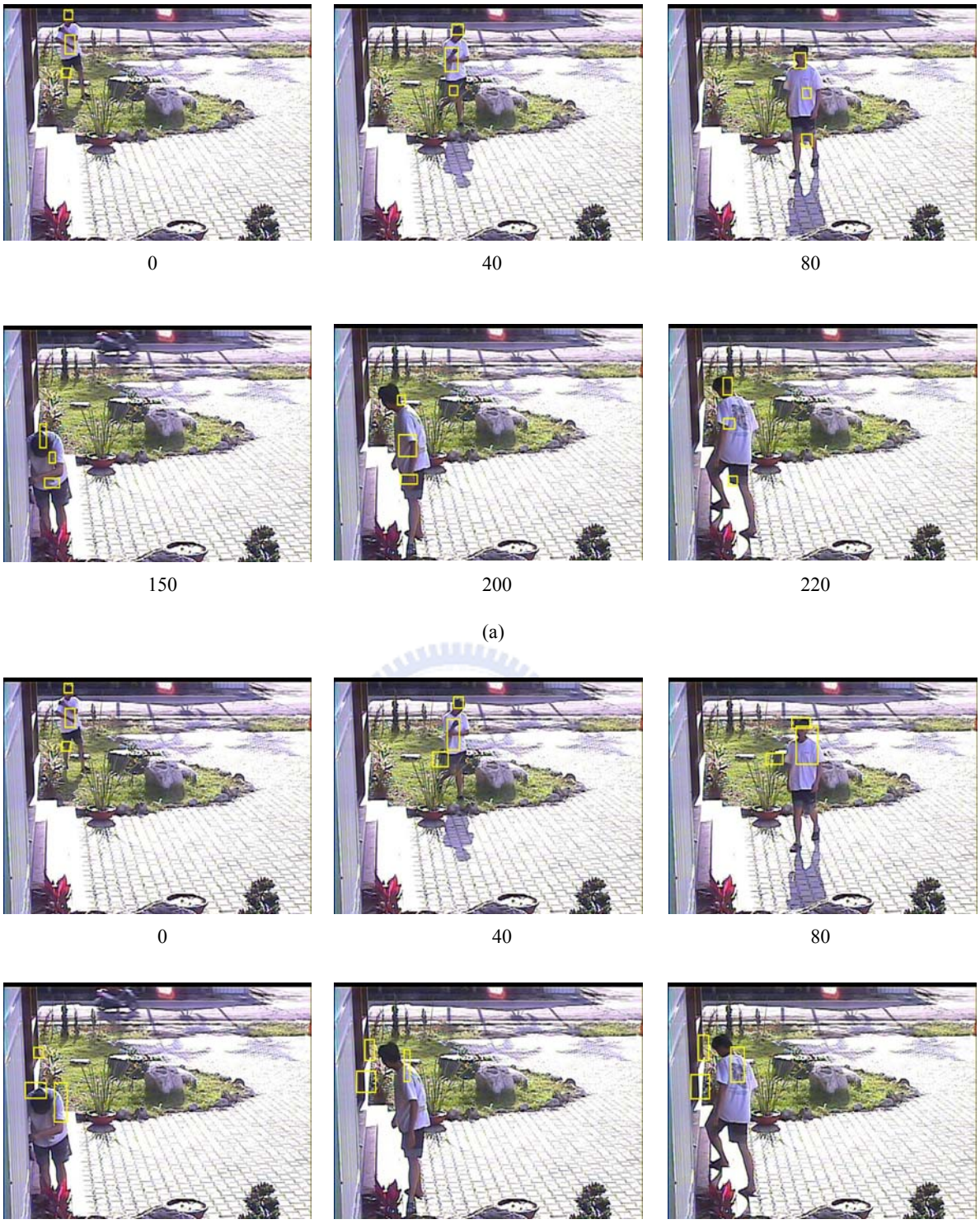


Fig. 6.13 The tracking results captured in an open space in front of a house. (a)Tracking results with failure detection and adjustment. (b) Tracking results without failure adjustment.

Using uniform quantization, the hair region and the background region may have similar feature vectors, so the head part may be tracked erroneously as shown in Fig. 6.12 (c). By using the equalized color histogram model, we can distinguish them as shown in Fig. 6.12 (b).

6.3 Tracking Failure Adjustment

In our experiments on tracking the head, torso, hip-leg, and whole body on 20 video sequences longer than 100 frames, the tracking failure rates are 35%, 35%, 40%, 20% in the 100-th frame, respectively. Note that if the bounding rectangle of a target object contains less than a half of regions of other objects, the target object is regarded as tracked correctly; otherwise it is considered as tracked incorrectly. In the test results, most of the failures, about 100%, 100%, 87.5%, 75%, are propagated from previous frames. If we can keep the correctly tracked body parts and adjust the positions of other body parts, error will not propagate easily to the following frames and the accuracy can be improved.

6.3.1 Tracking Results

Fig. 6.13 shows the human tracking results of scene 1 in the TCU video set by applying our proposed method with failure adjustment and that without failure adjustment [19]. The numbers below the images denote the frame number after tracking initialization. Fig. 6.13(a) is the tracking results using failure detection and adjustment, while Fig. 6.13(b) is the results without failure detection. Since several non-human regions have similar appearance to parts of the target person, the regions may be mistaken as the body parts as shown in the 40th and 80th frames of Fig. 6.13(b). In the frames, the appearances of the hip-leg part are similar to those of the stone. In the 200th and 220th frames, another background object is mistaken as the

0 20 40

(a)

0 20 40

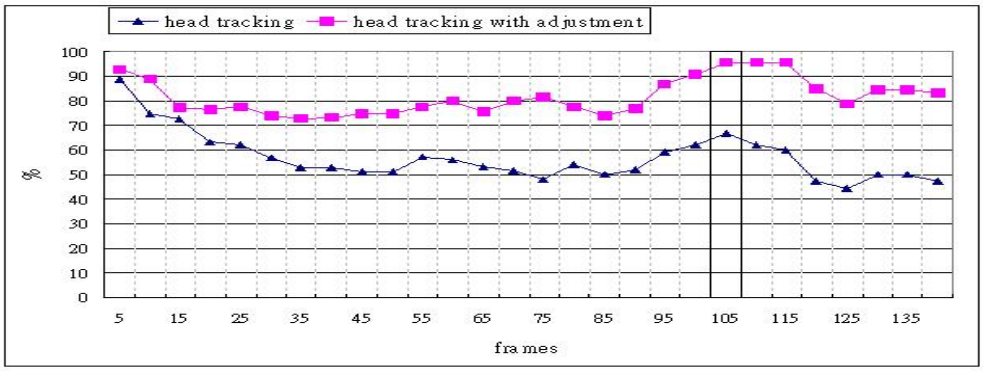
(b)

Fig. 6.14 The tracking results in a corridor. (a) Tracking results with failure detection and adjustment. (b) Tracking results without failure adjustment.

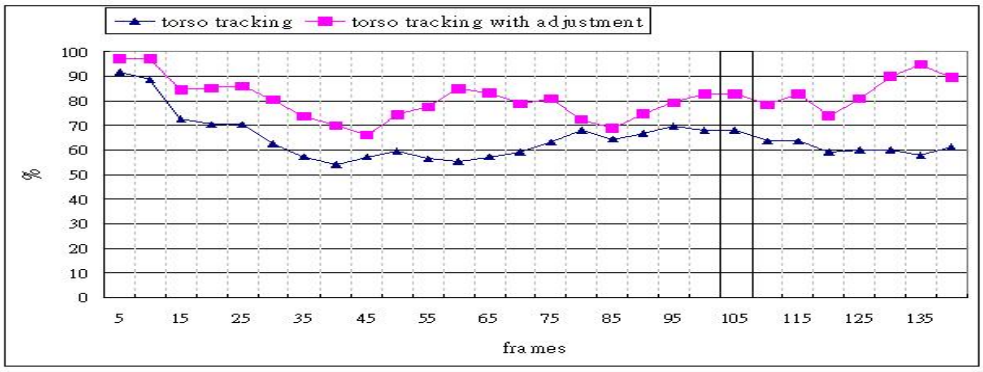
head part. In the 80th, 150th, 200th, and 220th frames, the tracking rectangles of the torso are slightly departed from the torso part, since the torso appearances are not similar to those in previous frames due to the motion of arms. Using our proposed method, we can detect the abnormal part and adjust its position as shown in Fig. 6.13(a). Note that in the 150th and 200th frames, the head tracking rectangle is slightly departed from the head part. According to our failure adjustment, the head is corrected in the 220th frame. Fig. 6.14 shows the tracking results in a corridor. The results also show that the failure adjustment is useful for tracking.

6.3.2 Analysis of Tracking Accuracy

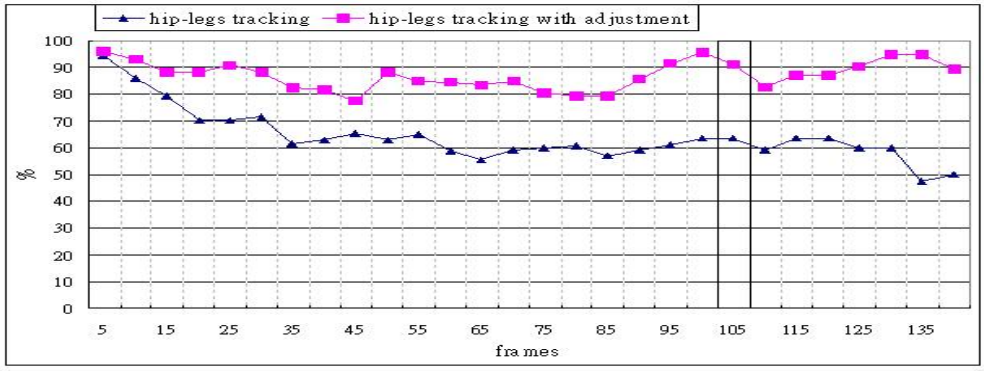
Fig. 6.15(a-d) show the tracking accuracy curves of the three body parts and the



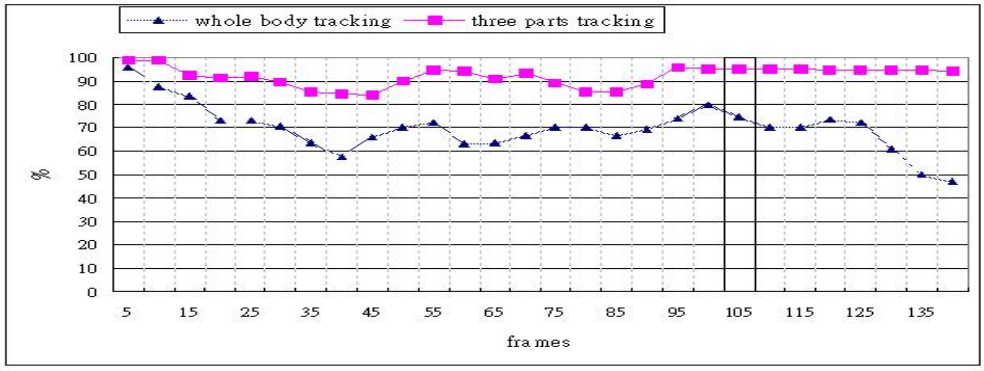
(a)



(b)



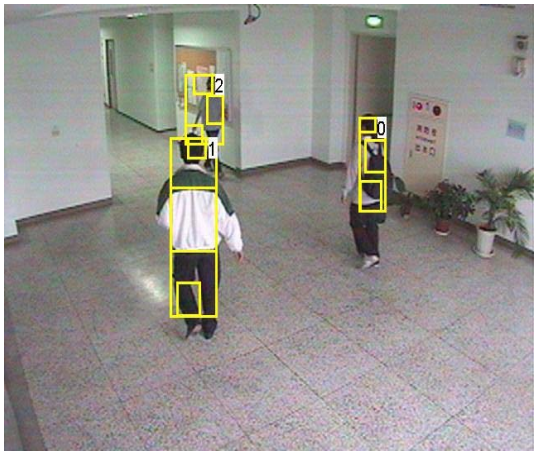
(c)



(d)

Fig. 6.15 Tracking rates without and with failure detection. (a) head. (b) torso. (c)hip-leg. (d) whole body.

0



5

10

12

14

20

Fig. 6.16 The tracking results with failure adjustment and inter-person occlusion detection.

whole body of the test video clips. Different persons appearing in a scene are recorded in different lengths of time intervals. Here, we test on 22 video clips with 72 target persons. The experiments were performed on 22 video clips with 72 target persons in six scenes. The input images are in color with resolution of 720×480. The number of particles is set within 200 to 1000, which was automatically modified by using the entropy of particle weights. The accuracy rate is defined as the ratio of the number of correctly tracked objects to the total number of detected objects. The accuracy rate of the *i*th frame is the ratio of the persons tracked correctly from initialization to the frame. Since the sequence lengths of different persons are different, the denominators of the accuracy rates for the sequence of frames are varied.

The curves in Fig. 6.15(a-c) show that the tracker with failure adjustment can improve the accuracy rate. The method of body part tracking without failure adjustment is similar to that proposed by Nummiar et al. [19] except for histogram extraction and weighting calculation, as described in Sec. 3. In frame 105, for instance, the tracking accuracy rates with failure adjustment are 95%, 83%, and 91% for the three parts respectively. Note that the result curves are not monotonically decreasing, since the particle filter can adjust the target parts to the correct positions, and the numbers of frames that different persons walk in a scene are not the same. Comparing with the accuracy rates of the tracker without failure adjustment, 67%, 68%, and 64%, the tracking rates of the three body parts are improved about 28%, 15%, and 27%. In these samples, the torso parts of the target persons in the sequences longer than 70 frames are relatively stable than other parts as shown in Fig. 6.13. Therefore the accuracy curves of the two methods in the 80-th frame of Fig. 6.15(b) are much similar.

To test the effect of the multi-part tracker with failure adjustment for human

(a) (b)

Fig. 6.17 The tracking failure samples. (a) A person bending down. (b) A sample affected by shadow.

tracking, we define the detected body region as the minimum bounding rectangle (MBR) that encloses the rectangles of the body parts. The single whole-body tracker is the same as that for the body part. Fig. 6.15(d) shows the accuracy curves for the whole-body tracker (drawn as triangles) and the MBR of the whole body from the three-part tracker with failure adjustment (drawn as squares). In frame 105, for instance, the accuracy of the three-part tracker with failure adjustment is 95%. Comparing with the accuracy rate 75% of the whole-body tracker, the tracking rate is improved about 20%.

6.3.3 Multi-person tracking

When we track multiple humans, the main problem is occlusion. When a person is occluded, the figure-ground segmentation may fail. We can use the system dynamic model to predict whether two persons are touching in a frame. In case of occlusion, we will find the occluded one and predict the target person until two persons are not touching anymore as described in Sec. 5.4.1.

Fig. 6.16 shows the images of a video that has multiple walking persons. In the first frame, the three persons from right to left are labeled as numbers 0, 1 and 2. In

the fifth frame, person 1 hides the hip-leg part of person 2, and in the 12th and 14th frames, person 0 hides almost the whole body of person 2. In the images the white rectangles denote the tracked persons. The results show that our tracker can track target persons even during inter-person occlusion.

6.3.4 Tracking failure analysis

In our tracking system, we assume that the relative positions of body parts are fixed in a certain range. However, the assumption cannot be applied to track the body parts of a person when his posture is not trained, such as bending down as shown in Fig. 6.17(a). In this situation, we can still track the same target person, but not his body parts correctly. Since these parts belong to the same person, we can still track the target person when he stands up.

In our failure adjustment, we use the background model to extract foreground regions. However, several false objects, such as shadow, may be regarded as foreground objects. In Fig. 6.17(b), the target person is still tracked but the torso part is too large and the hip-leg part includes shadow. Usually, the false detected foreground regions only affect the tracking results several frames. When the three tracked body parts are not located in the correct relative positions, the failure adjustment scheme will adjust the positions of the body parts.

7. Conclusions and Future Works

In this research, we designed a human tracking system. In the system, firstly, we have developed a spatially-extended background model for foreground detection. In the background model, we have used the probabilities of joint random vectors between near pixels to model the spatial relations. To reduce the cost of modeling the pixel-pairs, we calculate the mutual information in each pixel-pair for finding the spatially-dependent pixel-pairs.

In general environments, when the background regions are stable, the Gaussian background model is suitable to segment foreground regions. However, when background regions change, the model is unsuitable. To detect foreground regions more accurately with respect to either changed or still background regions, we should combine our propose model with Gaussian background model. To achieve this, some heuristic rules should be created for deciding which model should be selected. This is left for future studies.

To track a human, we decompose a human body into three parts, the head, torso, and hip-leg, and use color-based particle filters to track the three parts separately. We combined the appearance models of target object and background scene to calculate the weight of each particle. To reduce the redundancy during calculating color histograms in the overlapped regions of the particles, we have created a cumulative histogram map for each frame. We have also proposed an SVM-based method to detect the lost tracking part. In the tracking algorithm, we have used a particle filter for tracking an individual part. Since the particle number affects the tracking performance and tracking speed, we use entropy of particle weights to modify the

particle number dynamically. To further improve the tracking accuracy, we have designed a histogram equalization method for color histogram comparison. The experimental results show that the three parts tracking algorithm can improve the tracking accuracy significantly.

In this research, we assume that a human is standing up and the three body parts can be segmented from top to bottom. If a human crouches down or lies down, the body part decomposition may fail. Our experimental results show that in the case of failure, the three parts will not be labeled correctly. However, the failure will be adjusted after the target person stands up again, since we can detect the failure by SVM. To improve the body part decomposition, we may train detectors for different body parts. This is left for future research.

In our method, each of the three parts is tracked by a particle filter independently. The relative positions of the body parts are used to detect the tracking failure. We can reduce tracking failures by preventing the particles of abnormal poses to be generated. To achieve the goal, we needs to combine the state vectors of the three parts into a single vector to be tracked by a particle filter. Then the particle weights are adjusted according to the relative positions of the body parts. Also the behaviors of intruders defined on object appearances and the trajectories found will be analyzed. These are all left for future research.

Bibliography

[1] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Journal of Pattern Recognition*, vol. 36, no. 3, pp. 585-601, Mar. 2003.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, Aug. 2000.

[3] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, Jul. 1997.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1337-1342, Oct. 2003.

[5] Y. N. T. T. M. H. Shiry Ghidary, "Human detection and localization at indoor environment by home robot," in *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, Nashville, USA, 2000, pp. 1360-1365.

[6] A. Elgammal, R. Duraiswam, D. Harwood, and L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151-1163, Jul. 2002.

[7] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42-56, Oct. 2000.

[8] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: algorithms and evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 25, no. 7, pp. 918-923, Jul. 2003.

[9] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 22, no. 8, pp. 747-757, Aug. 2000.

[10] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 27, no. 5, pp. 827-832, May 2005.

[11] H. Wang and D. Suter, "A re-evaluation of mixture-of-Gaussian background modeling," in *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, PA, USA, 2005, pp. 1017-1020.

[12] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172-185, Jun. 2005.

[13] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image and Vision Computing*, vol. 22, no. 2, pp. 143-155, Feb. 2004.

[14] E. Durucan and T. Ebrahimi, "Change detection and background extraction by linear algebra," *Proceedings of the IEEE* , vol. 89, no. 10, pp. 1368-1381, Oct. 2001.

[15] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE transactions on image processing* , vol. 13, no. 11, pp. 1459-1472, Nov. 2004.

[16] L. Li and M. K. H. Leung, "Integrating intensity and texture differences for robust change detection," *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 105-112, Feb. 2002.

[17] Y. Wang, T. Tan, K.-F. Loe, and J.-K. Wu, "A probabilistic approach for foreground and shadow segmentation in monocular image sequences," *Pattern Recognition*, vol. 38, no. 11, pp. 1937-1946, Nov. 2005.

[18] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-Based Probabilistic Tracking," in *Proceeding of 7th European Conference on Computer Vision*, vol. 1, Copenhagen, Denmark, 2002, pp. 661-675.

[19] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99-110, Jan. 2003.

[20] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Los Alamitos, CA, USA, 2001, pp. 511-518.

[21] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 23, no. 4, pp. 349-361, Apr. 2001.

[22] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, Jun. 1998.

[23] H. Deng and D. A. Clausi, "Unsupervised image segmentation using a simple MRF model with a new implementations scheme," *Pattern Recognition*, vol. 37, no. 12, pp. 2323-2335, Dec. 2004.

[24] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90-126, Nov. 2006.

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

[26] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1-45, Dec. 2006.

[27] T. B. Moeslund and E. Granum, "A Survey of Computer Vision-Based Human Motion Capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231-268, Mar. 2001.

[28] D. M. Gavrila, "Visual analysis of human movement: A survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82-98, Jan. 1999.

[29] H. Sidenbladh, "Detecting human motion with support vector machines," in *Proceeding of International Conference on Pattern Recognition*, vol. 2, Cambridge, UK, 2004, pp. 188-191.

[30] Y. Ivanov, A. Bobick, and J. Liu, "Fast Lighting Independent Background Subtraction," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 199-207, Jun. 2000.

[31] A. S. Micilotta, E. J. Ong, and R. Bowden, "Detection and tracking of humans by

probabilistic body part assembly," in *Proceeding of British Machine Vision Conference*, Oxford, UK, 2005, pp. 429-438.

[32] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-575, May 2003.

[33] L. Zhao and C. Thorpe, "Stereo-and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148-154, Sep. 2000.

[34] C. Shan, T. Tan, and Y. Wei, "Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, no. 7, pp. 1958-1970, Jul. 2007.

[35] E. Polat, M. Yeasin, and R. Sharma, "Robust Tracking of Human Body Parts for Collaborative Human Computer Interaction," *Computer Vision and Image Understanding*, vol. 89, no. 1, pp. 44-69, Jan. 2003.

[36] S. L. Dockstader and N. S. Imennov, "Prediction for human motion tracking failures," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 411-421, Feb. 2006.

[37] Q. Zhou and J. K. Aggarwal, "Object tracking in an outdoor environment using fusion of features and cameras," *Image and Vision Computing*, vol. 24, no. 11, pp. 1244-1255, Nov. 2006.

[38] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, Aug. 1998.

[39] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle

Filters for Online Nonlinear/Non-Gaussian Bayesian Trackin," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[40] D. A. Forsyth and M. M. Fleck, "Body Plans," in *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997, pp. 678--683.

[41] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: single-frame classification and system level," in *Proceeding of IEEE Intelligent Vehicles Symposium*, Parma, Italy, 2004, pp. 1-6.

[42] "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," in *Proceeding of European Conference on Computer Vision*, Prague, Czech Republic, 2004, pp. 69-82.

[43] S. Ioffe and D. Forsyth, "Probabilistic Methods for Finding People," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 45-68, Jun. 2001.

[44] C. Chang, R. Ansari, and A. Khokhar, "Efficient tracking of cyclic human motion by component motion," *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 941-944, Dec. 2004.

[45] T. J. Roberts, S. J. McKenna, and I. W. Ricketts, "Human pose estimation using learnt probabilistic region similarities and partial configurations," in *Proceeding of European Conference on Computer Vision*, Prague, Czech Republic, 2004, pp. 291-303.

[46] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Strike a Pose: Tracking People by Finding Stylized Poses," in *Proceeding of Computer Vision and Pattern Recognition*, vol. 1, Washington, DC, USA, 2005, pp. 271-278.

[47] B. Wu and R. Nevatia, "Detection and Tracking of Multiple, Partially Occluded

Humans by Bayesian Combination of Edgelet based Part Detectors," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247-266, Nov. 2007.

[48] C. Lerdsudwichai, M. Abdel-Mottaleb, and A.-N. Ansari, "Tracking Multiple People with Recovery from Partial and Total Occlusion," *Pattern Recognition*, vol. 38, no. 7, pp. 1059-1070, Jul. 2005.

[49] S. Khan and M. Shah, "Tracking people in presence of occlusion," in *Proceeding of Asian Conference on Computer Vision*, Taipei, Taiwan, 2000, pp. 1132-1137.

[50] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462-467, May 1968.

[51] D. Fox, "Adapting the Sample Size in Particle Filters Through KLD-Sampling," *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985-1003, 2003.

[52] CAVIAR. [Online]. http://www.dai.ed.ac.uk/homes/rbf/CAVIAR