摘要

在這篇論文裡，我們發展了一套具有分析人類行動能力的**即時視覺監視系統**（**real-time visual surveillance system**）。設備爲一支彩色單眼錄影機；環境參數爲靜止不動的背景（stationary background）。在一開始，系統會結合傳統的電腦追蹤演算法-**背景相減法**及**混合高斯模型（GMM）**，將出現在鏡頭裡的目標物偵測出來，並萃取出其輪廓。接著，系統使用形狀及幾何分析所獲得的特徵值，成功的將人體各部位區分開成尚未定義的區塊。並利用我們所發展的**階層式形狀統計相似演算法（HSSS）**標明被區分開的區域（頭、手、軀體…等）。當系統成功地將以上區塊分辯出來，便會估算出被觀察者身體各部位的物理特徵值，例如：型體重心、主軸角度、長寬比例…等。最後再利用這些估算值，與事先建立好的資料庫作比對。此資料庫是以動作（actions）爲基礎所建立的。藉此系統可以自動地監視被觀察者以及適時地發出警告訊息。此系統可以20~25Hz 的速度、240x160 的解析度在 Pentium-M 1600MHz 的 PC 上作用。

Abstract

In this thesis, we develop a **real-time visual surveillance system** for human behavior analysis. It operates on monocular color-scale video imagery with a stationary background scene. At the first step in the system process, it extracts the silhouette of the target object by traditional video tracking method, background subtraction combined with Gaussian Mixture Model (GMM). And furthermore, it detects the contour of the object silhouette. At the second step, the system employs a combination of shape analysis and geometry analysis on the contour to decompose the detected silhouette to several undefined parts（unlabeled body parts）. After the decomposition process, it labels each the separated part （head、torso、hands、feet）by the use of our **hierarchical statistical-shape-similarity algorithm (HSSS)**. As the above steps have been processed successfully, the last step in our system is to extract local features of the detected body part（orientation、centroid. . . etc）, and the global features of the entire silhouette（aspect ratio、 block density. . . etc）, and then these features can be used to guide the high-level human behavior analysis. In the on-line behavior analysis process, an unknown sequence will be matched with the templates collected in our database. The database is established offline by the use of real video captures, which is a group of labeled reference sequence representing typical behaviors. In short, our system can detect the human body parts and classify the posture of human at individual imagery, then identify the event of a query sequence which involves human beings. It runs at 20~25Hz for 240 x 160 resolution images on a single Pentium-M 1600Mhz PC.

Table of Contents

List of Figures

List of Tables

1 Introduction

As an active research topic in computer vision, visual surveillance attempts to detect, recognize and track certain objects from image sequences, and more generally to understand and describe object behaviors. The aim of this work is to develop intelligent visual surveillance system to replace the traditional passive video surveillance. In short, the indeed visual surveillance system is not only to put cameras in the place of human eyes, but also to accomplish the entire surveillance task as automatically as possible.

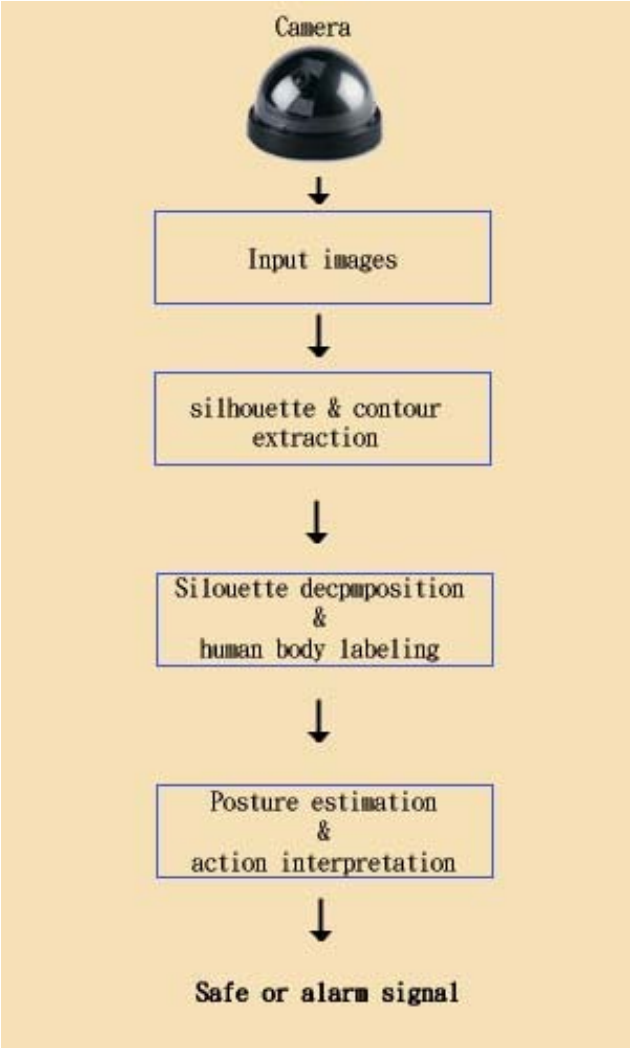


Figure 1-1 General framework of a visual surveillance system

Figure 1-1 shows the general framework of a visual surveillance system. The prerequisites for effective automatic surveillance using single camera include the following stages: **modeling of environments**, **detection of motion**, **classification of moving object**, **tracking**, **understanding and description of behaviors**. In order to extend the surveillance area and overcome occlusion, fusion of data from multiple cameras is needed. This fusion can involve all the above stages.

1.1 Motion Detection

Nearly every visual surveillance system starts with motion detection. Motion detection aims at segmenting regions corresponding to moving object from the rest of an image. Subsequent processes such as tracking and behavior analysis are greatly dependent on it. The process of motion detection usually involves environment modeling, motion segmentation, and object classification. The following discussions would be all considered in a fixed camera environment.

Environment Modeling

The key problem in environment modeling is to automatically recover and update background images. Unfavorable factors, such as illumination variation, shadows. . .etc, introduce many difficulties to accomplish the goal. There are many algorithms for resolving the above mentioned problems. Temporal average of an image sequence [15], adaptive Gaussion estimation [16], and parameter estimation based on pixel process [17] are all good algorithms for this mission. A classic method was proposed by Ridder et al. [18]. They model each pixel value with a Kalman filter to compensate for illumination variance.

Motion Segmentation

After the background is modeled, the next step is to analyze the activities of the foreground objects. For capturing and analyzing the foreground objects, the movement contributed by each object has to be independently captured. Therefore, motion segmentation has to be executed in advance. When on mentions motion segmentation in an image sequence, it means to detect regions corresponding to moving object such as vehicles and human beings. At present, most segmentation methods use either temporal or spatial information in an image sequence. **Background subtraction** [19] is a popular method for motion segmentation, especially under static background situation.. It detects moving regions by calculating the difference between the target image and the reference image in a pixel-by-pixel fashion. It is simple, but extremely sensitive to changes in dynamic scenes derived from light variation and extraneous events. Therefore, it is highly dependent on a good background model to reduce the influence of these changes. Lipton et al. proposed a method called **temporal differencing** [8] to solve the above mentioned problem. They use a threshold function to determine changes after the absolute difference between current image and previous image is obtained. It is very adaptive to dynamic environments, but generally does a poor job of extracting all the relevant pixels, e.g., there may be a hole left inside moving entities. Another way to do the motion segmentation job is **Optical flow** [20]. But unfortunately, it is computationally complex and sensitive to noise, and cannot be applied to real-time video stream.

Object Classification

The last mission for motion detection is object classification. As different moving regions may correspond to different moving targets in a natural scene, to further track objects and analyze their behaviors, it is necessary to correctly classify moving

objects. For instance, the image sequences captured by surveillance cameras mounted in road scenes probably include humans, vehicles, flying birds . . .etc. At present, there are two main categories of approaches for classifying moving objects. A lot of different descriptions of shape information can be applied to classify moving objects, such as points, bounded boxes, silhouettes and blobs. VASM [14] takes image blob representation for the detected moving objects, and uses the aspect ratio of the image blobs bounding box to classify moving-object blobs into four classes: single person, vehicles, human groups, and clutter. This kind of approach is so called **Shape-based classification**. The other approach is **Motion-based classification**. In general, non-rigid articulated human body motion shows a periodic property, so this has been used as a strong cue for classification of moving objects. Cutler et al. [21] describe a similarity-based techniques to detect and analyze periodic motion. We know, for periodic motion, its self-similarity measure is also periodic. Therefore, time-frequency analysis is applied to detect and characterize the periodic motion, and tracking and classification of moving objects are implemented using periodicity.

1.2 Object Tracking

After motion detection, surveillance systems generally track moving objects from one frame to the next in an image sequence. The tracking algorithms usually have considerable intersection with the motion detection during processing. Useful mathematical tools for tracking include the Kalman filter, the Condensation algorithm, the dynamic Bayesian network, the geodesic method. Recently, McKenna et al. [19] propose an adaptive background subtraction method in which color and gradient information are combined to cope with shadows and unreliable color cues in motion segmentation. Isard et al. [22] make a big contribution to the tracking field, which is

so called **active contour-based tracking**. They adopt stochastic differential equations to describe complex motion models, and combine this approach with deformable templates to cope with people tracking. In contrast to any other tracking algorithms, an active contour-based algorithm describes objects more simply and more effectively. In addition, it is able to significantly reduce computational complexity. Even under disturbance or partial occlusion, an active contour-based algorithms can still continuously track objects.

1.3 Understanding and Description of Human Behaviors

After successfully tracking the moving objects from one frame to another in an image sequence, the next step we should do is to choose the model of human and to match the detected moving objects with it. The essence of human motion is typically contained in the movements of the torso, the head and the four limbs, so the **stick-figure** model is usually the first choice in people's mind. Karaulova et al. [23] use a stick figure representation to build a novel hierarchical model of human dynamics using hidden Markov models (HMMs). **2-D contour** is a kind of human model directly relative to human body projections in an image plane. Ju et al. [24] propose a cardboard human model, in which the human limbs are represented by a set of jointed planar ribbons. The main disadvantage of 2-D models is the requirement of the restrictions on the viewing angle. To overcome this disadvantage, many approaches apply the **volumetric models**, such as elliptical cylinders, cones [25], spheres. Plankers et al. [26] present a **hierarchical human model**. It includes four levels: skeleton, ellipsoid meatballs, polygonal surface skin, and shaded rendering. In short, the more accuracy human model achieves, the more computation time is needed.

There is always a trade-off between the accuracy requirement and computation efficiency..

Behavior understanding involves the analysis and recognition of motion patterns, and the production of high-level description of actions and interactions. It may simply be thought of the classification of time varying feature data, i.e., matching an unknown sequence with a group of labeled reference sequences representing typical behaviors. **Dynamic time warping** [27][28]is a template-based dynamic programming match technique widely used for speech recognition. It has the advantage of conceptual simplicity and robust performance, and has been used recently in the match of human movement patterns. An HMM is a kind of stochastic state machines [29]. It allows more sophisticated analysis of data with spatio-temporal variability. There are some proposed methods in this field, such as time-delay neural network（TDNN）[30], syntactic techniques [31], and non-deterministic finite automata（NFA）[32].

All the effort introduced above have one common goal. i.e. **automatic surveillance**. Although there is a lot of progress in visual surveillance field, some key problems remain open, for example, what is the most efficient representation of human body model, how to properly represent semantic concepts and how to map motion characteristics to semantic concepts. In the following, we will shortly introduce our visual surveillance system.

1.4 The Proposed Surveillance System

Figure 1-2 shows the architecture of our surveillance system. It operates with one single fixed camera in a stationary background scene. First, the system takes the

monocular color-scale video imagery from the camera. Then it extracts the silhouette of the detected moving objects in the input image. Object contour will be detected from the object silhouette. Gaussian filter [33] and negative curvature minimum criteria [56][57] are applied to smooth the detected object contour and to decompose the object silhouette. The system labels the decomposed parts of the object silhouette corresponding to human body parts by our proposed algorithm **HSSS**. After the process has been done successfully, it extracts the global features of the entire object silhouette and the local features of the labeled human body parts. Hereby, our visual surveillance system employs a hierarchical analysis to estimate the posture and interpret the action（behavior）of the detected moving object in the image sequence. Furthermore, a safe state remains or an alarm signal is sent out.

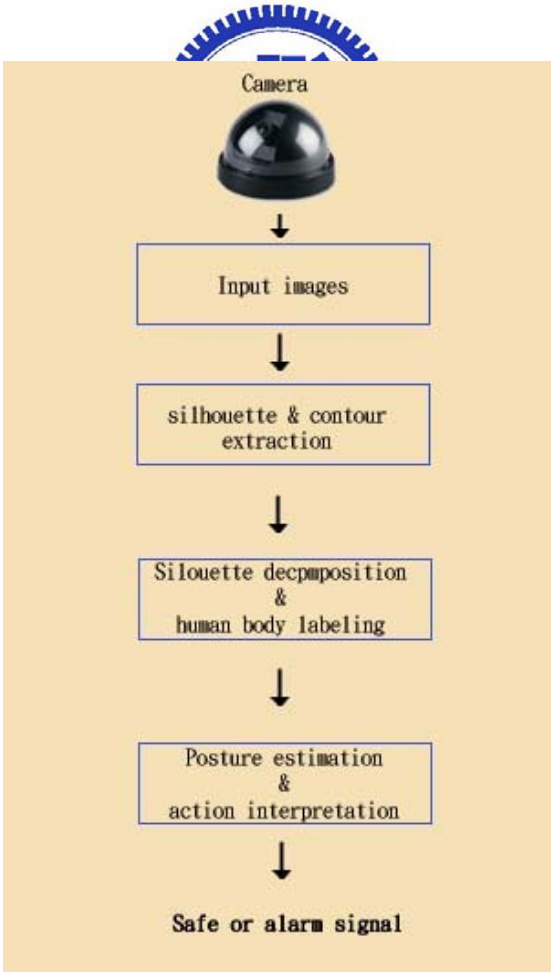


Figure 1-2　Architecture of the proposed surveillance system

2 Recent Developments and Activities

There have been a number of famous visual surveillance systems developed in the post few years [4-8]. The real-time visual surveillance system W4 [4] employs a combination of shape analysis and tracking, and constructs models of people's appearances in order to detect and track groups of people as well as monitor their behaviors even in the presence of occlusion and in outdoor environments. This system uses a single camera and grayscale sensor. The VIEWS system [5] developed by the University of Reading is a 3-D model based vehicle tracking system. The Pfinder system developed by Wren et al. [6] is used to recover a 3-D description of a person in a large room. It tracks a single non-occluded person in complex scenes, and has been extensively used in many real-world applications. As a single-person tracking system, TI, developed by Olsen et al. [7], detects moving objects in indoor scenes using motion detection, tracks them using first-order prediction, and recognizes behaviors by applying predicates to a graph formed by linking corresponding objects in successive frames. This system cannot handle small motions contributed by background objects. The system at CMU [8] can monitor activities over a large area using multiple cameras that are connected into a network. It can detect and track multiple persons and vehicles within cluttered scenes and monitor their activities over long periods of time.

Due to the broad range of applications it can cover, visual surveillance motivates the interests of researchers worldwide. The IEEE has sponsored the IEEE International Workshop on Visual Surveillance on three occasions, in India (1998), the U.S.（1999）,

and Ireland (2000). In June and August of 2000, two special issues on visual surveillance was published [9] [10]. In March of 2001, a special issue on visual analysis of human motion was published [11]. In October of 2001, a special issue on third-generation surveillance systems was published [12]. In October of 2002, a special issue on understanding visual behavior was published [13]. Also, visual surveillance has been investigated worldwide under several large research projects. The Defense Advanced Research Project Agency（DARPA）supported the Visual Surveillance and Monitor（VASM）project [14] in 1997, whose purpose was to develop automatic video understanding technologies that enable a single human operator to monitor behavior over complex environments such as battlefields and civilian scenes.

All of the above activities are evidence of a great and growing interest in visual surveillance.

3 Human Modeling

Human modeling is an essential part of model-based human detection. Although a great number of human models have been proposed in the literature, few of them are appropriate for human detection. Most models are developed for other purposes, such as human tracking or figure animation [36]. These models are either too complicated to be practical for efficient human detection, or can just be used to detect a particular person rather than all instances of humans. The common drawbacks of previous human models are：

(1) The representations of human shapes are not invariant to similarity transforms, thus, they can only detect people of a fixed size or orientation.

(2) The models are usually specific to a particular person, and do not model the statistical variance among individuals.

(3) Although some models such as deformable templates can handle certain global shape variance, they have difficulty dealing with large articulated motion and partial occlusion.

3.1 Related Work

Human modeling is a hot area and has attracted the attention in the past few years[6, 24, 36-52]. Among different types of human models, most models employ part-based representations to handle articulation. They vary widely in their level of detail. One group of researchers crudely model the body as a collection of articulated planar patches [24]. Another group of people develop 3-D models with deformable limb shapes [36].

For part-based 2D models, the representation of parts varies from planar patches [24] and 2D ribbons [37,38] to deformable models [39]. The advantage of using 2D models for recognition is that the matching is between 2D and 2D. The disadvantage is that it is hard for 2D models to deal with shape variations due to the change of viewpoint. For 3D models, if 3D data is available we can match the model directly against the data. Gavrila and Davis [40] proposed a complex 3D model of the body that takes into account kinematic constraints, but their method requires searching through a high dimensional pose parameter space for 3D pose recovery. If only 2D data is available, we need to match the 3D model against the extracted 2D data.

Assumptions about the viewing conditions vary from scaled orthographic projection [41] to full perspective [42,43]. To account for large variations in depth, Hogg [44] modeled the body in terms of articulated 3D cylinders viewed under perspective projection. More sophisticated tapered cylinders [43,45] or super-quadrics [46] have been employed. Bowden et al. [47] encapsulated the correlation between 2D image data and 3D skeleton pose in a hybrid 2D-3D model trained on real life examples. The

model they used allows 3D inference from 2D data, but their method does not generalize easily to new camera positions, because their 2D model is not invariant to viewpoint. The common drawback with the above models is that they do not model the statistical variation among individuals and the effects of clothes on human shape. Thus, they may be used for human tracking or figure animation, but they are not appropriate for detecting people of various shapes and clothing.

Marr and Nishihara [48] proposed a hierarchical 3D human model. At the highest level of the hierarchy, the body is modeled as a large extended cylinder, which is then resolved into small cylinders forming limbs and torso, and so on to fingers and toes. This hierarchical representation is stable in the presence of noise and sensitive to fine-level features, but is impractical because it contains few actual constraints to support human detection.

Contour-based representations have been used to model the 2D human shape. Baumberg et al [49]. and Sullivan et al. [82] employed a deformable template to handle shape deformation, where the shape model is derived from a set of training shapes. The orthogonal shape parameters are estimated using Principal Component Analysis(PCA). One drawback with this approach is that the model and the extracted contour should be aligned first, which is not a trivial task. Another drawback is that some invalid shapes are produced by the combination of two or more linear deformations. Gavrila et al. [50] developed a template hierarchy to capture the variety of human shapes, and the model contains no invalid shapes. The common drawback with the above approaches is that they do not model individual parts, and so they can only handle limited shape variety due to articulation and cannot deal with occlusion very well.

Skeleton-based representations [51] have been used to model the topological structure of the human body, but they do not model the shapes of body parts. These approaches are sensitive to noise and cannot distinguish two classes with the same topological structure but different geometrical structures.

Some models incorporate other cues or features into the model. Pentland [6] introduced a blob-based representation that combines skin color and contour to represent a body part. While the color-blob representation of a person is quite useful, it is not invariant under clothing/lighting changes and so it requires an initial model learning procedure for different subjects and a smoothly changing image background. Papageorgiou et al. [52] developed a wavelet-based representation to model pedestrians, but this representation is not invariant under rotation and cannot handle large part movements and occlusion very well.

In summary, previous work for human modeling just operate on some special purpose or in some limited field.

3.2 Part-Based Human Body Model

Shape representation is a major problem in computer vision and is the basis for recognition. Here we define the word shape：

Definition 3.1

The **Shape** is the geometry of an object's occluding contour in 2-D space.

The requirements of a good description that facilitates recognition lead to representations that are segmented and hierarchical. Thus, shape decomposition is a key stage in a part-based shape representation. Here, we need to make some definitions clear：

Definition 3.2

A **Part** of an object is a region bounded by a portion of the outline of a silhouette and one or more **cut**s.

A **Cut** of an object is the boundary between two adjoining parts of the object. It strictly passes through just two points on the outline of the object.

A silhouette can be decomposed in many different ways as shown in Fig. 3-1, but for the task of recognition not just any partitioning scheme will do. The decomposed parts

must satisfy certain requirements for recognition:

1. They should correspond to the natural body parts of an object.
2. The decomposition should be invariant under translation, rotation, and scaling.
3. The decomposition should be computable.

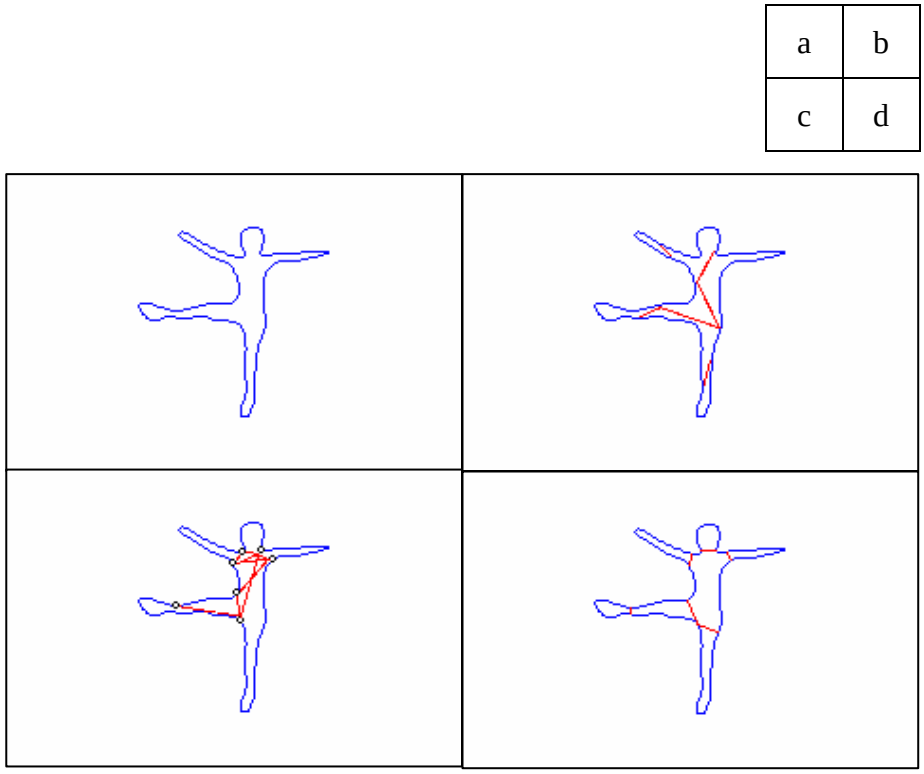


Figure 3-1 Silhouette decomposition (a) original silhouette (b) random decomposition (c) decomposition at NCM (d) natural decomposition

These requirements suggest that to break a shape into parts we should use its intrinsic geometry. Figure 3-2 shows the flow chart of our part-based human silhouette decomposition process. In the following, we will discuss each stage step by step. In

section 3.2.1, a human silhouette extraction process based on background-subtraction is introduced. After this extraction process, a contour smoothing algorithm applying Gaussian filter is presented in section 3.2.2. Then, curvature estimation for every point on the smoothed contour is introduced in section 3.2.3. At last, we will present the main process for human silhouette decomposition in section 3.2.4 by applying the features discussed above .

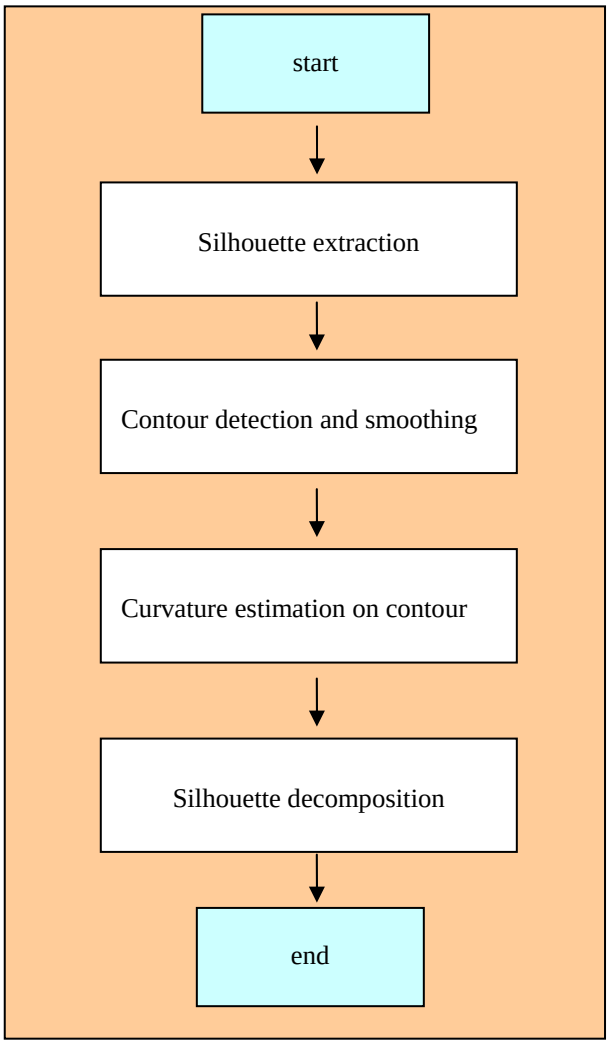


Figure 3-2 Flow chart of the proposed human silhouette decomposition

3.2.1 Human Silhouette Extraction

Background subtraction is one of the main techniques to extract moving objects from background scenes. A Gaussian Mixture Model (GMM)[53]is a frequently used model for background subtraction. A famous approach is Stauffer's paper [53] which models each background pixel's distribution using a GMM; this model allowed to monitor continuously a university campus.

Our system requires three steps to complete the process：background modeling, background estimation ,and background updating. Figure 3-3 shows the flow chart of our silhouette extraction process.

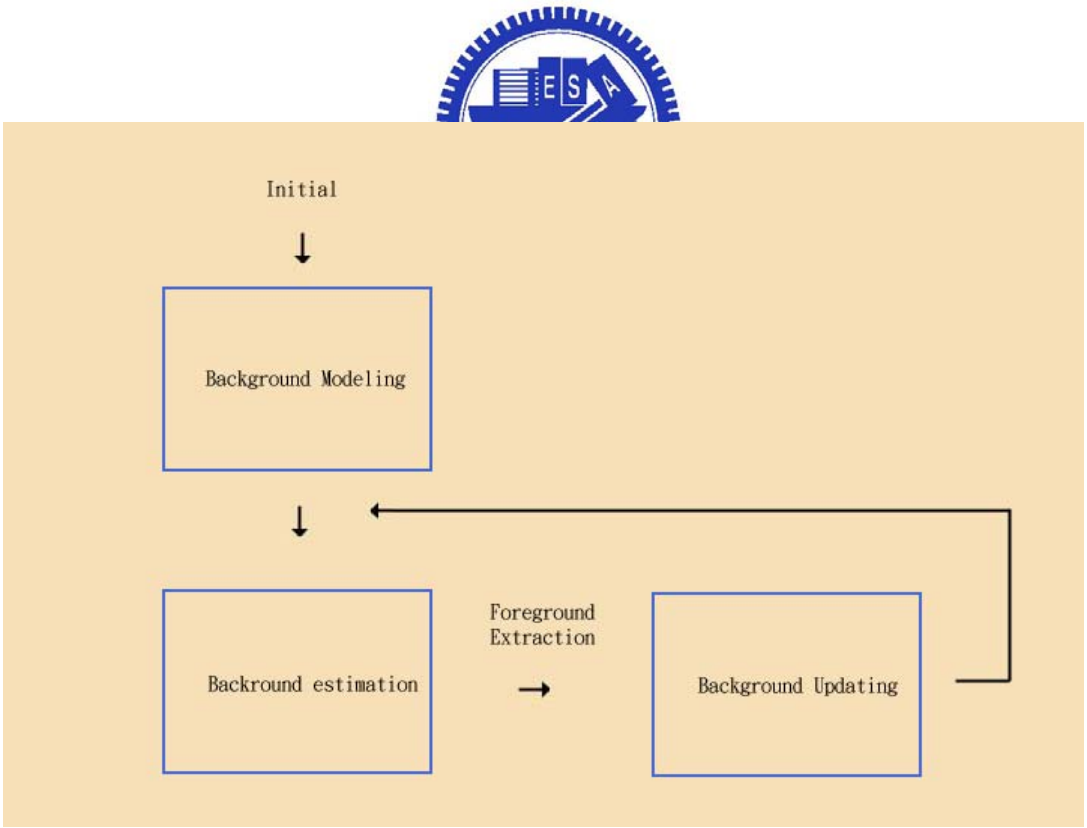


Figure 3-3 Flow of silhouette extraction.

Background modeling

For each application, the first and important step is to extract moving objects from the background (background subtraction defines the moving objects). Each background pixel is modeled using a mixture of Gaussian distributions[53]. The Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the "background process". Each pixel at (i, j) is modeled by a mixture of κ Gaussians, $G_{ij,1}...G_{ij,k}$,as stated in the formula：

The probability of observing pixel value

$$P(x_{ij,t_\rho}) = \sum_{h=1}^{k} \omega_{ij,h} \cdot G_{ij,h}(x_{ij,t}, \mu_{ij,h}, \Sigma_{ij,h}) \qquad (3.1)$$

After a learning duration T, we have images $I_1...I_T$.

where $I_t = \{x_{ij,t}\}$

$$x_{ij,t} = (R_{ij,t}, G_{ij,t}, B_{ij,t}) \quad , \quad \forall t = 1,...,T$$

For each pixel at (i, j) , cluster $x_{ij,1},...,x_{ij,T}$ into κ clusters $C_{ij,1}...C_{ij,k}$,

then have

$\mu_{ij,h}$ ：mean of the h th cluster at position (i, j), $C_{ij,h}$

$\Sigma_{ij,h}$ ：covariance matrix of the h th cluster at position (i, j), $C_{ij,h}$

$$G_{ij,h}(x_{ij,t_\rho}, \mu_{ij,t}, \Sigma_{ij,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma_{ij,t}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_{ij,t_\rho} - \mu_{ij,t})^{Trans.} \Sigma^{-1} (x_{ij,t} - \mu_{ij,t})},$$

where $n = \dim(x_{ij,t})$

$\omega_{ij,h}$ ：weighted parameter of $G_{ij,h}$

usually defined as $\frac{size(C_{ij,h})}{T}$

Normally we choose κ equals to 3 for indoor scenes and κ equals to 5 for outdoor scenes. For computational convenience, the covariance matrix is assumed to be the form： $\Sigma_{ij,h} = \sigma_{ij,h} \cdot \mathrm{I}$. This assumes that the red, green, and blue pixel values are independent and have the same variances.

Background estimation

By following the approach proposed in Stauffer's thesis [53], we add some post processes to extract moving objects with more accuracy. In summary, background estimation requires to work on：pixel-level process, frame process, region process.

Pixel level process： This process comes first. Every new pixel value $x_{ij,t}$ is checked against the existing κ Gaussian distributions until a match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution. The background estimation problem is solved by specifying the Gaussian distributions, which have the most supporting evidence and the least variance. Because a moving object has larger variance than a background pixel, so in order to represent background processes, first the Gaussians are ordered by the value of $\frac{\omega_{ij,h}}{\Sigma_{ij,h}}$ in decreasing order. The background distribution stays on top with the lowest variance by applying a threshold, where

$$B = \arg\min_{b} \left(\frac{\sum_{h=1}^{b} \omega_{ij,h}}{\sum_{h=1}^{k} \omega_{ij,h}} \right) > Threshold \qquad (3.2)$$

Then the first B distributions are chosen as the background model. All pixels $x_{ij,t}$ which do not match any of these distributions will be marked as foreground. At this stage, the obtained foreground mask (we use **Mask** to represent it) contains errors. The foreground part may have holes due to misclassified pixels. shadow represents another error source, together with the noises occurred in the imagery process , they both make the background pixels misclassified as foreground pixels frequently. To avoid these errors, we continue the processes at the region level and the frame level.

Frame level process：The frame level process comes second, and it is basically defined by frame differences. Let I_t represent the current image, I_{t-1} the previous image, and I_{t+1} the subsequent image. We use D_{t-1} and D_t to represent frame differences. We have $D_{t-1} = I_t - I_{t-1}$ and $D_t = I_{t+1} - I_t$. The pixels being identical in all three images are in the foreground region of image I_t. We ensure that these are also in the foreground mask obtained from the previous pixel process, indicated by the following updating formula：

$$MASK = MASK + D_t \wedge D_{t-1} \qquad (3.3)$$

Region level process：The region level process comes last. The foreground mask at this step may contain Salt and Pepper noises or small holes. We designed an extra post processing step. Most approaches use opening and closing to remove such noise and fill in small gaps. But this method cannot fill in large holes. We use a different method by applying a 5x5 window. By running this window on our foreground mask, we can remove noise, shrink and fill small or large holes. Each pixel P in the foreground

mask is the center point in the 5x5 window, so there are 8 points surrounding P forming a 3x3 window, and another 16 points surrounding this 3x3 window. For each foreground pixel, we will check if there are less than half the number of foreground points surrounding it. If these points are not connected, then this central foreground pixel is an isolated noise pixel and will be removed from foreground mask; if there are more than half the number of foreground points surrounding it, and if these points are connected, then this central point is confirmed to be a foreground pixel. Under these circumstances, we need to fill in gaps between this point and surrounding connected foreground point within the current window.

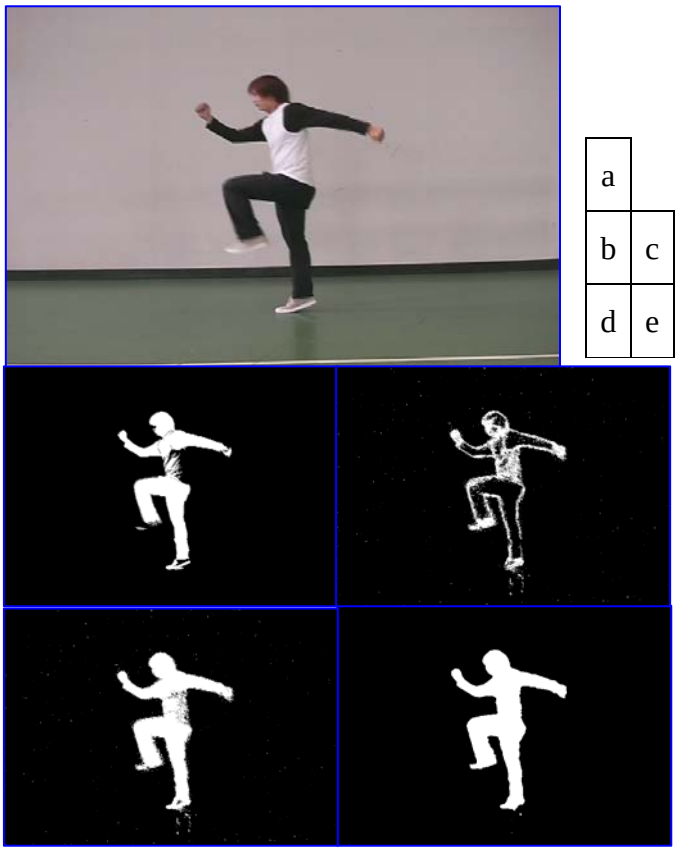


Figure 3-4 Background subtraction (a) the original image (b) pixel-level image (c)frame difference (d) frame-level (e) region level

Background updating

Based on the background estimation results, the background is updated as follows: If $x_{ij,t}$ matches h th distribution, then the parameters of the h th distribution are updated as follows:

$$\mu_{ij,h,t} = (1-\rho)\mu_{ij,h,t-1} + \rho x_{ij,t} \quad (3.4)$$

$$\sigma^2_{j,h,t} = (1-\rho)\sigma^2_{j,h,t-1} + \rho(x_{ij,t} - \mu_{ij,h,t})^t(x_{ij,t} - \mu_{ij,h,t}) \quad (3.5)$$

where ρ is the learning rate, $\omega_{ij,h,t}$ remains unchanged. The parameters for unmatched distributions remain unchanged. But $\omega_{ij,h,t} = (1-\alpha)\omega_{ij,h,t-1}$ will be adjusted as $\omega_{ij,h}$. If $x_{ij,t}$ matches none of the κ distributions, we will check if κ equals to 5, then the least probable distribution is replaced by a distribution where the current value acts as its mean value; if κ is less than 5, a new distribution will be added to background model.

Figure 3-4 shows the result that extracts a person's silhouette who runs in an indoor environment. Figure 3-4(a) shows the result which detects the foreground object in the pixel-level process. Figure 3-4(b) is $D_t \wedge D_{t-1}$ image and Figure 3-4(c) is $MASK = MASK + D_t \wedge D_{t-1}$ which is processed in the frame-level stage. The final result obtained after applying region-level process is shown in Figure 3-4(d).

3.2.2 Human Contour Smoothing

At first, we need to make a definition of human contour.

Definition 3.3

The contour of a human silhouette here is defined as the first layer boundary inside the silhouette.

After extracting the silhouette of the intended human target, we need to detect the contour of the human silhouette. It is a trivial work to be achieved by applying some classical edge detection algorithm, [54]. Unfortunately, through the silhouette extraction stage, there are still noises. Under there circumstances, the detected contour would be saw-toothed effect somewhere. Thus, an additional curve smoothing process is necessary. The most promising candidate would seem to be smoothing with a low-pass Gaussian filter, as has been proposed in many other areas of image analysis. This section will briefly present the basic method and terminology for filtering a curve by Gaussian filter.

The curve to be smoothed is represented as two coordinate functions of a path parameter t :

$$p(t) = (x(t), y(t))^t \qquad (3.6)$$

where

$$x = x(t)$$

$$y = y(t)$$

In order to filter out high frequencies in this curve, we convolve these functions with one dimensional Gaussian $G_\sigma(t)$ of standard deviation σ :

1-D Gaussian Function

$$G_\sigma(t) = \frac{1}{\sigma \cdot \sqrt{2\pi}} e^{-\frac{t^2}{2\varsigma^2}} \qquad (3.7)$$

Then we will have the smoothed curve :

Smoothed curve :

$$P(t) = (X(t), Y(t)) \qquad (3.8)$$

where

$$X(t) = G_\sigma(t) \otimes x(t)$$

$$Y(t) = G_\sigma(t) \otimes y(t)$$

As we use a convolution operator with an one dimensional Gaussian filter, there are two parameters need to be predefined initially :

σ : *the standard deviation*
w : *the convolution window size*

The considerable factors to define these parameters depend on the trade-off between the influence of noise and the number of pixels near by. The high frequency noises

would not be filtered out by applying a smaller σ. And sharp areas would be smoothed by applying a larger σ or w. By experimental-based methodology, we choose the standard deviation σ between 2 to 10, and the convolution window size is predefined as $w = 3 \sim 5\sigma \cdot 2 + 1$. Figure 3-5 shows the results obtained by applying the above mentioned method.

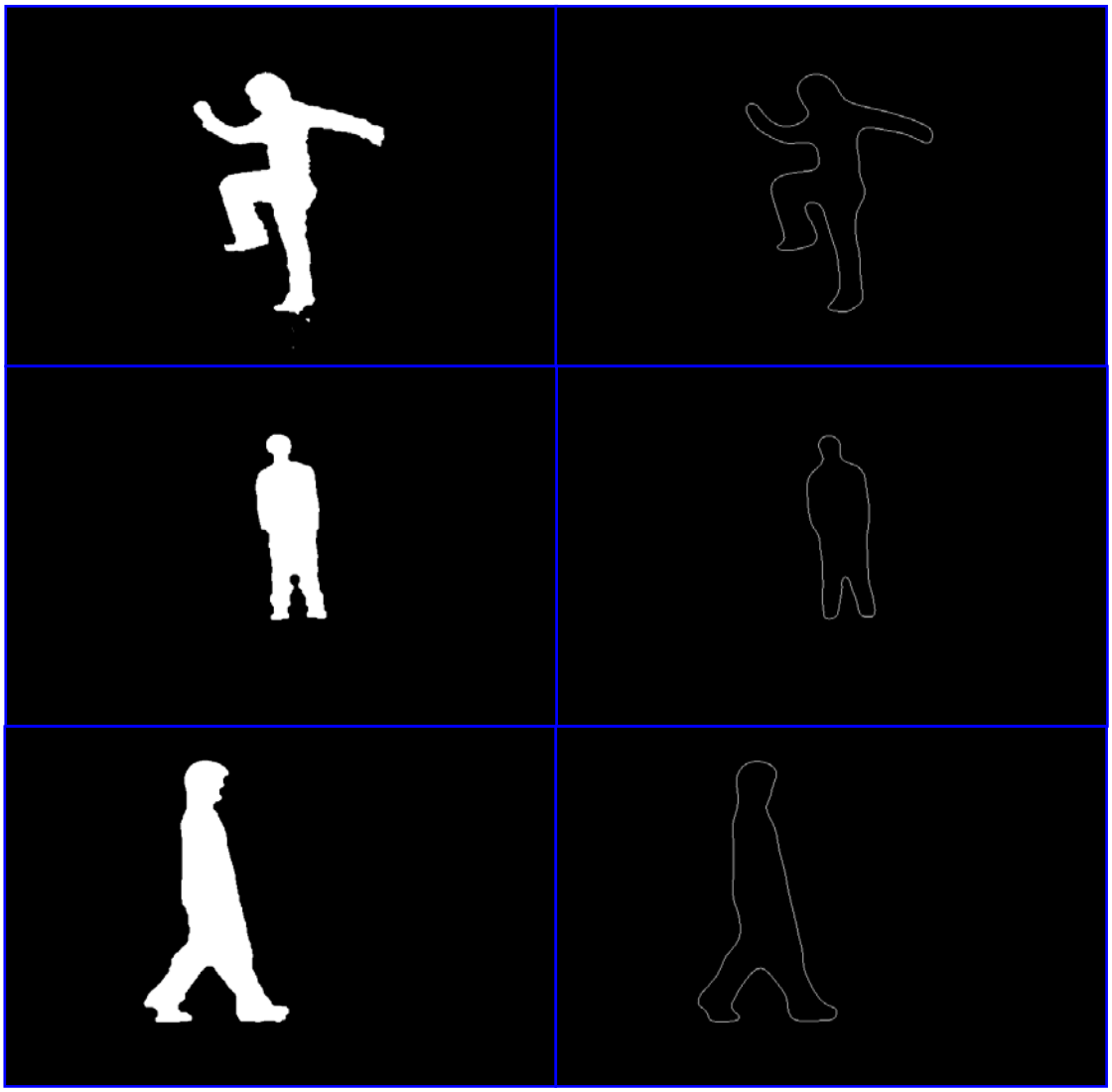
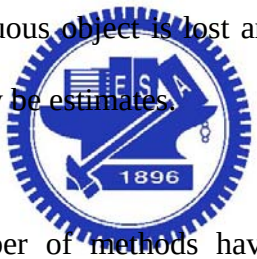


Figure 3-5 Left column shows the noisy extracted silhouette images. Right column shows the smoothed contour of left. ($\sigma = 8$)

3.2.3 Curvature Estimation on Contour

A precise estimation of curvature properties plays an important role in the interpretation of digital binary data such as maps, engineering drawings, and the like. Curvature also is a key notion in the recognition of objects from digital pictures. Specifically maxima, minima, and zero crossings of curvature carry important shape clues.

Digital curvature is computed from a discrete set of points, either representing a digital line or the discrete boundary of some digital objects. The digital set of points is a representation of some continuous pre-digitized object. In the digitization process, exact information on the continuous object is lost and therefore curvature cannot be calculated exactly, but it can only be estimates.



In the literature, a large number of methods have been proposed for curvature estimation. And we know that the ability of a method is reflected by the accuracy and precision of estimation. In the literature on the differential geometry of curves [55], three equivalent formulations of curvature are found. They are respectively based on the orientation of the tangent, the second derivative of the curve considered as a path, or on the local touching circle. For continuous case, the three formulations are equivalent, but not so in digital case. In the following, we would introduce these three formulations briefly.

Definition of curvature

Consider a continuous object X with boundary ∂X. Let $p(s) = (x(s), y(s))^t$ be the length parameterized path following ∂X in a counterclockwise fashion.

By definition, the curvature k of a curve or path p is given by the directional change of the tangent t of p.

Definition 3.4 (Orientation –based continuous curvature)

$$k(s) = \theta'(s) \qquad (3.9)$$

where

$$\theta(s) = \angle(t, x^{+} - axis)$$

Formulated in this way, the sign of $k(s)$ indicates whether the curve locally at s is convex ($k(s) > 0$), or concave ($k(s) < 0$).

Alternatively one can express the curvature in the norm of the second derivative of path x.

Definition 3.5 (Path –based continuous curvature)

$$k(s) = \begin{cases} + \left\| p''(s) \right\| & (\text{contour locally convex}) \\ - \left\| p''(s) \right\| & (\text{contour locally concave}) \end{cases} \qquad (3.10)$$

For an arbitrary (non-path-length) variable u, Definition 3.5 is reformulated into the following equation giving the correct magnitude of curvature as well as the correct sign：

$$k(u) = \frac{x'(u)y''(u) - x''(u)y'(u)}{((x'(u))^2 + (y'(u))^2)^{\frac{3}{2}}} \qquad (3.11)$$

A third definition is derived from the osculating circle touching at $p(s)$, defined as the limiting circle through $p(s-\Delta s), p(s)$, and $p(s+\Delta s)$, when $\Delta s \to 0$. Let $r(s)$ be the radius of the osculating circle at $p(s)$ then：

Definition 3.6 (Osculating circle – based continuous curvature)

$$k(s) = \begin{cases} +\dfrac{1}{r(s)} & (\text{contour locally convex}) \\ -\dfrac{1}{r(s)} & (\text{contour locally concave}) \end{cases} \qquad (3.12)$$

The three definitions are illustrated in Figure 3-6.

Here, for computational convenience, we apply a common method that employs Gaussian filter with convolution operator in Definition 3.5. At every point of the discrete curve, a limited window of fixed size w is predefined initially in the computation. From the point in the window, a local curvature estimation is made. The choice of a fixed window size implies that curvature feature should have compatible level of detail.

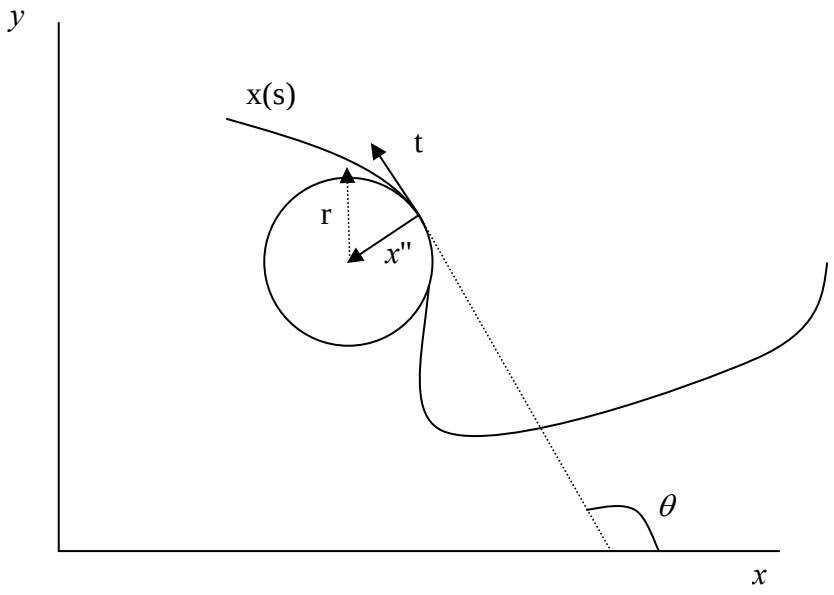


Figure 3-6 This figure illustrates the three different definitions of curvature：(1) the derivative of tangent orientation θ；(2) the norm of the second derivative $x''(s)$；(3) the inverse of the radius r of the osculating circle.

Just as what we have discussed in section 3.3.2, the smoothed contour can be parameterized as：

$$p(t) = (x(t), y(t)) \qquad (3.13)$$

where

$x = x(t)$

$y = y(t)$

Then the curvature at $p(s)$ is estimated as：

$$k(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{((x'(t))^2 + (y'(t))^2)^{\frac{3}{2}}} \qquad (3.14)$$

where

$$x'(t) = G'_\sigma(t) \otimes x(t)$$
$$y'(t) = G'_\sigma(t) \otimes y(t)$$
$$x''(t) = G''_\sigma(t) \otimes x(t)$$
$$y''(t) = G''_\sigma(t) \otimes y(t)$$

And the higher derivatives of Gaussian kernel is formulated as：

the first derivative of Gaussian kernel

$$G'_\sigma(t) = \frac{-t}{\sigma^3\sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}} \qquad (3.15)$$

the second derivative of Gaussian kernel

$$G''_\sigma(t) = \frac{1}{\sigma^3\sqrt{2\pi}}\left(\frac{t^2}{\sigma^2} - 1\right)e^{-\frac{t^2}{2\sigma^2}} \qquad (3.16)$$

3.2.4 Human Silhouette Decomposition

In this section, we will present the main process at the human silhouette decomposition stage of our system. All what we have discussed above in this chapter are sub-functions of this process. At the beginning, we will describe two important rules for achieving our subject to decompose the extracted silhouette.

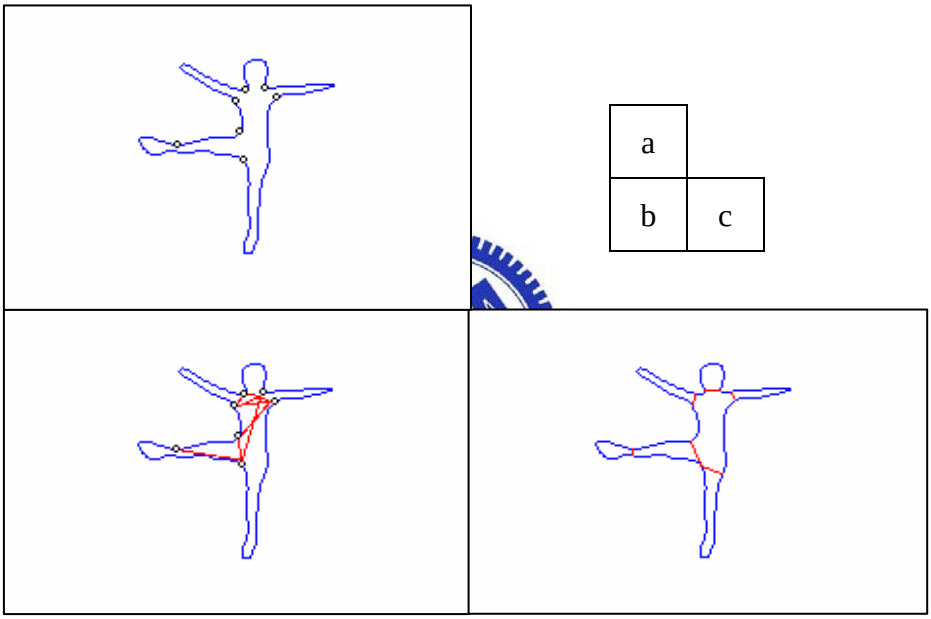


Figure 3-7 (a) points with NCM (b) non-principle decomposition at NCM (c) natural decomposition

Minima rule

According to the Hoffman and Richard's research [56], by human intuition about parts, a segmentation into parts occurs at negative curvature minima (NCM) as shown with small circles in Figure 3-7(a).

Minima rule：

For any silhouette, all negative minima of curvature of its bounding curve are boundaries between parts.

The minima rule constrains cuts to pass through the boundary points it provides, but does not guide the selection of cuts themselves. For example in Figure 3-7(b), the silhouette of a person is decomposed into parts at NCM, but some segmented parts do not correspond to the natural body parts of a person and one of the legs is not detected at all. This example demonstrates that not every pair of NCM forms a natural part, and some parts such as the limbs of animals may be bounded by a NCM and a non-NCM. Therefore, we need to introduce more constraints to achieve unique and natural shape decomposition.

Short cut rule

Singh et al. [58]noted that when boundary points can be joined in more than one way to decompose a silhouette, human vision prefers the partitioning scheme which uses the shortest cuts.

Short cut rule：

Divide silhouettes into parts using the shortest possible cuts.

The requirements of a cut by this rule：

(1) Be a straight line

(2) Cross an axis of local symmetry

(3) Join two points on the outline of a silhouette, such that at least one of the two points has negative curvature

(4) Be the shortest one if there are several possible competing cuts.

Because only one end of a cut is required to lie on a portion of the boundary with negative curvature, this enables us to decompose a shape such as a leg at the right position as shown in Figure 3-7(c). Singh et al.'s scheme restricts the cut to cross a symmetry axis in order to avoid short but undesirable cuts. However, robust computation of symmetry axes is difficult and complex since from their very definitions [48, 59,60] most axes are extremely sensitive to noise.

Human silhouette decomposition

In this thesis, the constraint on the salience of a part is used to replace the second requirement in the short-cut rule in order to avoid the computation of symmetry axes. According to Hoffman and Singh's research [57], there are three factors that affect the salience of a part:

(1) The size of the part relative to the whole object.

(2) The degree to which the part protrudes.

(3) The strength of its boundaries.

Among these three factors, the computation of a part's protrusion (the ratio of the perimeter of the part (excluding the cut) to the length of the cut) is more efficient and robust to noise and partial occlusion of the object. Thus, the protrusion of a part is

employed to evaluate its salience; the salience of a part increases as its protrusion increases.

Therefore, we combine the minima rule, the short-cut rule, and the salience requirement to constrain the conditions of a cut. In the following, we write our decomposition algorithm in a pseudo-code fashion.

Decomposition Algorithm:

Input：

S ：the silhouette of the detected object

C ：the contour of the silhouette

1. for each point $p(t) \in C$
2. estimate the curvature $k(t)$ at $p(t)$

   (here the curvature at each point is collected in the set K)
3. end-for
4. find the negative curvature minima point p_{ncm} with K

   (here the negative curvature minima points is collected in the set NCM)
5. if $NCM \neq \phi$
6. for each point $p_{ncm_i} \in NCM$
7. divide C into C_L and C_R, which start from p_{ncm_i}

   (where $length(C_L) = length(C_R)$, $length(C_L) + length(C_R) = length(C)$, C_L is the left part of C which starts from p_{ncm_i}, and C_R is the right one)

8. find $p_L = \arg\min_{p'} \left\| \overline{p_{ncm_i} p'} \right\|$ s.t. $\frac{\left\| \overset{\frown}{p_{ncm_i} p'} \right\|}{\left\| \overline{p_{ncm_i} p'} \right\|} > T_P,\ \ p' \in C_L,\ \ \overline{p_{ncm_i} p'} \in S$

9. if p_L exists

10. do the cut $\overline{p_{ncm_i} p_l}$

11. end – if

12. find $p_R = \arg\min_{p'} \left\| \overline{p_{ncm_i} p'} \right\|$ s.t. $\frac{\left\| \overset{\frown}{p_{ncm_i} p'} \right\|}{\left\| \overline{p_{ncm_i} p'} \right\|} > T_P,\ \ p' \in C_R,\ \ \overline{p_{ncm_i} p'} \in S$

13. if p_R exists

14. do the cut $\overline{p_{ncm_i} p_R}$

15. end – if

16. end-for

17. end-if

Here, we describe the decomposition algorithm in details： After we extract the silhouette S of the object in an image and detect the smoothed contour C of a silhouette. Then the curvature estimation discussed in 3.2.3 will be applied to calculate the curvature $k(t)$ for each point $p(t) \in C$. The estimated curvature values are collected in the set K. For efficiency and robustness purpose, at the negative-curvature-minima detection stage, we filter out the small magnitude of curvature to avoid parts due to noise and small local deformation. The points p_{ncm_i} with negative curvature minima are collected in the set NCM. For each point $p_{ncm_i} \in NCM$, let p_{half} be the point on C so that p_{ncm_i} and p_{half} divide the

contour C into two curves C_L and C_R, where $length(C_L) = length(C_R)$, $length(C_L) + length(C_R) = length(C)$. Then two cuts $\overline{p_{ncm_i} p_l}$, $\overline{p_{ncm_i} p_R}$ are formed passing through p_{ncm_i}, where $p_L,\ p_R \in C$ are located as follows：

$$p_L = \arg\min_{p'} \left\| \overline{p_{ncm_i} p'} \right\| \quad \text{s.t.} \quad \frac{\left\| \overset{\frown}{p_{ncm_i} p'} \right\|}{\left\| \overline{p_{ncm_i} p'} \right\|} > T_P,\ \ p' \in C_L,\ \ \overline{p_{ncm_i} p'} \in S \qquad (3.16)$$

$$p_R = \arg\min_{p'} \left\| \overline{p_{ncm_i} p'} \right\| \quad \text{s.t.} \quad \frac{\left\| \overset{\frown}{p_{ncm_i} p'} \right\|}{\left\| \overline{p_{ncm_i} p'} \right\|} > T_P,\ \ p' \in C_R,\ \ \overline{p_{ncm_i} p'} \in S \qquad (3.17)$$

$\overset{\frown}{p_{ncm_i} p'}$ is the smaller part of C between p_{ncm_i} and p', $\left\| \overset{\frown}{p_{ncm_i} p'} \right\|$ is the arc length of $\overset{\frown}{p_{ncm_i} p'}$, and $\frac{\left\| \overset{\frown}{p_{ncm_i} p'} \right\|}{\left\| \overline{p_{ncm_i} p'} \right\|}$ is the protrusion of the part bounded by curve $\overset{\frown}{p_{ncm_i} p'}$, and cut $\overline{p_{ncm_i} p'}$.

Eq. (3.16) means that p_L lying on C_L is located such that the cut $\overline{p_{ncm_i} p_l}$ is the shortest one among all cuts sharing the same end p_{ncm_i}, and fits the protrusion threshold T_P. The other end p_R is located in the same way using Eq. (3.17). Figure 3-8 shows an one-step example that decompose the human silhouette into parts, where p_{ncm_i} and p_{half} divide the contour C into two curves C_L and C_R, $\overline{p_{ncm_i} p_l}$ and $\overline{p_{ncm_i} p_R}$ are the cuts satisfy Eqs. (3.16) and (3-17).

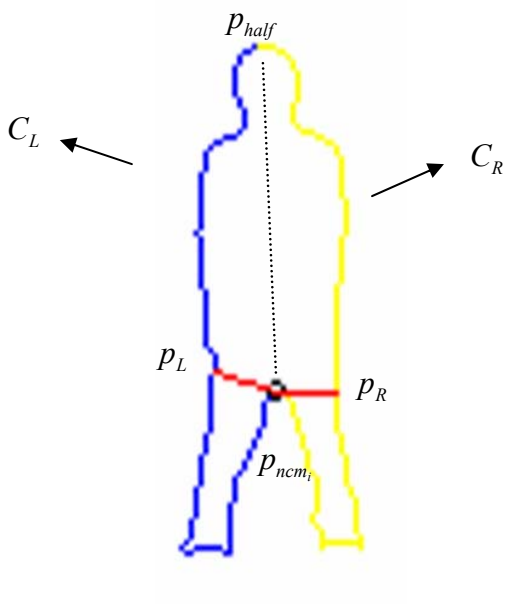
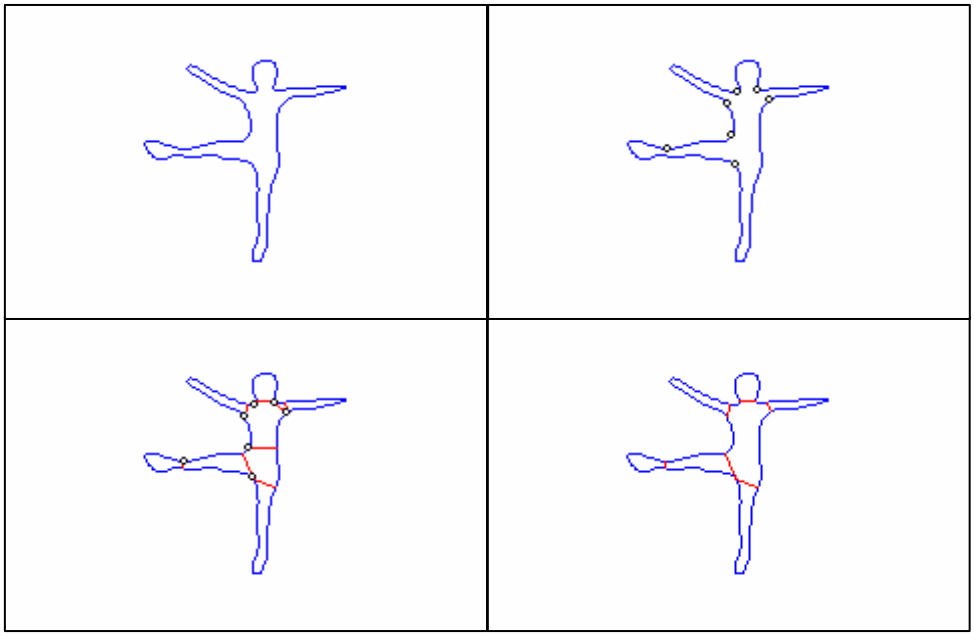


Figure 3-8 Computing the cut passing through the point p_{ncm_i}. p_L and p_R are another ends lie on C_L and C_R.

Over-segmentation alignment

When using Eqs. (3.16) and (3.17) to compute the cuts of a silhouette, they may result in over-segmented parts as shown in Figure 3-9(c). Therefore, a post processing step is needed to merge two over-segmented parts that share a cut into a larger one if this larger part cannot be decomposed into significant subparts using Eqs. (3.16) and (3.17). The order of grouping is from the largest to the smallest parts so that the largest one is selected when several possible competing merges exist. Figure 3-9 illustrates the whole procedure of the shape decomposition algorithm. The procedure stops when no part can be further decomposed into significant parts and no two parts can be merged into a non-decomposable larger part.



a	b
c	d

Figure 3-9 Human silhouette decomposition : (a) smoothed contour (b) NCM estimation (c) decomposition (before alignment) (d) decomposition (after alignment).

Fig. 3-10 shows several example results from the human silhouette decomposition algorithm. These results demonstrate that the algorithm can produce natural part decompositions that are robust to noise and local deformation. Nowadays, it is still an open problem to decompose a shape into a set of perfect subparts without using higher level information. For example, variations in the locations of subparts may occur due to self-occlusion and flexible deformation. There are also missing parts resulting from the inherent difficulty in finding the cut points. All of these will cause non-perfect decomposition. Solving this kind of problems would be our future work. In this thesis,

we focus on the limitations of binary scale and 2-D projection images.

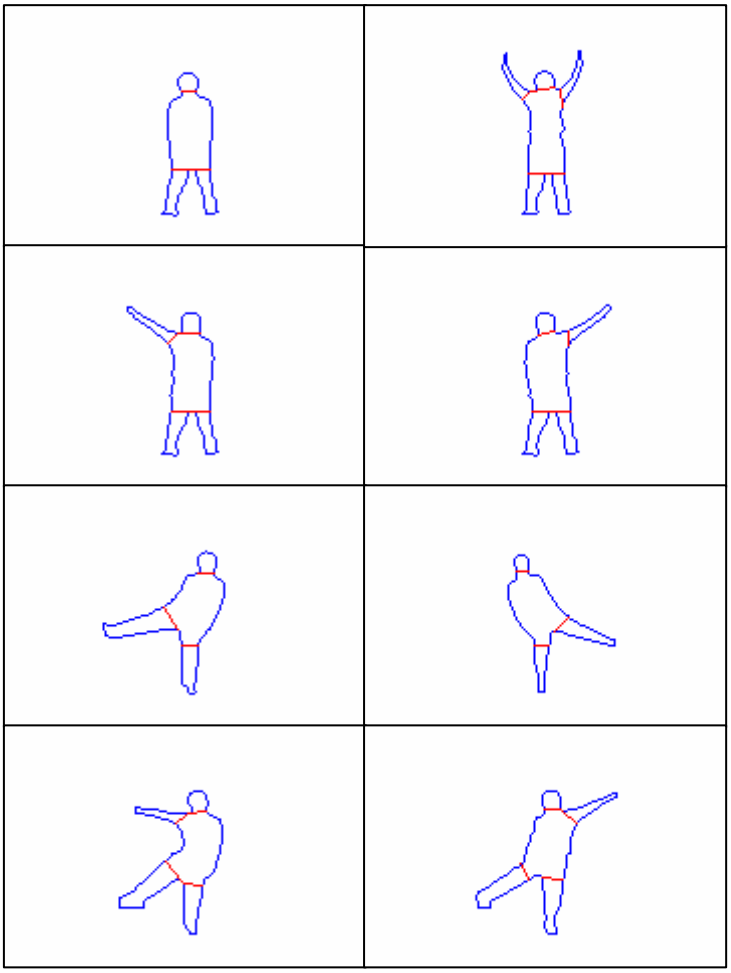


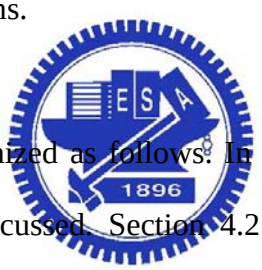
Figure 3-10 Example results of our human silhouette decomposition algorithm.

In summary, this chapter presents a whole human silhouette decomposition algorithm of our system. First, we extract the human silhouette by background subtraction algorithm. Second, the contour of the silhouette is detected and smoothed. Third, the curvature for each point on contour would be estimated. At last, the decomposition algorithm is applied. It can be used to decompose people in an image independent of

their sizes, poses, and articulation. In next chapter, we will introduce an algorithm to identify the parts from this chapter.

4 Human Body Parts Identification

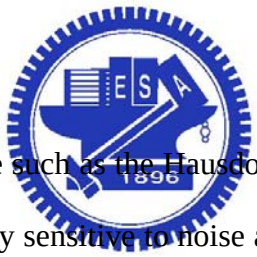
After the decomposition algorithm discussed in chap. 3 is applied in our proposed surveillance system, human body parts identification is an essential and necessary stage for human behavior analysis. In this chapter, we will propose a statistical shape-similarity-based algorithm for this purpose. Although a large number of shape similarity measures have been proposed, most of them are for some special applications, e.g. image retrieval, object recognition…etc. This chapter focuses on designing a similarity measure for deformable shape classification, especially for articulated objects such as humans.



The rest of this chapter is organized as follows. In section 4.1, the related work on shape similarity measure is discussed. Section 4.2 defines the architecture of the human body in our proposed system. Section 4.3 presents a statistical shape-similarity-based algorithm for human body parts identification in a hierarchical fashion. At last, a human body parts estimation for missed parts is presented.

4.1 Related Work on Shape Similarity Measure

The problem of determining the similarity of two shapes has been well-studied in several fields. The design of a similarity measure depends on how a shape is represented. Many global shape descriptors (see reviews in [61,62]) such as Fourier Transform, moments, and eigen shapes have been used to compare two shapes, but they cannot handle occlusion and local deformation such as articulations very well. Therefore, this section does not discuss the similarity measures based on global shape but concentrates instead on those that use local shape primitives, such as points, key points, lines, arcs, axes, or parts. Broad overviews of shape similarity measures can be found in [63, 64, 65].



A point-based similarity measure such as the Hausdorff distance is commonly used to compare two shapes, but it is very sensitive to noise and occlusion. A similar measure that is not as sensitive is the partial Hausdorff distance [66]. This measure can deal with occlusion and clutter very effectively; the measure itself is used to guide the search for an alignment transform in the discrete space. Technically speaking, different transformations such as similarity, affine, or non-rigid transforms can be used for shape alignment. However, the dimension of a non-rigid transformation space is too high to be searched efficiently. A match-based method has been proposed to avoid the search for the transformation in a high dimensional space. An alignment transform is calculated from the matched points, then the similarity measure is calculated as the sum of the residual distances between the corresponding primitives. The common drawback with the above measures is that they have to transform one shape to another before shape comparison because the distance metric is not invariant

under similarity transform.

Various cost functions have been proposed to evaluate the dissimilarity between two contours without aligning them. A cost function weights the similarity of the matched points on the basis of their local properties, such as the difference in the tangent or curvature of the contours at those points. The cost function itself is used to guide the search for the best match. Basri et al. [65] defined the cost function as “elastic energy” needed to deform (stretch or bend) one curve to another. However, the computation of elastic energy (which is defined in terms of curvature) is very sensitive to noise. Other alternatives are possible, for example, such as turning functions [67], arch height functions [68], size functions [69], or functions combining multiple local properties [70,71]. Given a choice of cost functions, several methods such as dynamic programming, gradient descent, or the shortest path algorithm have been employed to find the correspondence between contours that minimize the cost. The main drawback of these methods is their high computational complexity due to searching for correspondences at the point level. Furthermore, none of these cost functions is invariant under scaling and/or rotation of the point data.

Other features such as key points and lines have been used to reduce the computational cost because a digital contour usually consists of much fewer features than of points. Pope and Lowe [72] modeled an object with a graph whose nodes represent the feature values and whose edges represent the spatial arrangement (symmetric, parallel) of the features. Objects are considered similar if their graphs are isomorphic; a similarity metric based on a probability density estimator is used to identify if a shape is an instance of a modeled object. To handle occlusion, partial matching is allowed and the largest mutually compatible matches are found by

constructing an association graph to search for the maximal clique [73]. The main drawback of these methods is that they cannot handle articulated motion because the spatial relationships between features are assumed to be fixed. Moreover, it is difficult to find a coherent set of features that is shared by all possible shapes in a class and that can be extracted reliably.

Part-based representations is a more effective way to handle articulation and occlusion. They have several advantages over other representations such as points, lines, and arcs. First, articulation usually happens at part boundaries, thus, a part-based representation is a more natural and coherent description of articulated shapes. Second, a shape contains fewer aggregate parts than other features. Third, part-based methods find strong support from human vision [74]. The main concerns of a part-based similarity measure are how to decompose a shape into stable parts and how to set up correspondence among them. Parts generally are defined to be convex or nearly convex shapes separated from the rest of the object at concavity extrema [58], or at inflections [75]. One type of approach is to represent shapes as skeletons or graphs and then to use graph matching or qualitative properties such as topology to compare shapes. The main drawback of these approaches to part-based shape analysis is that the shape decomposition is not stable. Since only qualitative properties are used for shape classification, they cannot distinguish two shapes with the same body part structure but different body part shapes and geometric relationships. Zhu and Yuille [51] developed a similarity measure to compare silhouettes based on both the local shapes of parts and the topology but the method can not handle shape degeneration or resolution changes very well. Several curve evolution approaches [76,77,78] have been proposed to model shapes of an object at different scales, but the related similarity measure is sensitive to occlusion and is not invariant under scaling.

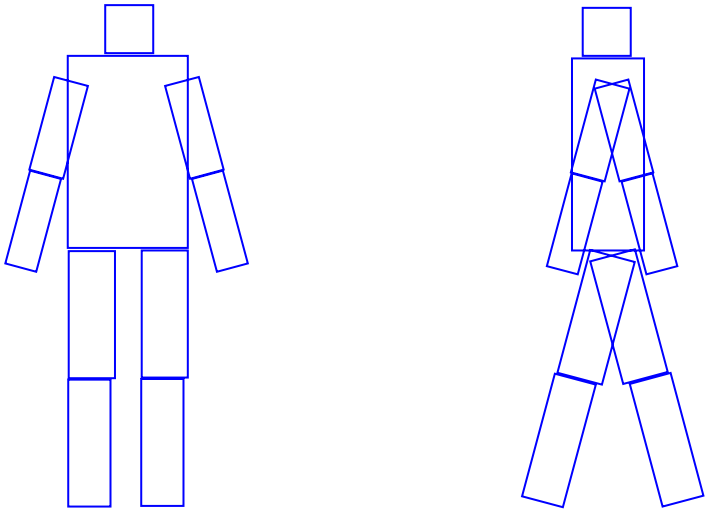
Leung et al. [79,80,81] have proposed a method which combines the intensity pattern and the spatial relationships between the facial features to detect faces from the cluttered environment. However, they do not use the spatial relationship to help detect face features, and no size relationship and recursive procedure is involved in face detection. The main reason is that the facial features have very distinctive patterns and can be detected based on their intensity patterns. In human detection, we rely heavily on the spatial and size relationships to identify the human body parts, because the body parts such as arms and legs do not present very distinctive texture patterns.

In summary, the drawbacks of the above similarity measures are listed as follows：

1. Some depend on the position, size, and orientation of an object.
2. Some cannot support articulation and partial occlusion.
3. Some is not robust to noise, deformation, and blur resulting from image digitization and poor segmentation.

Point-based approaches are time consuming, while feature-based approaches are not stable and cannot handle articulation appropriately. In contrast, the part-based approach is a more promising direction, however, current methods cannot handle shape decomposition errors and shape degeneration very effectively. Above all, the above shape similarity measures cannot deal with large shape variations within a class. They are not appropriate for the purpose of classifying shapes such as those of humans.

4.2 Architecture of a Human Body



(a) (b)

Figure 4-1 Human model : (a) front view, (b)side view

A human body can be represented directly using a 3D model or indirectly using a collection of 2D models corresponding to different views. Since the goal of this thesis is to detect people in an image, 2D models are preferred, because a 2D model can be compared directly with a 2D shape without projecting the 3D model onto the image plane by searching a continuous viewpoint/pose space. The question of how many and which viewpoints to use is an open question and also depends on the application. In the case of pedestrian detection, we found two 2D human body models were sufficient— the front-view and the side-view models as shown in Figure 4-1. The two models share the same body parts. The main differences are the spatial relationships between the parts and the shape of the torso. The views not modeled by these two models are partially absorbed by the probability distributions of the spatial

relationships among the body parts encoded in the human model.

In order to make it more clear, we define the architecture of a human body as shown in Figure 4-2. Each human body model consists of six main body parts：the torso, the head, left arm, right arm, left leg and right leg. For more detail description, the human body model consists of ten body parts：the torso, the head, left upper arm, left lower arm, right upper arm, right lower arm, left upper leg, left lower leg, right upper leg, and right lower leg.

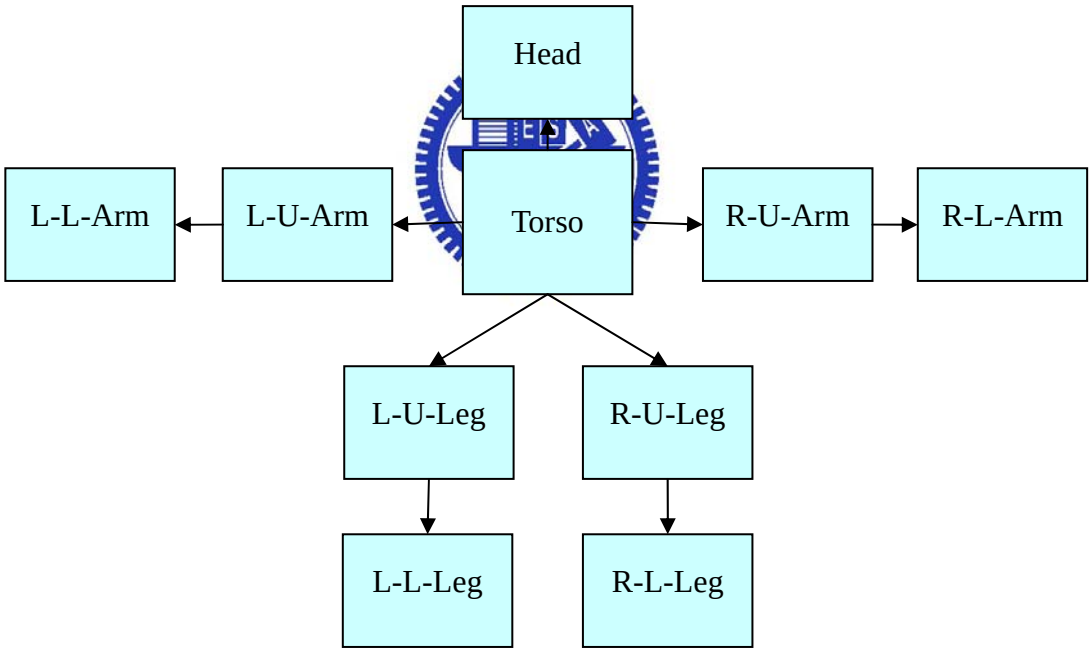


Figure 4-2　Architecture of human body (L/R-U/L-part：L/R means Left/Right, U/L means Upper/Lower)

There are still two definitions need to be introduced.

Definition

The **parent** of a body part is defined as the origin side of the arrow in the architecture of human body. E.g. torso is the parent of head, upper arms, and upper legs.

Definition

The **trunk** of a human object is defined as the one special merged body part which cover the torso and some other body parts, e.g. legs, upper legs, arms, upper arms. Figure 4-3 shows an example of trunk.

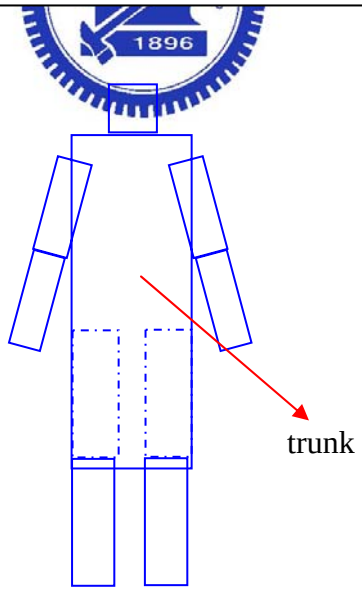


Figure 4-3 An example of the trunk in a human object

4.3 Statistical Shape-Similarity-Based Algorithm

4.3.1 Body Model for Shape-Similarity Measure

For computation convenience, we need to define the body model in a numeric fashion. Figure 4-4 shows an example to describe definition 4.3.

Definition 4.3

A body part B is parameterized with a vector

$B = (x, y, a, l, \theta)$

where

x ：the horizontal coordinate of the joint of B in its parent local coordinate

y ：the vertical coordinate of the joint of B in its parent coordinate

a ：the aspect ratio of B, $a = \frac{w}{l}$, where w is the width of B

l ：length of B

θ ：the orientation of B based on the joint of the body part.

※ the joint is defined as the intersection point between the major axis of B and the cut comes from its parent.

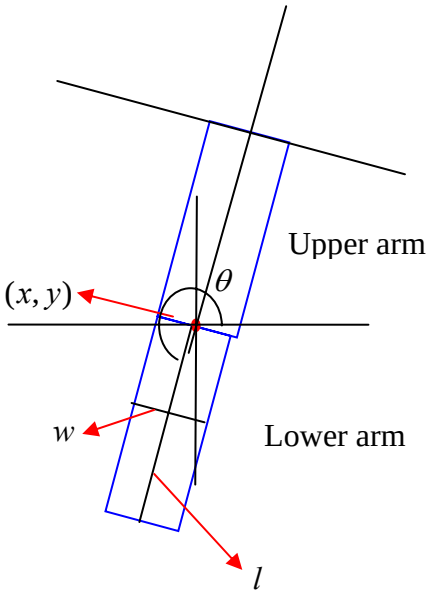


Figure 4-4 Parameterized human body part

The aspect ratio $a = \frac{w}{l}$ is invariant under similarity transforms, and it captures the global shape information of a body part while ignoring small local shape deformations. Thus the aspect ratio is appropriate for the purpose of recognition.

However, the aspect ratio, a, is too ambiguous to be used alone to distinguish different parts. For example, the head and the torso have similar aspect ratios. Therefore, besides aspect ratios of the body parts, the geometric relationships between them are also modeled. There are two more measures we apply：

1. Position (x, y)

The position , (x, y), of a body part is defined as the joint location of the body part, expect the position of torso is defined as its geometric center. (x, y) is the relative position comes from its parent part's coordinate. The joint of a body part is defined as the intersection point between the major axis and the cut comes from its parent.

2. Length l

In section 4.2, we have shown that the architecture of our body model consists of ten sub-parts, or six main parts. Assuming that the ten-sub-parts model is parameterized with vectors B_1, B_2, …, B_{10}, where $B_i = (x_i, y_i, a_i, l_i, \theta_i)$. Here, we define the length ratio matrix $S = \{s_{ij}\},\ i, j = 1,...,10$, where $s_{ij} = \frac{l_i}{l_j}$.

The advantage of locating the position, (x, y), of a body part as the joint point instead of the geometric center of a body part is that the location of the joint become invariant to its parent's orientation. In summary, a body part is parameterized with a vector $B = (x, y, a, l, \theta)$.

Obviously, the aspect ratio, a, of a body part and the length ratio matrix, S, is invariant under rotation and scaling. Because the lengths of the body parts are constrained by the length ratio matrix, S, only the relative positions of the six main body parts needs to be modeled. Let the six relative position of the six main body parts as $X = \{(x_1, y_1),...,(x_6, y_6)\}$, where $(x_1, y_1) = (0,0)$ is the position of torso. To make this vector invariant under rotation and scaling, the coordinates of the joints are represented in a normalized torso coordinate system with the length of the torso normalized to be 1. Then $X = \{(x_1, y_1),...,(x_6, y_6)\}$ would be transform into $U = \{(0,0),...,(u_6, v_6)\}$, where $(u_1, v_1) = \frac{1}{l_1}(x_i, y_i)$, l_1 is the length of the torso.

Finally, we define the body model in our proposed system for the shape-similarity measure which consists of five model matrices in the following：

Definition 4.4 (Body Model)

The body model consists ten sub parts for shape similarity measure is defined as

$M = \{A, S, U, \Theta\}$

where

$A = \{N(\overline{a_1}, \sigma_{a_1}), N(\overline{a_2}, \sigma_{a_2}), ..., N(\overline{a_{10}}, \sigma_{a_{10}})\}$ ：aspect ratio vector

$S = N(\overline{s_{ij}}, \Sigma_s\}$, $i, j = 1,...,10$ ：length ratio matrix, where $s_{ij} = \frac{l_i}{l_j}$

$U = \{(0,0),..., N(\overline{(u_6, v_6)}, \Sigma_{(u_6, v_6)}\}$ ：normalized relative position vector for six main parts.

where $(u_1, v_1) = \frac{1}{l_1}(x_i, y_i)$, l_1 is the length of the torso

$\Theta = \{\theta_1, ..., \theta_{10}\}$ ：the orientation vector

The above probability distributions provide metrics to evaluate the shape, size relationship, and configuration similarities between the detected human object and the body model. Their parameters (means and covariances) are estimated from the measurements provided by Tilley[](see Appendix). Figure 4-5 shows the diagram of the statistical human body model.

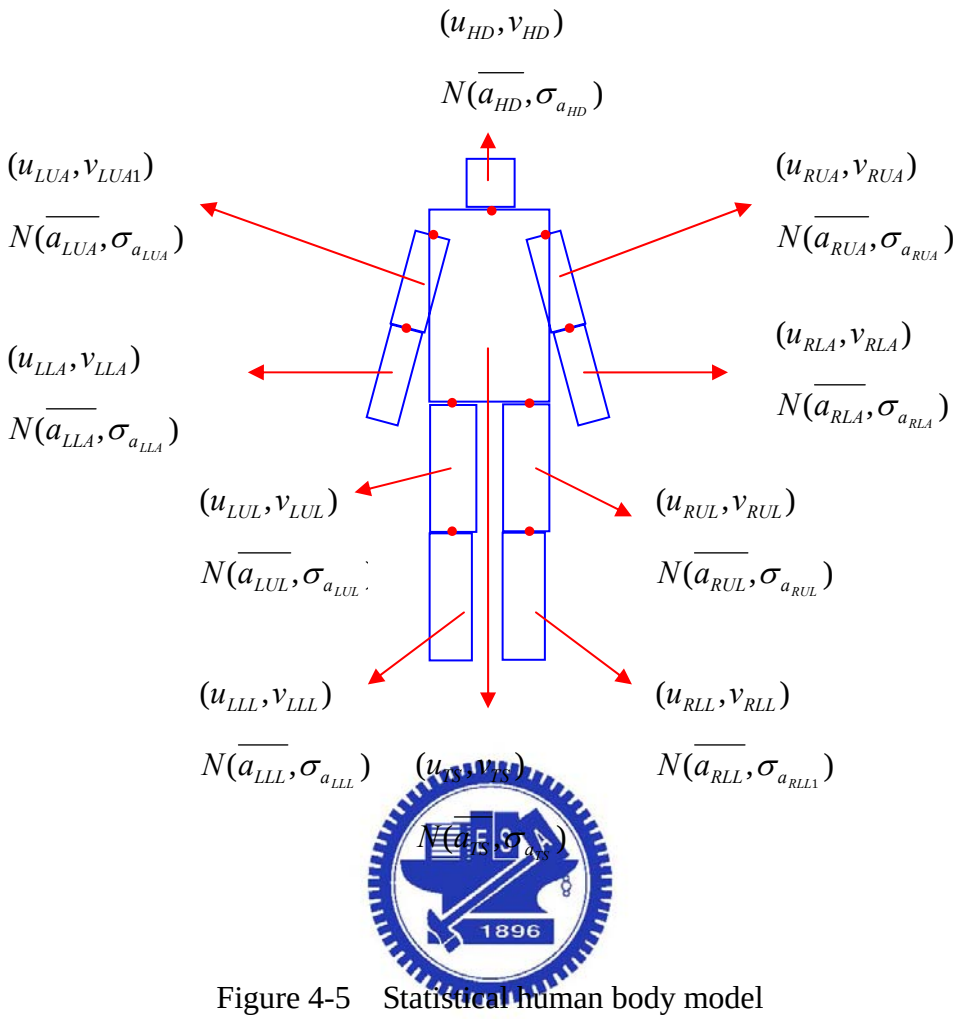


Figure 4-5 Statistical human body model

4.3.2 Moment Function for Local Shape Description

Moments and functions of moments have been used as pattern features in a number of applications to achieve invariant recognition of two-dimensional image patterns. Any model aiming at describing shapes should be invariant under translation, scaling and rotation [83][84]. In this section, the mathematical basis of geometrical moments is presented in the framework of the theory of orthogonal polynomials and the question of how well an entity can be characterized by a finite set of moments is investigated: first in the form of how to rebuild the entity from its moments, then in the evaluation of the reconstruction error.

Definition 4.5 (general definition of moments of order $(p+q)$)

$$m_{pq} = \iint_{\zeta} \psi_{pq}(x,y) f(x,y) dxdy \tag{4.1}$$

where

p, q ：positive integers

f ：the intensity function $\Re \to \{0,1\}$

ζ ：the definition domain of f

ψ_{pq} ：the kernel of the moment function

The definition of Eq (4.1) means that f is projected onto ψ_{pq}. Here, we apply the

geometric moments defined with basis set $x^p y^q$ and the central moments to estimate the geometric center and the orientation of a body part. The $(p+q)^{th}$ two-dimensional geometrical moment $\overline{m_{pq}}$ in a discrete-time image can be defined as follows：

Definition 4.6 (geometric moments of order $(p+q)$)

$$\overline{m_{pq}} = \sum_{1}^{M} \sum_{1}^{N} x_i^p y_j^q f(x_i y_j) \tag{4.2}$$

Where

p , q ：positive integers

f ：the intensity function $\Re \to \{0,1\}$

Definition 4.7 (geometric center)

The geometric center (x_c, y_c) of an object in an image can be estimated as

$$x_c = \frac{\overline{m_{10}}}{m_{00}}$$

$$y_c = \frac{\overline{m_{01}}}{m_{00}} \tag{4.3}$$

By applying the Eq. (4.2), the geometric center of a body part can be estimated by the geometric moments ：$\overline{m_{00}}$, $\overline{m_{10}}$, $\overline{m_{01}}$. The moment of order zero $\overline{m_{00}}$ represents the total intensity of the body part. For an areal entity, this moment is equal to its area. The first order moments $\overline{m_{10}}$ and $\overline{m_{01}}$ provide the intensity about the x -axis and y -axis of the entity respectively.

It is often convenient to evaluate the moments with the origin of the reference system shifted to the intensity centroid of the entity. This transformation makes the moments independent from the position of the entity. The moments computed with respect to the intensity centroid are called **central moments**.

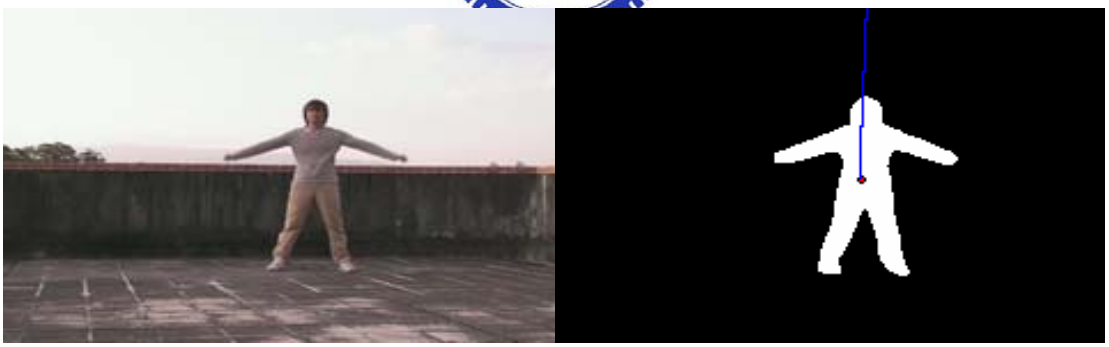


Figure 4-6　The geometric center and orientation of the detected human object.

Definition 4.8 (central moments of order $(p+q)$)

$$\overline{\mu_{pq}} = \sum_{1}^{M} \sum_{1}^{N} (x_i - x_c)^p (y_i - y_c)^q f(x_i y_j) \tag{4.4}$$

where

p , q ：positive integers

f ：the intensity function $\Re \rightarrow \{0,1\}$

The second-order moments are measures of variance of the entity intensity function about the origin. The central moments $\overline{\mu_{20}}$ and $\overline{\mu_{02}}$ assess the variances around the mean. $\overline{\mu_{11}}$ gives the covariance measure. Figure 4-6 shows an example of the estimations of the detected human object.

Definition 4.9 (orientation)

The orientation θ of an object in an image can be estimated as

$$\theta = \frac{1}{2} tan^{-1}\left(\frac{b}{a-c}\right) \tag{4.5}$$

where

$a = \overline{\mu_{20}}$

$b = 2 \cdot \overline{\mu_{00}}$

$c = \overline{\mu_{01}}$

4.3.3 Hierarchical Identification

As what we discussed in chapter 3, after the decomposition process there is a collection of unidentified parts. In this section, a hierarchical statistical-shape-similarity algorithm （HSSS）for human body parts identification is presented. In the following, we will introduce how the algorithm works by applying the body model proposed in section 4.3.1. Figure 4-7 shows the flow chart of the **HSSS** algorithm.

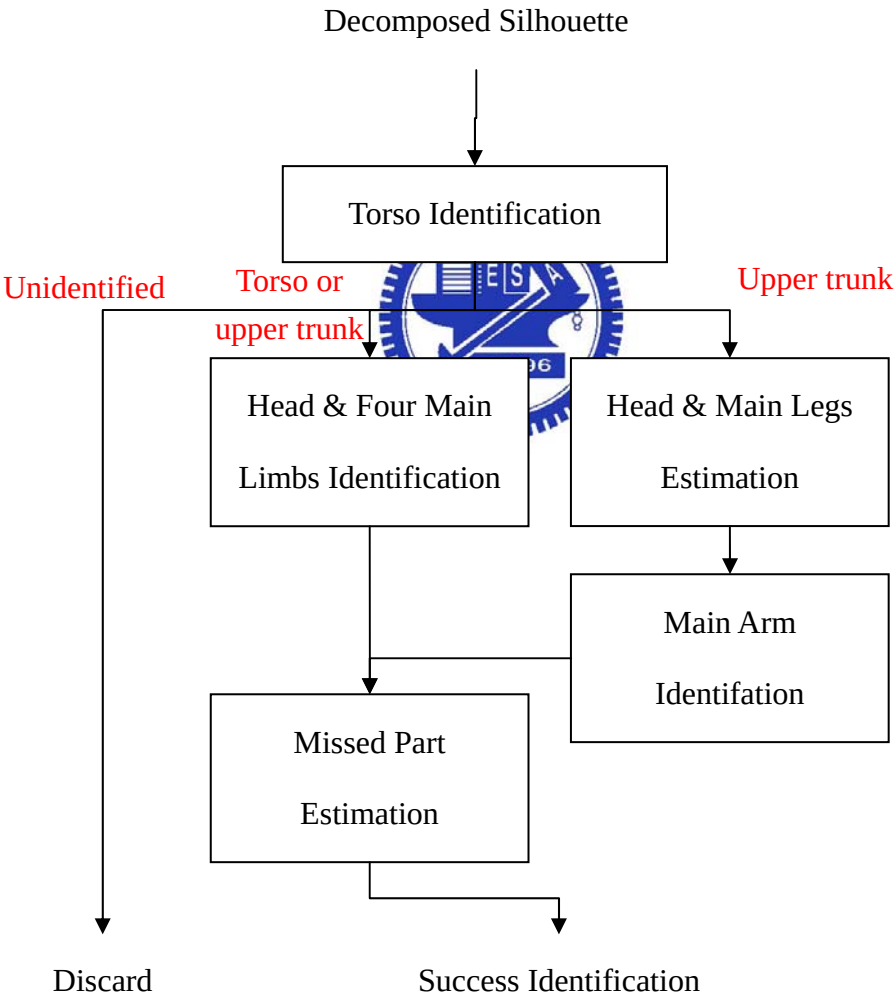


Figure 4-7 Flow chart of the proposed **HSSS** algorithm

Local-Moment-to-Global-Moment Hypothesis

According to our observation, the geometric center of the torso is the nearest one with the global geometric center of the entire human object.

Hypothesis 4.1 (Local-moment-to-global-moment hypothesis)

The geometric center of torso is the nearest one nearby the global geometric center of the human object.

We have observed thousands of human postures. Under most variable posture, the geometric center of torso is always the nearest one nearby the human object's, even bending down. Figure 4-8 shows the results to describe the hypothesis.

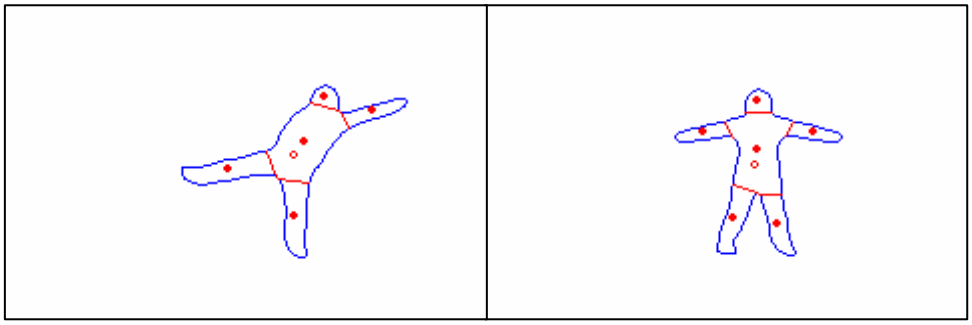


Figure 4-8 The solid points indicate the local geometric center, and the non-solid point indicates the global geometric center.

Torso Identification

According to the architecture of a human object proposed in section 4.2, obviously torso is the only basis of the architecture. Thus, the first step to the identification mission is to classify the torso of the observed human object. If the identification is unsuccessful, we decide to discard this image.

We have presented an important hypothesis to accomplish this mission. Even so, it is still un-robust and weak to classify the torso part. We apply the second measure, **aspect ratio**, to make the identification process more robust. In summary, if an unidentified decomposed part satisfies the conditions below, the system will identify the part as "torso" :

Torso identification conditions：

1. $C_j = \min_i d(C_i, C_{global})$

   where

   C_i ：the geometric center of unidentified decomposed part i.

   C_{global} ：the global geometric center of the human object

2. $a_j \sim N(\overline{a_{torso}}, \Sigma_{a_{torso}})$ ：aspect ratio measure

The identification result will be in just three kind of conditions：

1. Unidentified

   If no parts fit the conditions, the system cannot identifies, and the image is discarded.

2. Torso or upper trunk

3. Truck

   In this condition, we assume that the legs is occluded. Thus, a legs estimation algorithm will be applied. The algorithm will be introduced in section 4.3.4.

Head and Four Main Limbs Identification

After the successful identification of torso, we need to go forward to identify other parts. In order to achieve this purpose, three measures have been introduced in the sections 4.3.1：normalized relative coordinate, aspect ratio, and length ratio. If an unidentified decomposed part satisfies the distributions bellow, then it will be classified as the corresponding part.：

Parts identification conditions：

1. $(u_j, v_j) \sim N((\overline{u_1, v_1}), \Sigma_{(u_1, v_1)})$ ：normalized relative coordinate
2. $a_j \sim N(\overline{a_i}, \Sigma_{a_i})$ ：aspect ratio measure
3. $s_{jj} \sim N(\overline{s_{ji}}, \Sigma_{s_{ji}})$ ：length ratio

In this section, we have presented the first iteration of our **HSSS** algorithm. In this iteration, the system identifies the human body parts based on the probability

distributions model presented in section 4.3.1. After this process, some body parts may not be identified. Later, we will introduce the missed part estimation algorithm in section 4.3.4. Figure 4-9 shows the results of the first iteration of **HSSS** algorithm.

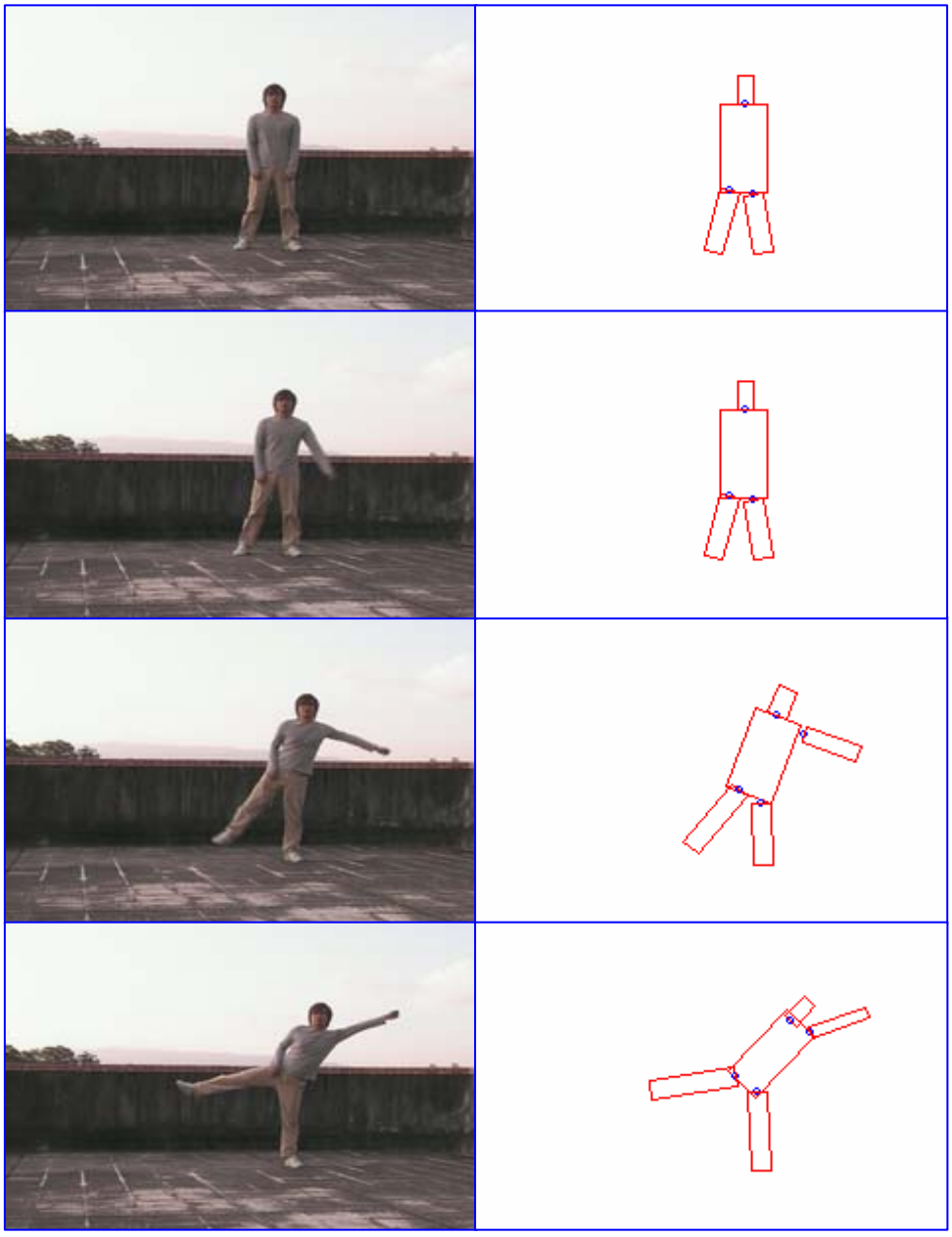


Figure 4-9 First iteration of proposed **HSSS** algorithm

4.3.4 Missed Human Body Parts Estimation

In this section, we will present the last process for the HSSS algorithm. The last iteration is to estimate the parameters of the unidentified body parts. This is done by applying the parameters from the body model proposed in section 4.3.1 and the identified body parts. The parameter vectors of the missed body parts is still be modeled as $B_i = (x_i, y_i, a_i, l_i, \theta_i)$.

The parameters (x_i, y_i), a_i, l_i can be trivially estimated from the statistical body model and the identified torso body vector.

Missed body parts $B_i = (x_i, y_i, a_i, l_i, \theta_i)$ **estimation**：

1. $(x_i, y_i) = \overline{(u_i, v_i)}$
2. $a_i = \overline{a_i}$
3. $l_i = \overline{s_{i(torso)}} \cdot l_{torso}$

The orientations θ_i of the missed body parts cannot be predicted from the model and the identified parts, because the orientation relationships between the body parts are not encoded in the human model. This is solved in the second iteration of the HSSS algorithm by aligning the predicted body part with the contour of the detected human object. The procedure of the alignment is as follows. For each missed body part fi, run Steps 1 to 2:

Steps of orientation θ_i estimation for missed body part：

4. $\theta_i = \theta_{torso}$
5. Let O be the rectangle rendered by the estimated parameters of B_I.

   $\theta_i = arg\max_{\theta} N(O \cap E)$

   where

   E ：the contour of the detected human object

   $N(s)$ ：the number of points in the set s

To make the algorithm having more robustness, other cues such as stereo, motion, and the intensity pattern can be used to constrain the search of the body parts to be within the region of similar attributes. Figure 4-10 shows the results by applying the second iteration of the **HSSS** algorithm.

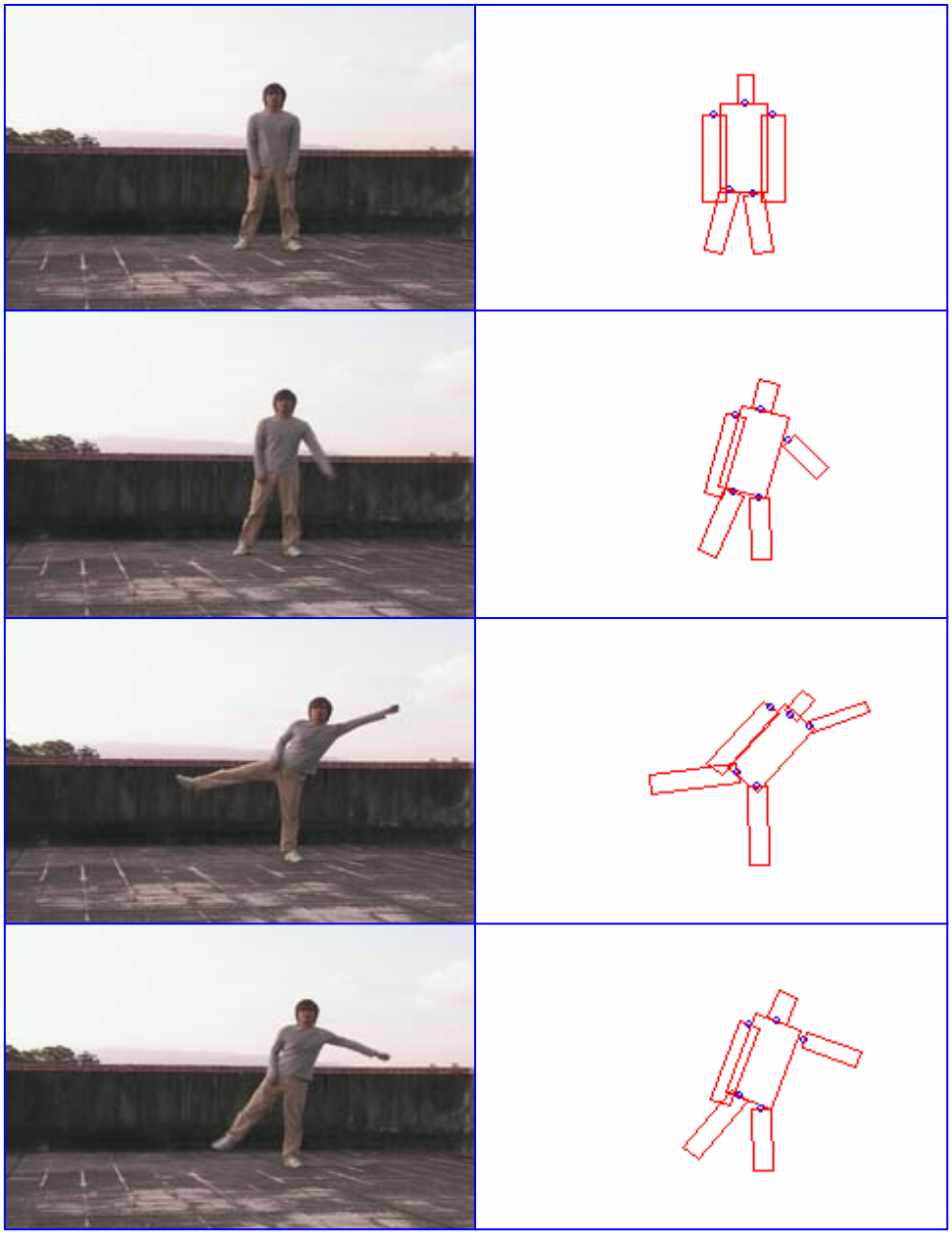


Figure 4-10 The results after second iteration of the proposed HSSS algorithm.

5 Concluding Remark

In this thesis, we have developed a human-silhouette-based visual surveillance system for the human behavior analysis. There are two main contributions have been proposed：

Computational human silhouette decomposition：

In this part, we employ a combination of shape analysis and geometry analysis on a human object's silhouette and contour. By applying efficient computational scheme, we successfully reduce the computation time for the real time processing purpose. The proposed decomposition algorithm is based on the human cognition, and makes the decomposed human body parts closer to the corresponding natural body parts.

Robust human body parts identification：

In this part, we have proposed a robust and effective algorithm to accomplish the human body parts identification task. We name it the Hierarchical Statistical-Shape-Similarity algorithm (HSSS). It runs at two fast passes and significantly identifies the human body parts of a detected human object in many postures under rotation and scaling invariant.

In summary, by the robustness and the efficiency power of our algorithms, they can work in a real-time visual surveillance system. Our system runs at 20~25Hz for 240 x 160 resolution images on a single Pentium-M 1600Mhz PC.

Appendix: Parameters of the Statistical Human Model

In section 4.3.1, we have proposed an statistical human model for human body parts identification. The parameters of the human body model are estimated based on a large quantity of data accumulated over more than 40 years by Henry Dreyfurs, associates and published by Tilley. Tilley provides both the body measurements of people at different ages and the clothing corrections. The following tables list the parameters used in this thesis:

Table 1 Body parts index names

TS	HD	AR	LG	UT	LT	BD	U	L	LA	LL	RA	RL
torso	head	arm	leg	upper trunk	lower trunk	body	upper	lower	left arm	left leg	right arm	right leg

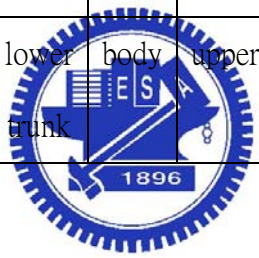


Table 2 The means and the standard deviations of the aspect ratio

		TS	HD	AR		LG		UT	LT	BD
				U	L	U	L			
front view	$\bar{a}$	.61	.78	.25 .12		.25 .08		.92	.43	.30
	σ_a	.10	.09	.12 ..05	.13 .06	.12 .05	.13 .05	.08	.10	.08
side view	$\bar{a}$	.45	.78	.25 .12		.25 .08		.73	.22	.26
	σ_a	.11	.09	.12 ..05	.13 .06	.12 .05	.13 .05	.09	.11	.09

Table 3　The means of length ratios

	TS	HD	AR	LG	UT	LT	BD
TS	1.0	.52	.95	1.47	1.0	1.47	3.0
HD	1.92	1.0	1.83	2.84	1.92	2.84	5.76
AR	1.05	.55	1.0	1.55	1.05	1.55	3.16
LG	.68	.36	.66	1.0	.68	1.0	2.04
UT	1.0	.52	.95	1.47	1.0	1.47	3.0
LT	.68	.36	.66	1.0	.68	1.0	2.04
BD	.33	.18	.32	.49	.33	.49	1.0

Table 4　The standard deviations of the length ratios

	TS	HD	AR	LG	UT	LT	BD
TS	0	.05	.05	.09	.01	.09	.08
HD	.18	0	.21	.36	.18	.36	.53
AR	.05	.07	0	.04	.05	.04	.10
LG	.04	.06	.02	0	.04	.01	.10
UT	.01	.05	.05	.09	0	.09	.08
LT	.04	.06	.02	.01	.04	0	.10
BD	.01	.02	.01	.02	.01	.02	0

Table 5 The means of the coordinates of the body parts in the normalized torso coordinate system

		TS	HD	LA	LL	RA	RL	UT	LT
front view	$\bar{x}$	0	0	-.31	-.163	.163	.31	0	0
	$\bar{y}$	0	.5	.353	-.472	-.472	.353	0	-.5
side view	$\bar{x}$	0	0	0	0	0	0	0	0
	$\bar{y}$	0	.5	.353	-.472	-.472	.353	0	-.5

Table 6 The covariance of the coordinates of the body parts in the normalized torso coordinate system (front view)

	$\times 10^{-1}$									
$\overline{x_{HD}}$	.34	.11	.08	.03	.04	.13	-.06	.15	-.09	.04
$\overline{y_{HD}}$	.11	.92	.09	.56	.13	.91	-.15	.92	-.10	.55
$\overline{x_{LA}}$	.08	.09	.30	.07	.16	.19	-.17	.14	-.33	.06
$\overline{y_{LA}}$	.03	.56	.07	.95	.05	1.13	-.03	1.01	-.07	.91
$\overline{x_{LL}}$	.04	.13	.16	.05	.39	.23	-.35	.27	-.16	.05
$\overline{y_{LL}}$	.13	.91	.19	1.13	.23	1.82	-.27	1.89	-.19	1.13
$\overline{x_{RL}}$	-.06	-.15	-.17	-.03	-.35	-.27	.41	-.32	.17	-.06
$\overline{y_{RL}}$	.15	.92	.14	1.01	.27	1.89	-.32	1.27	-.20	1.23
$\overline{x_{RA}}$	-.09	-.10	-.33	-.07	-.16	-.19	.17	-.20	.35	-.08
$\overline{y_{RA}}$	.04	.55	.06	.91	.05	1.13	-.06	1.23	-.08	.90

Table 7 The covariance of the coordinates of the body parts in the normalized torso coordinate system (side view)

	$\times 10^{-1}$									
$\overline{x_{HD}}$	.18	.10	.06	.02	.02	.09	-.08	.13	-.07	.02
$\overline{y_{HD}}$	.10	.85	.07	.45	.11	.87	-.13	.88	-.19	.50
$\overline{x_{LA}}$	.06	.07	.13	.08	.14	.15	-.11	.17	-.25	.07
$\overline{y_{LA}}$	.02	.45	.08	.91	.06	1.11	-.04	1.06	-.04	.83
$\overline{x_{LL}}$	.02	.11	.14	.06	.20	.13	-.30	.25	-.13	.06
$\overline{y_{LL}}$	.09	.87	.15	1.11	.13	1.67	-.25	1.71	-.12	1.02
$\overline{x_{RL}}$	-.08	-.13	-.11	-.04	-.30	-.25	.22	-.21	.15	-.05
$\overline{y_{RL}}$	.13	.88	.17	1.06	.25	1.71	-.21	1.18	-.21	1.14
$\overline{x_{RA}}$	-.07	-.19	-.25	-.04	.13	-.12	.15	-.21	.16	-.07
$\overline{y_{RA}}$	.02	.50	.07	.83	.06	1.02	-.05	1.14	-.07	.92

References

[1] 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, March 2001.

[2] D. M. Gavrila, “The visual analysis of human movement: A survey,” Computer Vision and Image Understanding, Vol. 73. No. 1, pp. 82-98, Jan. 1999.

[3] A. Pentland, “Looking at people: Sensing for Ubiquitous and wearable computing,” IEEE Tran. on PAMI, Vol. 22, No. 1, pp. 107- 119, Jan. 2000.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis, “W^4: Real-time surveillance of people and their activities,” IEEE Tran. on PAMI, Vol. 22, No. 8, pp. 809-830, Aug. 2000.

[5] T. N. Tan, G. D. Sullivan, and K. D. Baker, “Model-based localization and recognition of road vehicles,” Int. J. of Computer Vision, vol. 27, no. 1, pp. 5-25, March 1998.

[6] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” IEEE Tran. on PAMI, Vol. 19, No. 7, July 1997.

[7] T. Olson and F. Brill, “Moving object detection and event recognition algorithms for smart cameras,” in Proc. DARPA Image Understanding Workshop, pp. 159–175, 1997.

[8] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, “Moving target classification and tracking from real-time video,” in Proc. IEEE Workshop on Applications of Computer Vision, pp. 8–14, 1998.

[9] S. J. Maybank and T. N. Tan, “Special section on visual surveillance-introduction,” Int. J. of Computer Vision, vol. 37, no. 2, pp. 173–174, 2000.

[10] R. T. Collins, A. J. Lipton, and T. Kanade, “Introduction to the special section on video surveillance,” IEEE Trans. on PAMI, vol. 22, pp. 745–746, Aug. 2000.

[11] A. Hilton and P. Fua, “Foreword: modeling people toward vision-based understanding of a person’s shape, appearance, and movement,” Computer Vision and Image Understanding, vol. 81, no. 3, pp. 227–230, 2001.

[12] C. Regazzoni and V. Ramesh, “Special issue on video communications, processing, and understanding for third generation surveillance systems,” Proc. IEEE, vol. 89, pp. 1355–1367, Oct. 2001.

[13] S. G. Gong and H. Buxton, “Editorial: understanding visual behavior,” Image and Vision Computing, vol. 20, no. 12, pp. 825–826, 2002.

[14] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L.Wixson, “A system for video surveillance and monitoring,” Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., CMU-RI-TR-00-12, 2000.

[15] N. Friedman and S. Russell, “Image segmentation in video sequences: a probabilistic approach,” in Proc. 13th Conf. Uncertainty in Artificial Intelligence, pp. 175-181, 1997.

[16] M. Köhle, D. Merkl, and J. Kastner, “Clinical gait analysis by neural networks: Issues and experiences,” in Proc. IEEE Symp. Computer-Based Medical System, pp. 138–143, 1997.

[17] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, “Using adaptive tracking to classify and monitor activities in a site,” in Proc. IEEE Int. Conf. on CVPR, pp. 22–29, 1998.

[18] C. Ridder, O. Munkelt, and H. Kirchner, “Adaptive background estimation and foreground detection using Kalman-filtering,” in Proc. Int. Conf. Recent Advances in Mechatronics, 1995, pp. 193–199.

[19] S. 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, 2000.

[20] D. Meyer, J. Denzler, and H. Niemann, “Model based extraction of articulated objects in image sequences for gait analysis,” in Proc. IEEE Int. Conf. on Image Processing, pp. 78–81, 1998.

[21] R. Cutler and L. S. Davis, “Robust real-time periodic motion detection, analysis, and applications,” IEEE Trans. on PAMI, vol. 22, pp. 781–796, Aug. 2000.

[22] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” in Proc. European Conf. Computer Vision, pp. 343–356, 1996.

[23] I. A. Karaulova, P. M. Hall, and A. D. Marshall, “A hierarchical model of dynamics for tracking people with a single video camera,” in Proc. British Machine Vision Conf., pp. 262–352, 2000.

[24] S. Ju, M. Black, and Y. Yaccob, “Cardboard people: a parameterized model of articulated image motion,” in Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp. 38–44, 1996.

[25] Q. Delamarre and O. Faugeras, “3D articulated models and multi-view tracking with physical forces,” Computer Vision and Image Understanding, vol. 81, no. 3, pp. 328–357, 2001.

[26] R. Plankers and P. Fua, “Articulated soft objects for video-based body modeling,” in Proc. IEEE Int. Conf. on Computer Vision, pp. 394–401, 2001.

[27] K. Takahashi, S. Seki, H. Kojima, and R. Oka, “Recognition of dexterous manipulations from time varying images,” in Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 23–28, 1994.

[28] A. F. Bobick and A. D. Wilson, “A state-based technique to the representation and recognition of gesture,” IEEE Trans. on PAMI, vol. 19, pp. 1325–1337, Dec. 1997.

[29] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden Markov models for complex action recognition,” in Proc. IEEE Conf. on CVPR, pp. 994–999, 1997.

[30] M. Yang and N. Ahuja, “Extraction and classification of visual motion pattern recognition,” in Proc. IEEE Conf. on CVPR, pp. 892–897, 1998.

[31] Y. A. Ivanov and A. F. Boblic, “Recognition of visual activities and interactions by stochastic parsing,” IEEE Trans. on PAMI, vol. 22, pp. 852–872, Aug. 2000.

[32] T. Wada and T. Matsuyama, “Multi-object behavior recognition by event driven selective attention method,” IEEE Trans.on PAMI, vol. 22, pp. 873–887, Aug. 2000.

[33] G. Deng, L. W. Cahill, “An adaptive Gaussian filter for noise reduction and edge detection,” In Proc. IEEE Conf. Nuclear Science Symposium and Medical Imaging , pp. 1615 – 1619, 1993.

[34] A. L. Yuille and T. A. Poggio, “Scaling theorems for zero-crossings,” IEEE Trans. PAMI, vol. 8, pp. 15–25, Jan. 1986.

[35] D. J. Williams and M. Shah, “Edge contours using multiple scales,” Computer Vision, Graph, Image Processing, vol. 51, pp. 256–274, 1990.

[36] I.A. Kakadiaris, D. Metaxas, “Model-Based Estimation of 3D Human Motion with Occlusion Based on Active Multi-Viewpoint Selection,” in Proc. IEEE Conf. on CVPR, pp. 81-87, June, 1996.

[37] D.A. Forsyth, Fleck, “Body plans,” in Proc. IEEE Conf. on CVPR, pp. 678 – 683, June 1997.

[38] M.K. Leung, Yee Hong Yang, “A model based approach to labelling human body outlines,” in Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 57-62, Nov. 1994.

[39] Kakadiaris, I.A.; Metaxas, D.; Bajcsy, R.; “Active motion-based segmentation of human body outlines,” in Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 50-56, Nov. 1994.

[40] D.M. Gavrila, L.S. Davis, “3-D model-based tracking of humans in action: a multi-view approach,” in Proc. IEEE Conf. on CVPR, pp. 73-80, June 1996.

[41] C. Bregler, j. Malik, J, ”Tracking people with twists and exponential maps,” in Proc. IEEE Conf. on CVPR, PP. 8-15, June 1998.

[42] M. Yamamoto, A. Sato, S. Kawada, T. Kondo, y. Osaki, ”Incremental tracking of human actions from multiple views,” in Proc. IEEE Conf. on CVPR, PP. 2-7, June 1998.

[43] S.Wachter and H. H. Nagel, “Tracking Persons in Monocular Image Sequence,” Computer Vision and Image Understanding, Vol. 74, No. 3, pp. 174-192, 1999.

[44] D. Hogg, “Model-based Vision: a Program to See a Walking Person,” Image and Vision computing, Vol. 1, No. 1, pp. 5-20, 1983.

[45] J. Deutscher, B. North, B. Bascle, A. Blake, “Tracking through Singularities and Discontinuities Random Sampling,” in Proc. IEEE Conf. on Computer Vision, pp. 1144-1149, 1999.

[46] C. Sminchisescu, B. Triggs, ”Covariance scaled sampling for monocular 3D body tracking,” in Proc. IEEE Conf. on CVPR, PP. 447-454, 2001.

[47] R. Bowden, T. A. Mitchell, M. Sarhadi, “Reconstructing 3D Pose and Motion from a Single Camera View,” BMVC, pp. 904-913, 1998.

[48] D. Marr and H.K. Nishihara, “Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes,” in Proc. of the Royal Society of London, B-200, pp. 269-294, 1978.

[49] A. Baumberg, D. Hogg, “Learning Flexible Models from Image Sequences,” Proc. European Conf. on Computer Vision, pp. 299-308, 1994.

[50] D.M. Gavrila and V. Philomin, “Real-Time Object Detection for “Smart” Vehicles,” in Proc. IEEE Int. Conf. on Computer Vision, Corfu, Greece, 1999.

[51] S. C. Zhu and A. L. Yuille, “Forms: A Flexible Object Recognition and Modeling System,” Int. J. of Computer Vision, Vol. 20, No.3, 1996.

[52] C. Papageorgiou, T. Poggio, “Trainable pedestrian detection,” in Proc. IEEE Int. Conf. on Image Processing, Vol. 4, PP. 35 – 39, Oct. 1999.

[53] C. Stauffer, W.E.L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proc. IEEE Conf. on CVPR, Vol. 2, 23-25, June 1999.

[54] J. Canny, “A computational approach for edge detection,” IEEE Trans. On PAMI, vol. 8, no 6 pp. 679-698, 1986.

[55] D. J. Struik, “Lectures on classical differential geometry,” Dover New York, 1984.

[56] D.D. Hoffman,W.A. Richards, “Parts of Recognition,” Cognition, Vol. 18, pp. 65-96, 1984.

[57] M. Singh, G. D. Seyranian, D. D. Hoffman, “Parsing Silhouettes: the Short-CutRule,” Perception and Psychophysics, Vol. 61, No. 4, pp. 636-660, May 1999.

[58] D.D. Hoffman, W.A. Richards, “Salience of visual parts,” Cognition, Vol. 63, pp. 29-78, 1997.

[59] A. Rosenfeld, “Axial Representation of Shape,” Computer Vision, Graphics, and Image Processing, Vol. 33, pp. 156-173, 1986.

[60] J. Ponce, “On Characterizing Ribbons and Finding Skewed Symmetries,” Computer Vision, Graphics, and Image Processing, 52(3), pp. 328-340, 1990.

[61] R. O. Duda, and P. E. Hart. Pattern Classification and Scene Analysis, Wiley-Interscience Publication, John Wiley and Sons, Inc., 1973.

[62] R. Bolles and R. Cain. “Recognizing and Locating Partially Visible Objects: the Local-Feature-Focus Method,” Int. J. of Robotics Research, Vol. 3, No. 1, pp. 57–82, 1982.

[63] D. Mumford, “Mathematical Theories of Shape: Do They Model Perception?,” SPIE, vol. 1570, Geometric Methods in Computer Vision, pp. 2–10, 1991.

[64] S. Loncaric, “A Survey of Shape Analysis Techniques,” Pattern Recognition, Vol. 25, pp. 17–23, 1992.

[65] R. Basri, L. Costa, D. Geiger and D. Jacobs. “Determining the Similarity of Deformable Shapes,” IEEE Workshop on Physics Based Modeling in Computer Vision, 135–143, 1995.

[66] D. Huttenlocher, G. Klanderman and W. Rucklidge. Comparing Images Using the Hausdorff Distance. IEEE Trans. on PAMI, Vol. 15, No. 9, pp. 850–863, 1993.

[67] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, J.S.B. Mitchell, “An efficiently computable metric for comparing polygonal shapes,” IEEE Trans. on PAMI, Vol. 13, No. 3, pp. 209 – 216, March 1991.

[68] Y. Lin, J. Dou, and H. Wang. “Contour Shape Description Based on an Arch Height Function,” Pattern Recognition, Vol. 25, pp. 17–23, 1992.

[69] C. Uras, and A. Verri, “Computing Size Functions from Edge Maps.,” Int. J. of Computer Vision, Vol. 23, No. 2, pp. 169–183, 1997.

[70] K. Yoshida, and H. Sakoe, “Online Handwritten Character Recognition for a Personal Computer Systems,” IEEE Trans. on Consumer Electronics, Vol. 28, No. 3, pp. 202–209, 1982.

[71] C. Tappert, “Cursive Script Recognition by Elastic Matching,” IBM J. of Research Development, Vol. 26, No. 6, pp. 765–771, 1982.

[72] A. R. Pope, D. G. Lowe, “Learning object recognition models from images ,” in Proc. IEEE Int. Conf. on Computer Vision, pp. 296- 301, May 1993

[73] M. Koch, R. Kashyap, “Using Polygons to Recognize and Locate Partially Occluded Objects,” IEEE Trans. on PAMI, Vol. 9, pp. 483–494, 1987.

[74] K. Siddiqi, B. B. Kimia, and K. J. Tresness, “Parts of Visual Form: Psychophysical Aspects,” Perception, Vol. 25, No. 4, pp. 399-424, 1996.

[75] F. Mokhtarian, A.K. Mackworth, ”A theory of multiscale, curvature-based shape representation for planar curves,” IEEE Trans. on PAMI, Vol. 14, Issue 8, P.P. 789-805, Aug. 1992.

[76] K. Siddiqi, B. B. Kimia, “Parts of Visual Form: Computational Aspects,” IEEE Trans. PAMI, Vol. 17, No. 3, pp. 239-251, 1995.

[77] L. J. Latecki and R. Lakmper, “Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution,” Computer Vision and Image Understanding, Vol. 73, No. 3, pp. 441-454, 1999.

[78] R. Malladi and J.A. Sethian, “A Unified Approach for Shape Segmentation, Representation, and Recognition,” Report LBL-36069, Lawrence Berkeley Laboratory, University of California, Berkeley, CA, August 1994.

[79] T. K. Leung, M. C. Burl, P. Perona, “Finding faces in cluttered scenes using random labeled graph matching,” In Proc. IEEE Int. Conf. on Computer Vision, pp. 637 – 644, June 1995.

[80] T. K. Leung, M. C. Burl, P. Perona, “Probabilistic affine invariants for recognition,” In Proc. IEEE Int. Conf. on CVPR, pp. 678 – 684, June 1998.

[81] M.C. Burl,M.Weber and P. Perona, “A probabilistic approach to object recognition using local photometry and global geometry,” Proc. European Conf. on Computer Vision, Vol. 2, pp. 628-642, 1998.

[82] M. Sullivan, C. Richards, C. Smith, O. Masoud, N. Papanikolopoulos, “Pedestrian Tracking from a Stationary Camera Using Active Deformabel Models,” Proc. Intelligent Vehicles, pp. 90-95, 1995.

[83] M. K. Hu, “Visual pattern recognition by moments invariants,” IRE Trans on Information Theory, Vol 8, No 1, pp 179-187, 1962.

[84] Rothe I, Susse H, Voss K, “The Method of Normalization to Determine Invariants,” IEEE Trans. on PAMI, vol. 18, No 4, pp 366-376, 1996