Detection of Defined Human Poses for Video Surveillance

Final Report for BTech 451

by

Xu He

Supervisor: Professor Reinhard Klette

Tamaki campus
Department of Computer Science
The University of Auckland
New Zealand
June 2014

Abstract

The project aims at motion detection for video surveillance by using a IPVS camera offered by Compucon New Zealand. Commercial off-the-shelf video surveillance cameras are very capable of recording images in computer hard disks at high resolutions and frame rates. They are also capable of applying motion detection criteria to achieve surveillance objectives while reducing the bandwidth burden for data transmission and recording in computer storage. However the detection criteria are based on change of pixel counts and not on the meaning of images or the environment seen or recorded. This is not accurate and leads to lots of false alarms. Computer data modelling in motion detection comes in at this point. Motion detection is typically used for real-time human detection, human tracking and human activity analysis in video surveillance system.

This is a one-year project, which offered by Compucon New Zealand, carries weights of three courses theached in the University of Auckland. This project is compulsory for a final-year BTech (Information Technology) honors degree student.

Keywords: Human motion detection, Video surveillance, Human tracking, Background subtraction, Raising up hands recognition.

Acknowledgments

I would like to thank my academic supervisor Professor Reinhard Klette for guiding and teaching. I would like to thank Dr Manoharan for giving me the opportunity. I would also like to thank TN Chan for keep pushing me. In the end, I would like to thank Zhengping Wang for providing his code and helping me with silhouette detection.

Xu He
Auckland
October 28, 2014

Contents

Chapter 1

Introduction

This is the final report for BTech 451 project taught at the University of Auckland. This is a one-year project including two semesters, 8-10 hours of work for the first semester, and 16-20 hours of work for the second semester. The project is research oriented on the methodologies of foreground extraction and defined human action analysis (human raised hands) for a real-time video surveillance system.

In this chapter, list the structure of the report and the background of the project.

1.1 Project Overview

Currently, video surveillance systems are applied very common in public places. The main purpose to use a video surveillance system is for security and recording. Many commercial and public places rely on security people for watching the recording. It is not efficient, so automatic security analysis is developing in recent years.

Traditional video surveillance system records all the information but it will consume a huge amount of storage space. Modern commercial video surveillance system are very capable of recording video when human motions are being detected, that is in order to reducing the bandwidth of data transmission and video storage.

However the detection are based on changing of pixels, not on the meaning of the video such as people behavior abnormal in front of camera. The detection results are not accurate and leads to lots of false alarms. At this point, we attempt to modeling a real-time system to take the meaningful defined human motions captured by a surveillance camera.

1.2 Company Information

This project is sponsored by TN Chan from the company Compucon New Zealand.

Figure 1.1: Compucon New Zealand

Compucon New Zealand is a computing system manufacturer and a digital technology system integrator, which was established in 1992, as shown in figure 1.1. It has been 100% New Zealand owned since 2011.

Compucon New Zealand is also part of an International Compute manufacturing group of companies founded in 1989 in Sydney. For more information about this company, please visit following link,

www.compucon.co.nz.

1.3 Data Collection

An IPVS camera will be used for recording sample data, the model is ACM-1511, shows in figure 1.3. This is a powerful IP indoor camera, which supports 8 frames per second at 1280 x 1024 resolution, or 30 frames per second at 640 x 480 resolution. We record sample video on a multi-media lab (see Figure1.2) in the Tamaki Campus of the University of Auckland.

For more information about this camera, please visit following link,

http://www.acti.com/product/detail/Bullet_Camera/ACM-1511.

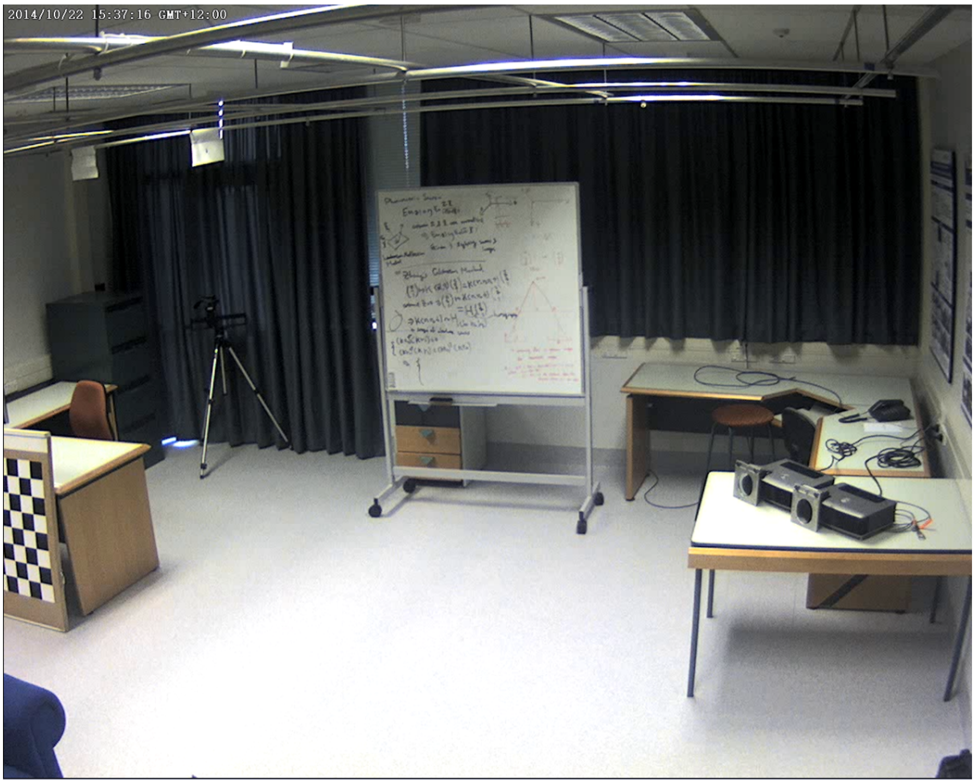


Figure 1.2: The multimedia lab in Tamaki Campus of the University of Auckland

1.4 Motivation and Goal

The application of video surveillance systems is today a common feature at public places. A frequent purpose of using a video surveillance system is to ensure security, such as detecting unusual behavior. A few years ago, commercial and public places used to rely fully on security personal, being increasingly replaced in recent days by surveillance cameras and automated, or semi-automated video analysis. Obviously it would not be efficient to have security personal now sitting in front of a screen for detecting unusual behavior in recorded data. Semi-automatic video security analysis applications are developed in recent years. Our project aims at understanding sudden changes in human poses in a public area, with a focus on sudden raising of hands in this paper (e.g. in the costumer area of a bank).

Traditional video surveillance systems record all the information for potential

Figure 1.3: ACM-1511

later use; this consumes a huge amount of storage space, and data are only used if suspicious events have been indicated by other means. Modern commercial video surveillance systems are capable of recording video only when there is any movement (e.g. human motion) detected, which is a first step towards reducing the bandwidth of data transmission and video storage.

Motion detection is commonly based on detecting changing pixels, not on interpreting the scene shown in a video, such as understanding the behavior of people in front of a camera. Motion detection results are still inaccurate to some degree, which may lead to false detections. We aim at providing a real-time surveillance system which classifies detected human silhouettes with respect to particular classes of human motions. In this paper we discuss the detection of hand raising.

On the other hand, crime of stealing such as bank robbery and store burglary happed very fast in real world. Sometimes the injured person can not trigger the security alarm by themselves due their behavior being restricted. It is very dangerous to trigger the alarm when the criminals threatening the peoples. To this end, an automatic alarm trigger can solve this problem. The security video surveillance system almost every where in the real world, if the automatic function of detection burglary acts can be applying to the video surveillance system.

This project will focus on two main area, foreground moving object detection and analyzing defined human action.

Recognizing of people holding up hands is the main goal on this project. At the end of the project, I hope to gain a good understanding of several computer vision techniques including foreground detection, human action analyzing, morphology processing, and how does those knowledge apply to real devices. Also, have a good experience on how to doing an academic research on computer science.

1.5 Structure of Report

In chapter 2, we will research and compare the several exist algorithm for silhouettes detection, then we choose one of the appropriate method. After implement the algorithm, we will show some foreground results.

In chapter 3, we will research on a set of method for refining detected silhouettes. The silhouettes results will contain noises, shadows and holes, which are affect human modeling. We will introduce the methods of shadow removal, noise removal, morphology operation and edge detection methods.

In chapter 4, we will discuss methods of pose understanding, we use detected silhouettes to matching a human model, then provide a bunch of classifier to recognize people raising up his hands.

At last, we will discuss the experiment results and make a conclusion.

Chapter 2

Silhouette Detection

This chapter discuss several method for moving object detection in order to get a human silhouette. The first subsection research on several famous methods. Then, we choose one suitable method for this project.

2.1 Overview about Methods

Currently, moving object detection has been researched over the past years but it is still a challenge problem. Several methods can be used for real time moving human detection of a surveillance system. However, there is no perfect algorithm for those purpose, these methods have their advantages and disadvantages. I briefly review methods as listed in surveys as follows,

The **Optical flow** is a very important method for moving object detection and analyzing. The definition of optical flow is defined by Gibson in 1950. The optical flow is the instantaneous speed of the moving object pixel in the space project on the observation plane. There are two assumptions in optical flow. First, the lighting condition should be not changing between frames, also called brightness constancy. The second is the motion speed is slow so that the motion speed can be differentiable. Optical flow can be used estimate the moving object foreground and tracing the moving object in real-time. Also, this method can be used when the camera is moving. However, dense optical flow computation are computationally expensive. This book [1] discusses the method step by step.

The **Background subtraction** approach is one of the basic techniques used for detection moving foreground objects (e.g. a human). It is widely used for video surveillance and not computational expensive. A non-moving camera is very suitable of using background subtraction approach, especially in our case. In another words, if the camera are moving, the whole processing procedure will fail due to the changing of almost every pixels. This paper [3] has proposed a comparison and

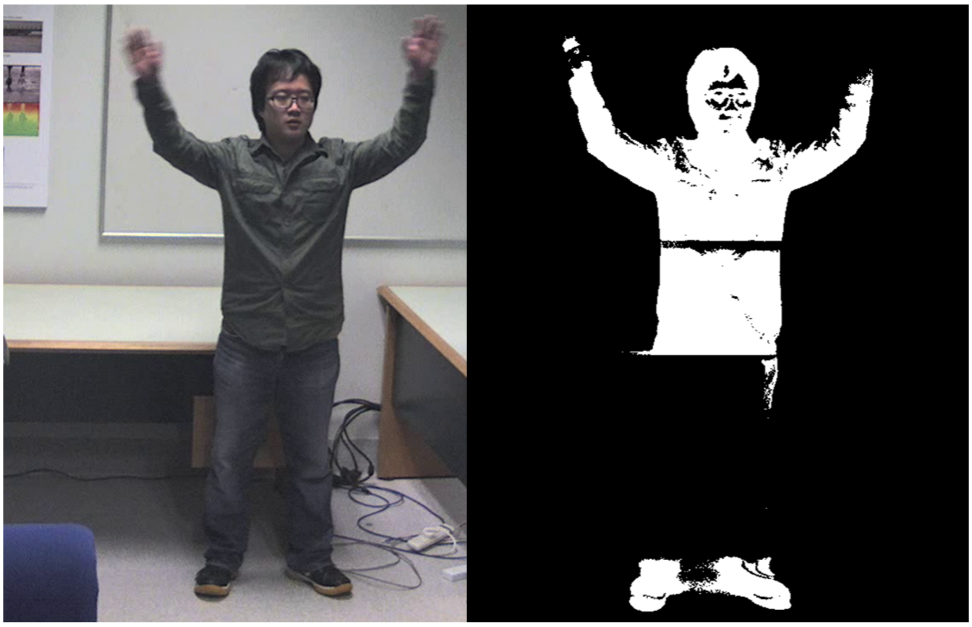


Figure 2.1: A sample using Gaussian mixture model.

description of real-time background subtraction algorithms for a video surveillance system.

All the background subtraction method have similar processing steps. The main processing steps is training a set of video sequence in order to get a background image, then set a threshold to subtract current frame between trained background frame. Also the background image will keeping update as time goes.

To subtraction background, we need to produce an background image first. There are several background modelling method has been presented.

(1) The *Frame differencing* algorithm is a simple and basic method for background subtraction by checking the difference in a set of consecutive frames. We considered the pixel as a foreground if the corresponding pixel have changed apparently by comparing threshold. Frame differencing is very easily to implement and use, but not very suitable for detecting slow moving object. Also, it is hardly to determine a good threshold value for any particular environment and the calculation step is limited by threshold which leads to inaccuracy results. This paper gives a solution using frame differencing [4].

(2) The *Gaussian mixture model* (or a mixtures of Gaussian) is defined by using a small number (say, between 3 to 5) Gaussian distributions for an additive description of background values.

There is a build-in method from OpenCV, BackgroundSubtractorMOG, which is implements Gaussian Mixture-based background and foreground segmentation algorithm described in [5]. This paper proposed a method to improve GMM mode by using a automatic method for adaptive lightning changing. An example by using OpenCV function, see figure 2.1.

(3) The *Median filtering* is a statistical background modeling method widely used in research area. Each background pixel is the median value of each corresponding pixel in all buffered frames. The background is defined as following equation,

$$B_t = Median(I_t, I_{t-1}, I_{t-2}, ..., I_{t-n}) \tag{2.1}$$

Where I_t is the frame at time t, B_t is the updated background frame. An n value is being used for decide how many n frames buffered to calculate background, thus the larger frame number we stored, the more frames we calculate. The *median filtering* has complexity $O(NlogN)$.

(4) The *approximate median filtering* introduced by McFarlane and Schofield [8], which is a statistical background modeling method complementary to simple *median filtering*. The following equation specifies how to update the background:

$$B(x, y, t) = \begin{cases} B(x, y, t-1) + 1, \\ \qquad\qquad \text{if } I(x, y, t) > B(x, y, t-1) \\ \\ B(x, y, t-1) - 1, \\ \qquad\qquad \text{if } I(x, y, t) < B(x, y, t-1) \end{cases} \tag{2.2}$$

where $I(x, y, t)$ is the value of an image pixel at position (x, y) at time t [1], and $B(x, y, t)$ is the value of a background pixel at position (x, y) at time t. Let $I(x, y, 0)$ (i.e. in the first frame) be the initial value of $B(x, y, 0)$. Then, in each subsequent time frame, we update background pixel values by comparing with previously assigned background pixel values.

2.2 The Chosen Method

We use the *approximate median filter* to update background image. And we use the source code provided by Zhengping Wang, a reference of his paper in [9].

The Figure **??** shows the steps of silhouette extraction algorithm.

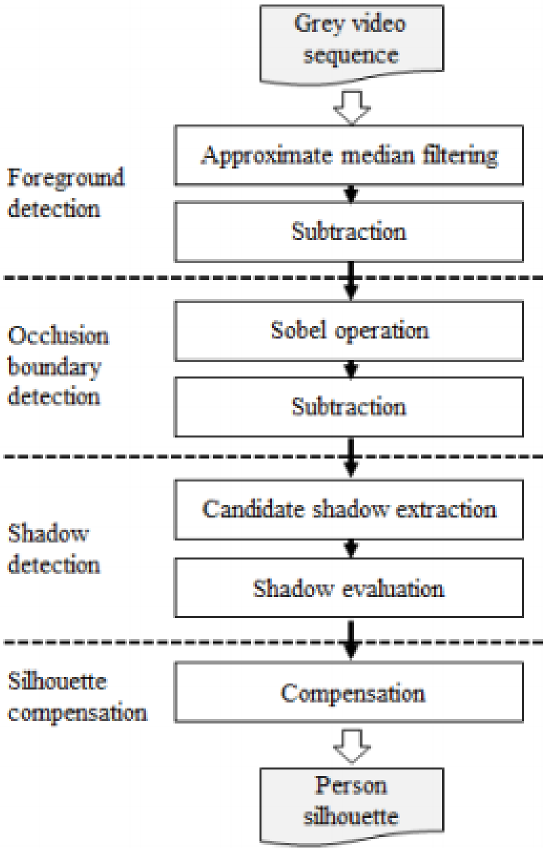


Figure 2.2: The steps of silhouette extraction (picture from the paper [9])

First, we training a background image by using *approximate median filter* as mentioned above. Then, estimate the background edges on the subtracted background image and raw occlusion boundaries of a person by using the Sobel operator (will discuss in section 3.2). Then, subtract the raw occlusion boundaries of a person and background boundaries in order to extract the true occlusion boundaries of a person. Finally, we can fill the true occlusion boundaries to get the foreground mask.

We estimate the foreground by subtracting the current frame from the back-

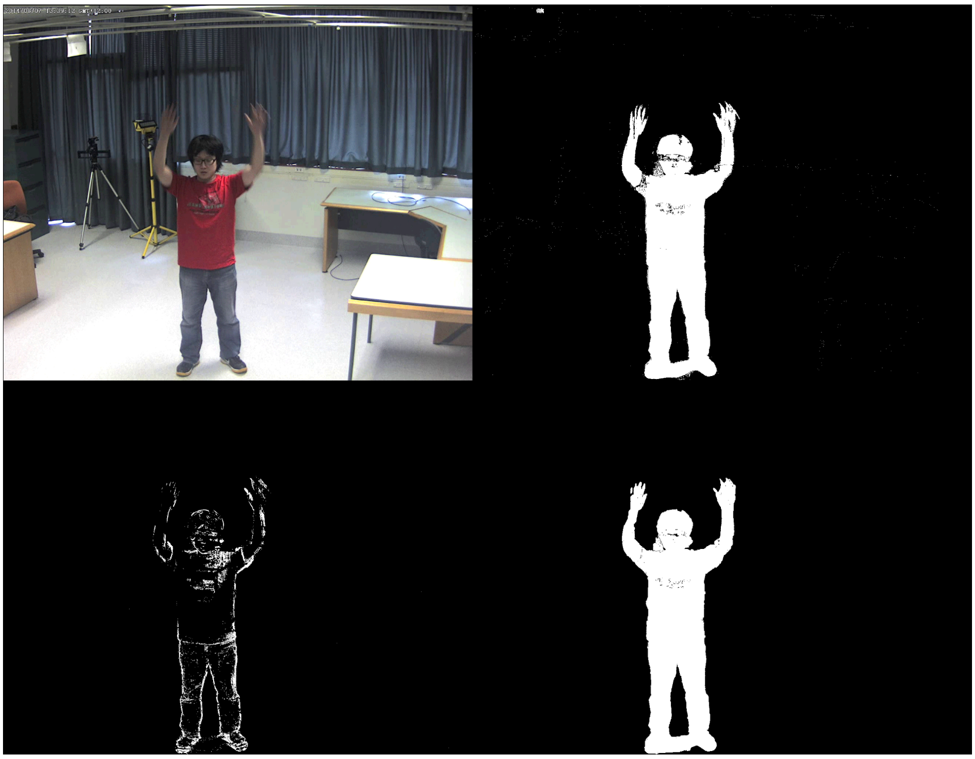


Figure 2.3: A screen shot of front silhouette. *top-left:* current frame, *top-right:* foreground detection result, *bottom-left:* occlusion boundary, *bottom-right:* result silhouette

ground image by using the following equation:

$$F(x,y,t) = \begin{cases} 1 \text{ if } |I(x,y,t) - B(x,y,t-1)| > \sigma_t \\ \\ 0 \text{ otherwise} \end{cases} \tag{2.3}$$

where $F(x,y,t)$ is the foreground pixel value at position (x,y) at time t, with initial values $F(x,y,0) = 0$ (i.e. all the pixels are considered to be background pixels at the beginning).

A pixel at (x,y) in Frame t is a foreground pixel if the absolute difference between the current value of $I(x,y,t)$ and the background value $B(x,y,t-1)$ is larger than a chosen threshold σ_t. Parameter σ_t is taken as the standard deviation of all

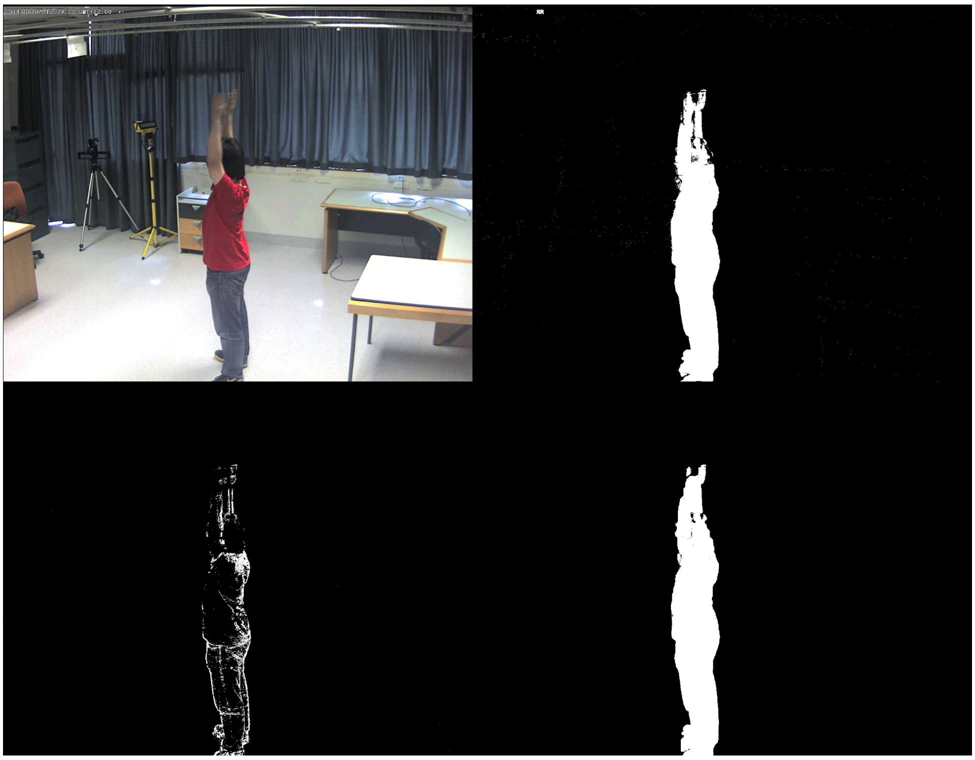


Figure 2.4: A screen shot of side silhouette of a person. *top-left:* current frame, *top-right:* foreground detection result, *bottom-left:* occlusion boundary, *bottom-right:* result silhouette.

input pixels in Frame t, defined by the following equation:

$$\sigma_t = \sqrt{\left(\sum_{x=1}^{N_{cols}} \sum_{y=1}^{N_{rows}} (I(x,y,t) - \mu_t)^2\right) / |\Omega|} \tag{2.4}$$

$$\text{with } \mu_t = \left(\sum_{x=1}^{N_{cols}} \sum_{y=1}^{N_{rows}} I(x,y,t)\right) / |\Omega| \tag{2.5}$$

where μ_t is the mean of all input pixels in Frame t, N_{cols} and N_{rows} are width and height of the frame, respectively, and $|\Omega|$ is the total number of pixels in the frame defined on carrier Ω [1].

Figure 2.3 shows a front human body silhouette result by using *approximate me-*

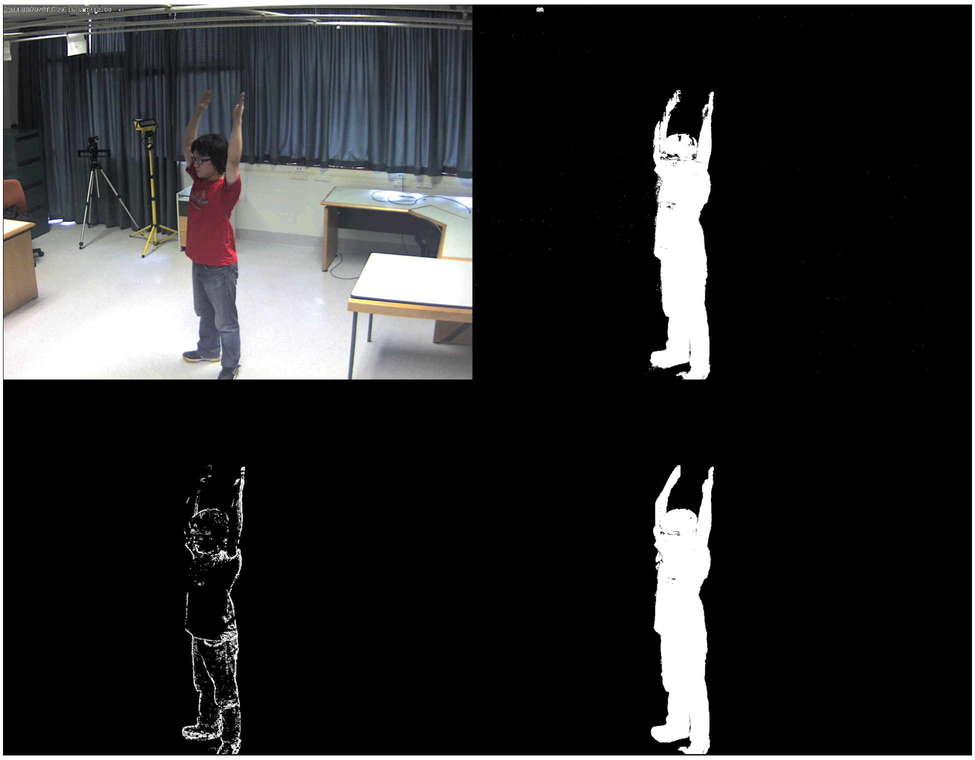


Figure 2.5: A screen shot of side silhouette of a person. *top-left:* current frame, *top-right:* foreground detection result, *bottom-left:* occlusion boundary, *bottom-right:* result silhouette.

dian filter of Zhengping's method, and figure 2.4 2.5 shows two different sides human body silhouette.

We use the Sobel operator as a simple and robust edge estimator on the subtracted background image for obtaining raw occlusion boundaries of a person.

We subtract the background boundaries from the raw occlusion boundaries (of a person) in order to extract the *true occlusion border* of a person.

Finally, we fill the true occlusion border to obtain the foreground mask, also called the *silhouette*.

In conclusion, these figures shown a good results of a human body silhouettes. However, the silhouettes still contain little holes and shadows. We will discuss the methods of shadow removal, holes filling, edges detection and noise removal in the next chapter.

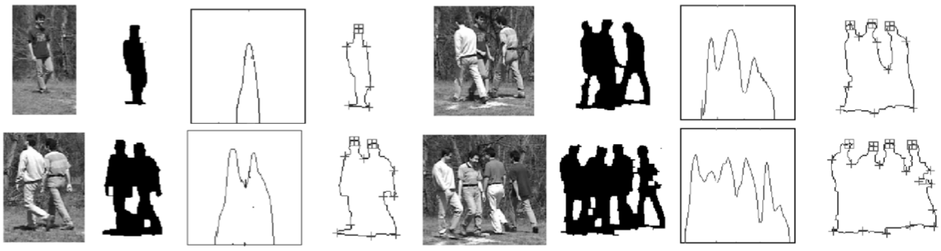


Figure 2.6: The group of people. The curve shows the histogram projection of silhouette. (This picture is from [19])

2.3 Silhouette Segmentation

The silhouette of each person may connected to each other when they are in the crowded scene. In order to get an isolated person silhouette, the silhouette segmentation came out to solve this problem.

W4 [19] presented a method by using the histogram projection of the foreground. This method also being used in Zhengping's master thesis [2].

The major steps is to find the vertical histogram of the extracted silhouette. Then using a linear smooth method to smooth the histogram curve, and reduce the minimal points as less as possible. Then the gap is the cut-off position, and the isolated silhouette can be considered as an single person.

The FIgure 2.6shows there is a gap between people, we can get the isolated silhouettes by cutting the gap position from the original silhouette image. However, the method just cut the complete silhouette into two even part (from the gap), but do not care which part is belong to the original person. . It is hard to distinguish which on is more front because surveillance camera is a monocular camera, we can only see people in two dimension. And it is also unnecessary, although we can have front people, the back people is still unseen if they overlap too much.

Chapter 3

Refining Detected Silhouettes

In this chapter, we will discuss the methods of refining detected human body silhouettes including shadow removal, holes filling and noise removal. After these process, we will get a meaningful results for future motion recognizing. The silhouettes results influenced by the lightning condition, room environment, background objects and camera position.

3.1 Shadow Removal

Shadow removal is an major problem in video surveillance. Moving shadow can also be detected by the background subtraction algorithm, but we do not need shadow pixel which will affect the further human detection analyzing.

Removing shadow normally can be done in a color image rather than a gray-scaled image, detect shadows in gray-scaled image is more complicated and challenge. An fast and robust shadow removal algorithm based on YUV color space has been reported in these paper [10]. An cast shadow method deal with Gaussian Mixture Models' drawback proposed in [11]. An approach based on RGB color space and K-means clustering algorithm [12].

In this project, we used an new shadow evaluation method proposed by Zhengping Wang in [9]. There are two main steps, candidate shadow detection and shadow evaluation, in order to detect shadow as accurate as possible.

Candidate shadow can be detected by forming an Gaussian distribution of the intensities of all detected foreground pixels. Real foreground pixels will close to the top of Gaussian curve, whereas shadow pixels are concentrate on each sides of Gaussian curve.

The candidate shadow can be detected as following equation 3.1,

$$S(x,y,t) = \begin{cases} 1 \;\; |I(x,y,t) - \mu| > \sigma \\ \\ 0 \;\; otherwise \end{cases} \tag{3.1}$$

Where, $S(x, y, t)$ is candidate shadow, $I(x, y, t)$ is current frame, $B(x, y, t)$ is background image, μ is the mean value of all foreground pixels' intensity, and an standard deviation σ is being used for reduce incorrect detection of shadow pixels.

Shadow evaluation step is used for correcting misclassified pixels. The paper defined a true shadow pixel should not be a pixel which is enclosed or semi-enclosed by pixels on an detected occlusion boundary of a person. Figure 3.3 shows an silhouettes after shadow removal. The shadow evaluation can be done by matching the shadow evaluator 3.1.

3.2 Holes Filling

In the extracted foreground silhouettes, there are a lot of holes shown in the body. Figure 3.2 shows a not saturated foreground silhouettes, which will leads the foreground object is not faithful to the original input frame. The reason is the the intensity value between background and foreground pixels are quite similar. Zhengping's shadow removal method can get the foreground value back by using his proposed method, but still contains little holes.

To fix this problem, we can use a morphology closing operator to fill small holes in an binary image.

However, morphology dilation depends on kernel size, large kernel size is not

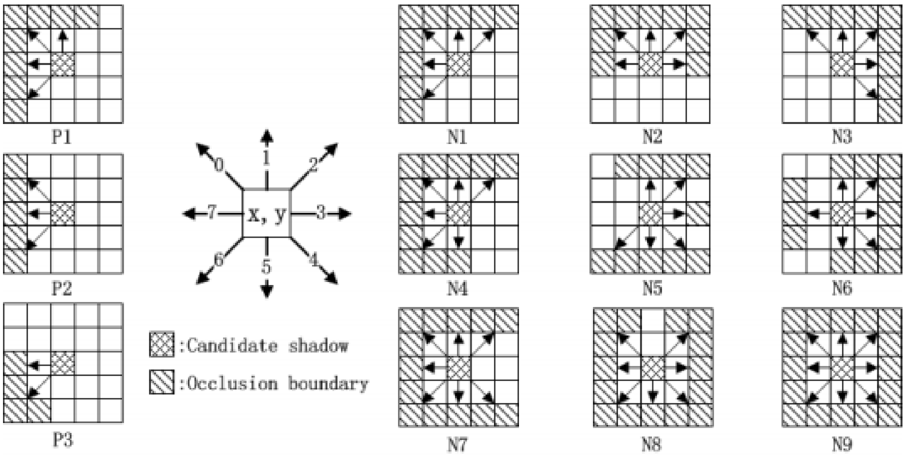


Figure 3.1: Shadow evaluator. (picture from the paper [9])

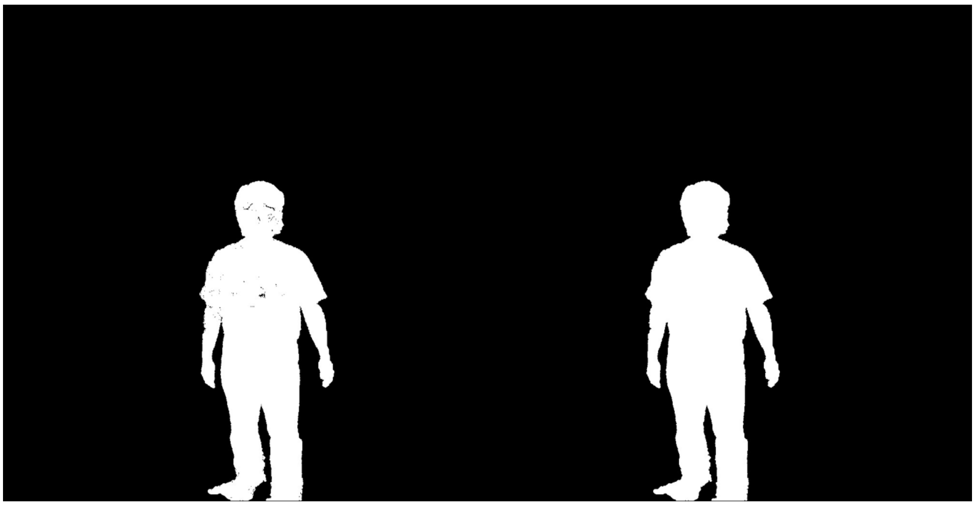


Figure 3.2: left: silhouettes with holes. right: filled silhouettes

computational cheap, small kernel size may not deal with large holes.

Here we use another simple method by using an OpenCV function, *cvDrawContours* with parameter CV_FILLED. The more specific steps as follow, we define a hole as a background region surrounded by a connected foreground region. First, perform edge detector on an image with holes. Then fill the pixels outside the contours, Finally, we can get an filled image by inverse the filled image before.

3.3 Noise Removal

morphological operation can be applied to both binary image and

Mathematical *morphological erosion* operation is an common tool for noise removal in image processing.

This is works well for the very small blob, but it can not handle large blob.

The large non human blob can be removed by following algorithm. Count the total number of every blob border pixels. Then set a threshold, if the total number less than threshold, we removed that blob from the frame. Here we use 800

An result image shows in figure 3.3.

3.4 Edge Detection

In this project, edge detection is being used for silhouette detection we mentioned in chapter 2. And also being used for extract contour of the silhouettes.

Edge Detection. Methods for edge detection follow the step-edge model. This means that we can detect an edge in an image by local maximal of absolute values of first-order derivatives or by zero-crossings in second-order derivatives.

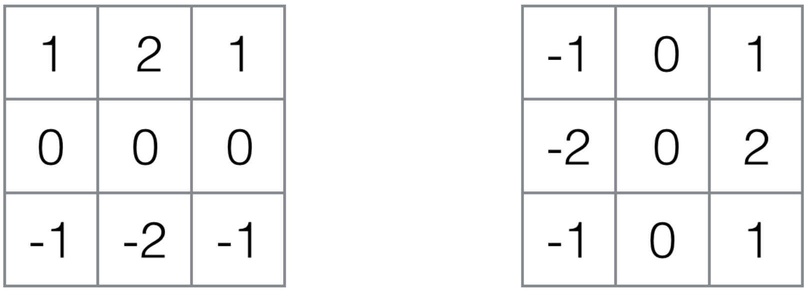
The *Sobel operator* is a simple example of a first-order derivative-type edge detector.

Sobel operator contains two 3 x 3 kernel, in figure 3.4, can be separately calculate the convolution between input image and one of the operator. Thus, $G_(x)$ is the horizontal Sobel edge convolution result, and $G_(y)$ is the vertical Sobel edge convolution result. Then the edge map can be calculate as following equation 3.2,

$$E(x,y,t) = \begin{cases} 1 \;\; if|G_(x)(x,y,t)| + |G_(y)(x,y,t)| > \alpha\sigma_(t)/2 \\ \\ 0 \;\; otherwise \end{cases} \tag{3.2}$$

Where $E(x,y,t)$ is the edge map, α is a low threshold in order to extract as many background edges as possible. So, true occlusion-boundaries can be detected by subtracting background boundaries and frame occlusion boundaries.

Figure 3.3: left: silhouettes with noise and shadow. right: silhouettes after shadow and noise removal



1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figure 3.4: Two Sobel operators left: horizontally operator right: vertically operator

Chapter 4

Pose Understanding

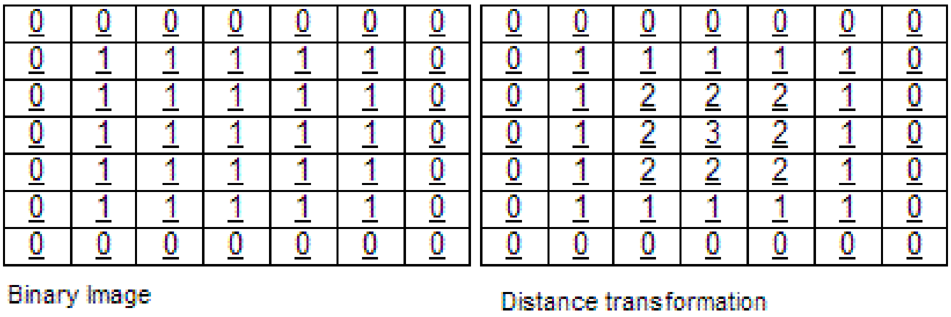
This chapter discuss the methods of human modeling based on human body silhouettes. After comparing the methods, this chapter discuss the proposed method for detecting raised-hands.

4.1 Matching a Human Model

Skeletonization is a common way for matching a detected region in an image with a model of a human. A moving target keep changing its boundary shape over time, a human skeleton can trace the feature of the shapes. There are many different techniques to perform a human skeleton, the methods list below,

Distance transformation is a set of geometrical mathematic operation to produce a foreground skeleton. The algorithm transfer the binary image to gray scaled image so that the gray level can represent the shortest distance between current pixel to the target pixel, as the FIgure 4.1 shows. To estimate the distance, the common way such as, euclidean distance, city block distance and chessboard distance are applying to the distance transformation. One common and simple method to perform distance transformation is calculate a set of erosion operation recursively, until all the foreground pixels are being removed. The specific information included in the paper [13].

Thinning Algorithm Thinning algorithm is general applying to pattern recognition, but also can be used to extracting the skeleton from human foreground picture. There are many different techniques to implement a thinning algorithm. Guo-Hall Thinning algorithm is explained in the paper [15] and Zhang-Suen Thinning algorithm is explained in [16]. Both algorithms thinning with two sub-iteration algorithms, or called parallel thinning algorithm, which is performs better than using sequential method. The thinning algorithm may generate different results due to different algorithms, however, the results are very similar, almost not influence the further application. The Figure 4.2 shows the difference of applying different thin-



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

Binary Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Distance transformation

Figure 4.1: *Left:* The binary image of original foreground. *Right:* The Grey-level image after applying distance transformation. (This picture from Wikipedia)

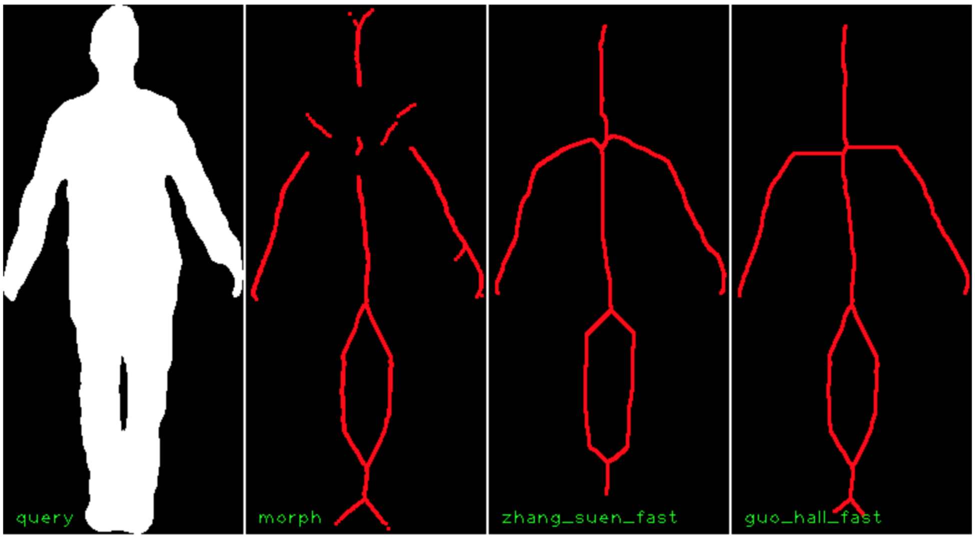


Figure 4.2: *From left to right:* The original foreground, the morphology method, the Zhang-Suen Algorithm, the Guo-Hall Algorithm. (The picture from the link [17])

ning algorithm. This link [17] contains the specific implementation and the results by comparing these algorithms.

Star skeletonization [14] proposed in 1998 by Fujiyosh and Lipton, which is an efficient and fast silhouette-based method for human modeling, subsequently for

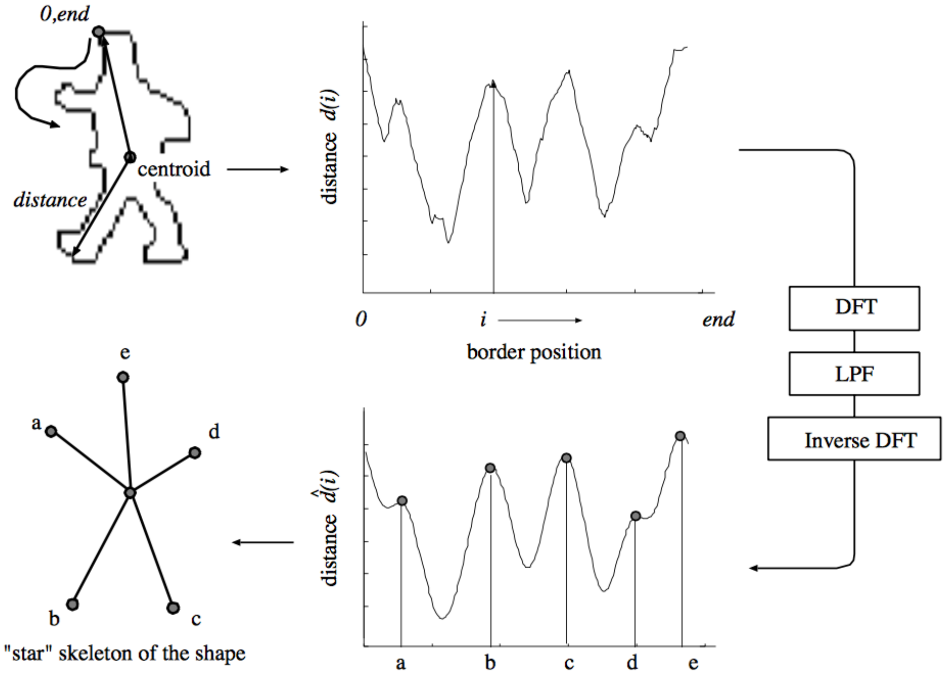


Figure 4.3: The steps of Star Skeletonization. (This picture is from [14])

the analysis of human poses. This method was proposed for classifying human poses into running or walking by using some kind of cycle detection in the Fourier domain. Advantage of this method is, it is not iterative and computationally cheap, which means very suitable for real-time processing. To estimate the human skeleton, the algorithm extracts the five *crucial points* includes head, left hand, right hand, left foot and right foot, this method described in [18].

Star Skeleton contains five steps(shown in figure 4.4), which is started at find the foreground contour. Then, the method defined a distance function in the traced foreground contour as the distance between the center of gravity and each contour pixels. After that, the method applying a low pass filter to smooth the distance signal in the frequency domain. In this end, the local maximal can be taken by finding zero-crossing. The star skeleton can be created by connecting the center of gravity and each local maximal points.

Thinning and distance transformation are computational expensive and highly

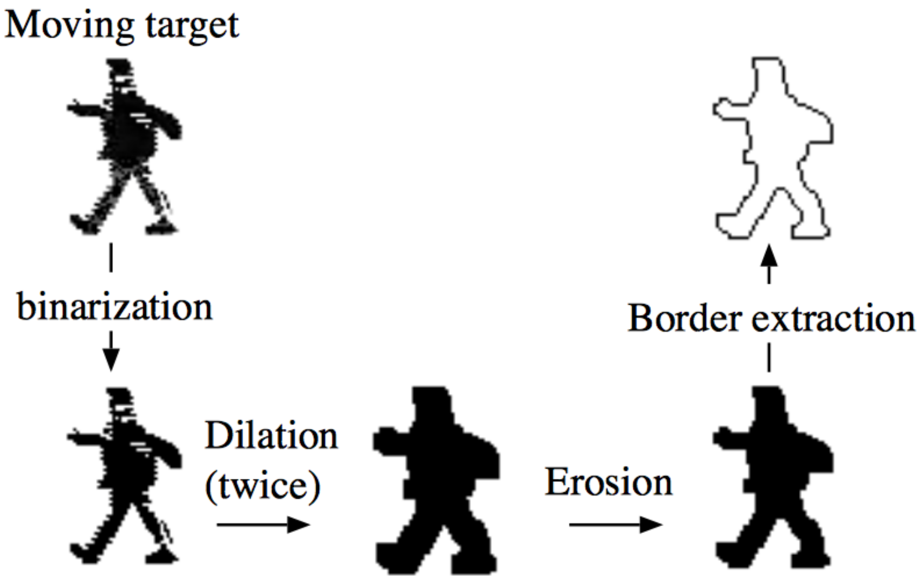


Figure 4.4: The pre-possess steps before star skeleton. Morphologically dilated twice followed by an erosion. (This picture is from [14])

depends on the correctness of the foreground, moreover, these methods are very sensitive to the noise and recursively operation, even improved by redesign the algorithm, such as parallel thinning algorithm.

Star Skeletonization only needs the information of the human border, and performs very fast than the other two skeletonization methods because it is not iterative.

4.2 Star Skeletonization

In our system, we use star skeletonization for modeling a human skeleton.

The proposed method is defined by a robust approach for detecting extrema points on the border of a detected silhouette. The algorithm for this subprocess consists of the following steps:

Step 1: Trace the border of the extracted silhouette. OpenCV provided the function *findContour* retrieves contours from the binary image.

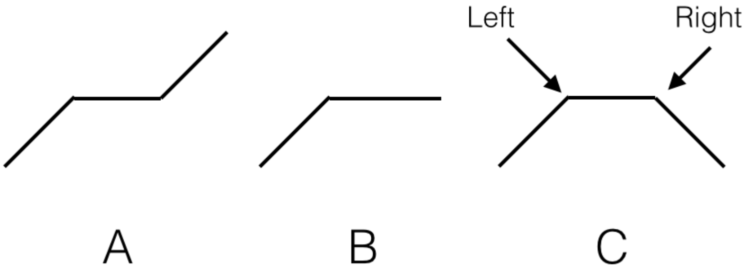


Figure 4.5: Three point sequences. A and B are not a maximal situation. C is a maximal situation.

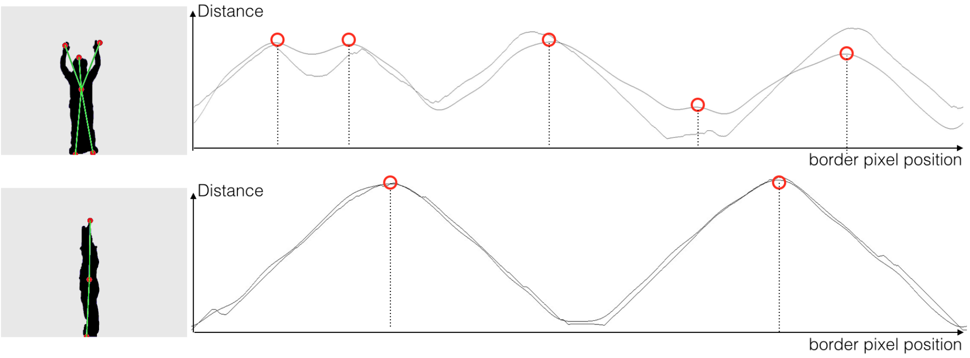


Figure 4.6: *Right:* Original distance signal D and smoothed distance signal $\hat{D}$ defined by a low pass in the Fourier domain. *Left:* Calculated skeletons.

Step 2: Find the center of gravity of the target border. Suppose that there are N border pixels, and the centroid of the border is denoted by (x_c, y_c), being defined as follows:

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad y_c = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{4.1}$$

for the N positions (x_i, y_i) of border pixels.

Step 3: In a second scan of the border, we define a distance function $D(i)$ which is the distance between the center of the gravity (x_c, y_c) and each border pixel point:

$$D(i) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{4.2}$$

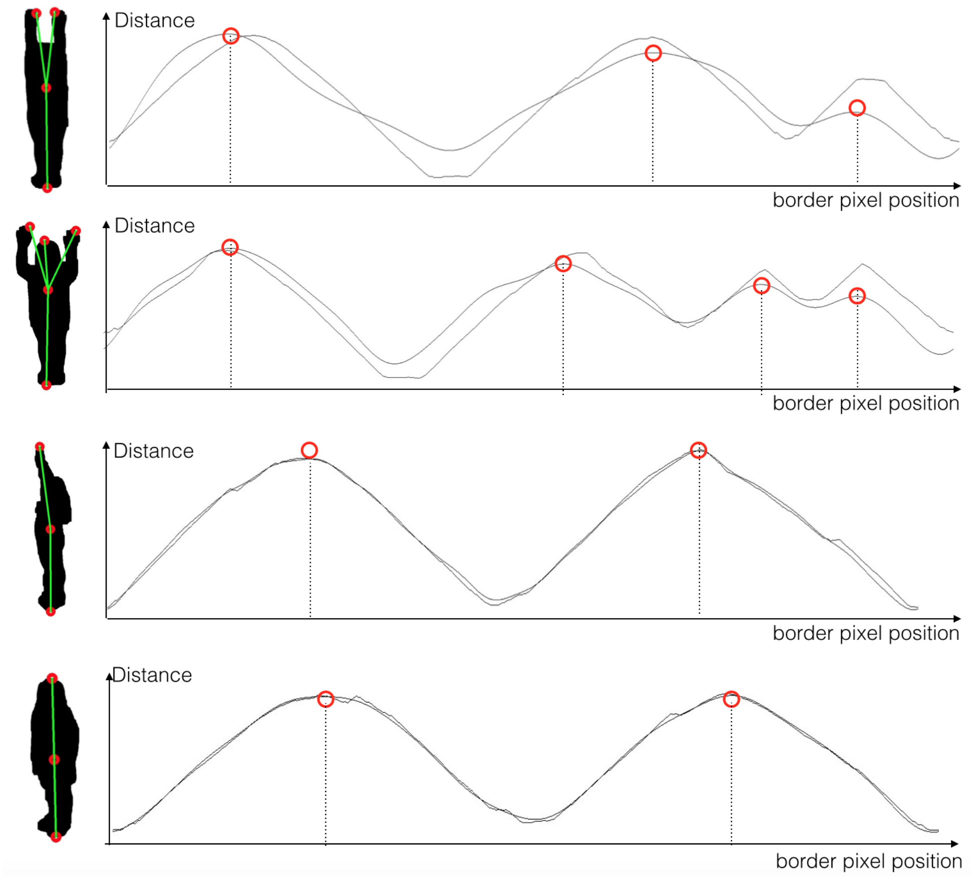


Figure 4.7: *Right:* Original distance signal D and smoothed distance signal $\hat{D}$ defined by a low pass in the Fourier domain. *Left:* Calculated skeletons.

Border tracing can be in clockwise or anti-clockwise order. In the experiments we used the Euclidean distance as shown in Equ. (4.2). However, we could also use the squared Euclidean distance or, for example, the L_1 (for saving computation time) without any significant impact on final results.

Step 4: The values $D(i)$ obtained in Step 3 are noisy (i.e. irregular). We smooth this distance signal $D(i)$ by applying a low-pass filter in the frequency domain. The low pass filter has a cutoff-threshold a for filtering out the high-frequency components. Let $\mathbf{D}$ be the Fourier transform of D. We set

$$\mathbf{D}(u) = 0 \text{ if } |u| \geq a \cdot N \tag{4.3}$$

where N is the total number of border pixels and u the frequency coordinate. The larger the threshold the more local maximal can be detected in the distance signal $\hat{D}$ (obtained after the inverse Fourier transform). Paper [14] used $a = 0.015$ as the threshold for their data and purposes, but we have higher image resolution and the goal of detecting raised hands, and we used $a = 0.0004$ in our experiments.

The Figure 4.11 shows the different signal and maximal.

Step 5: We take all the detected local maximal in the filtered signal $\hat{D}$ as extrema points. Figure 4.5 shows three different situations of $\hat{D}$ value sequences. Values may be constant within some intervals. A local maximum is calculated for situation C by taking the mean

$$i_{\max} = \frac{i_L + i_R}{2} \tag{4.4}$$

of the shown left and right endpoint of the interval (having indices i_L and i_R in $\hat{D}$) of constant values.

A human skeleton is now constructed by connecting the center of gravity with the detected local maximal for the given silhouette. Figure 4.6 illustrates in the top row the resulting skeleton for a front-view of a human silhouette. Figure 4.7 illustrates in the four different poses of a girl. Parameter a was chosen in a way such that the smoothed signal $\hat{D}$ of such a front-view human silhouette typically contains five maximal points. Each of those maximal points matches then typically a distinctive feature of the human body (i.e. feet, head, and possibly hands or shoulders). The bottom row in Fig. 4.6 illustrates a side view for raised hands.

4.3 Understanding Poses

For pose understanding, we use the human modeling results, by comparing the extrema points. An classifier needs to be construct to recognize human raising up hands towards different direction.

There are a brief and selective review of examples of related work.

W4 [19] is a proposed solution for multi-person tracking and activity recognition when using an outdoor video surveillance system. W4 combines various methods of silhouette shape analysis and tracking to determine whether people are carrying objects, or whether people move coordinated in a group. W4 starting at the foreground detection by using background scene modeling. Then W4 proposed a foreground region classifier to divide the foreground scene into three group, single person, people in group and other non-people foreground. For the single person, W4 filter out the non-human foreground if the foreground contains carried-object and backpack by using the symmetry analysis and periodical analysis. W4 deal with the people in group situation by using head detection and person segmentation. Finally the filtered isolated silhouette can be applying the tracking algorithm include, motion model, appearance model and trajectory recovery.

The paper [20] presents another human action recognition method by using three human silhouettes. To create a silhouette descriptor, the method maps the human silhouette into three polar coordinate systems that represent either the whole human body, the upper, or the lower body part, respectively. An action classifier can then be trained based on the derived descriptor.

Paper [21] proposed a method for recognizing the raising of a hand in a meeting or class room. The method locates arms in the geometric structure of detected edges, and then compares the angle of a detected hand with the x-axis. Paper [7] also discusses the detection of a raised hand. A *candidate region* (CR) region is identified from silhouettes by locating positions of body parts.

Star skeleton we mentioned above, was being made to several application. Star skeletonization is used in [22], based on a hidden Markov model, for action recognition.The method can identify human sit up, sit down, exercise and walking. The paper [23] proposed a method to improve the Star skeleton method. The paper used reference descriptor matrix and a set of tracking method to reconstruct the star skeleton model.

Based on the former research and observation on the experiment. It is hardly to identify the position of the meaning maximal point, such as which one is the head maxima. When people walking around and acting in front of the camera, the people can towards to any direction and the silhouette can be formed any shape. So there is a new classifier to deal with this situation.

4.4 Detection of Raised Hands

A person which is raising the hands can be recorded by a camera in various *poses* (i.e. positions and directions). Figure 4.8 illustrates three different *main poses* of a person raising hands as usually appearing in a surveillance camera. The figure illustrates that the front-view skeleton normally contains five maximal points, and raising hands are defining two of those.

The top row also shows a variation in front-view skeletons. For the first frame, only head and feet are detected. The shoulders do not lead to maxima in the filtered

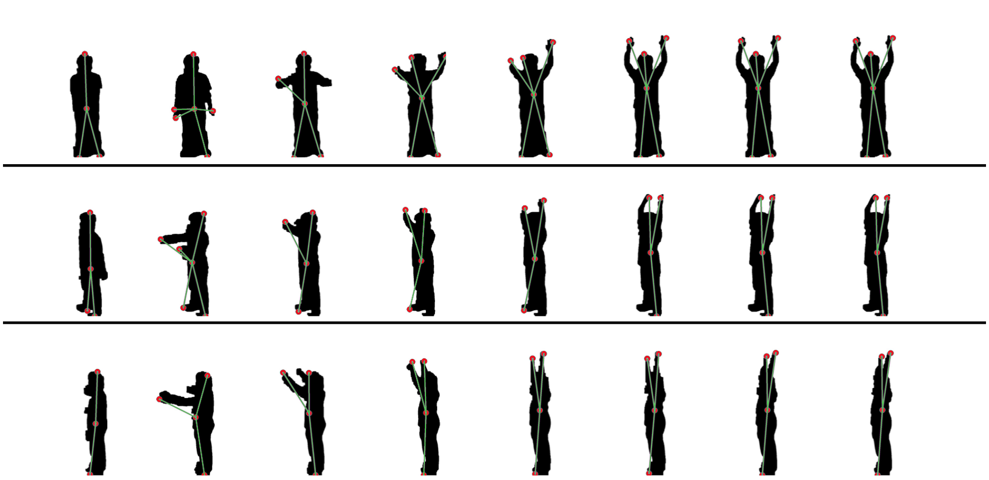


Figure 4.8: Different states of a person which raises the hands. *Top Row:* Front view. *Second Row:* Half-side view. *Bottom Row:* Side view.

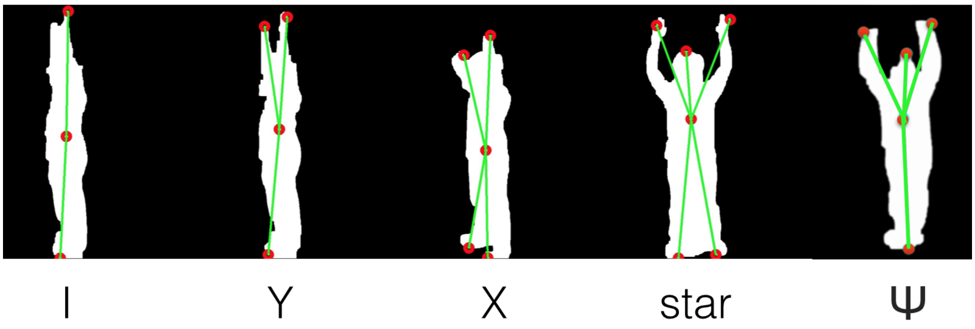


Figure 4.9: From left to right: *I-*, *Y-*, *X-*, *star*, *ψ-shape*,

distance signal. The second frame misses the detection of the left foot, and there are two maxima detected for the left hand. The third frame is missing the right hand. All the other frames lead to five maximal points.

The second row shows states of a half-side pose. The first frame does not detect hands since they are barely seen for this pose. The second frame contains all the five maximal points. The next six frames all miss to detect one of the feet.

When raising the hands, the skeleton forms a *Y-shape* for front and half-side pose.

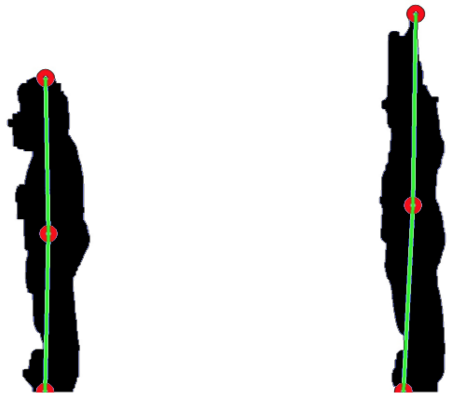


Figure 4.10: The two extreme cases denoted as *I-shape*. A temporal change in height indicates the raised hands. *Left:* A side view of a standing person. *Right:* A side view of a standing person having the hands up.

The third row shows a side-view skeleton. The skeleton is at first (i.e. first frame) just connecting two points (head and feet), and turns then into a Y-shape when the person is raising the hands.

Following above observations, the formed skeleton does not always contain exactly five maximal points due to differences in poses. That means, head, hands, or feet cannot be detected and tracked correctly as maximal points over a sequence of frames.

If a person is raising the hands, in all cases the common feature is that the positions of the two hands are on top of the head position at some stage. In consequence, we focus on the analysis of three maximal points defining the positions of three upward maximal.

We propose to detect raised hands by identifying one of five shapes, called *I*-, *Y*-, *X*-, ψ-, or *star-shape*, shows in Figure 4.9.

A single upper maxima in the *I*-shape, or two upper maxima in the *Y*-, *X*-, ψ-, or *star*-shape, are considered to be hands, and the bottom two maxima in the *X*- and *star*-shape are considered to be the feet.

For the *Y*-, *X*-, ψ-, or *star*-shaped skeletons, we specify two thresholds for comparing hand(s) and feet maximal with an adaptive estimate H for the height of the shown person (based on calculating heights of silhouettes over some frames by a sliding mean). Assume that the y-axis is pointing upward in the image. Then we use

$$y_{\text{hands}} > 0.8 \cdot H \ \& \ y_{\text{feet}} < 0.2 \cdot H \tag{4.5}$$

By using this method, we are able to identify Y-, X-, or *star*-shapes fairly accurate in a given frame.

The *I*-shape (see Fig. 4.10) requires a particular consideration. Here we compare the current height with the previously estimated height H.

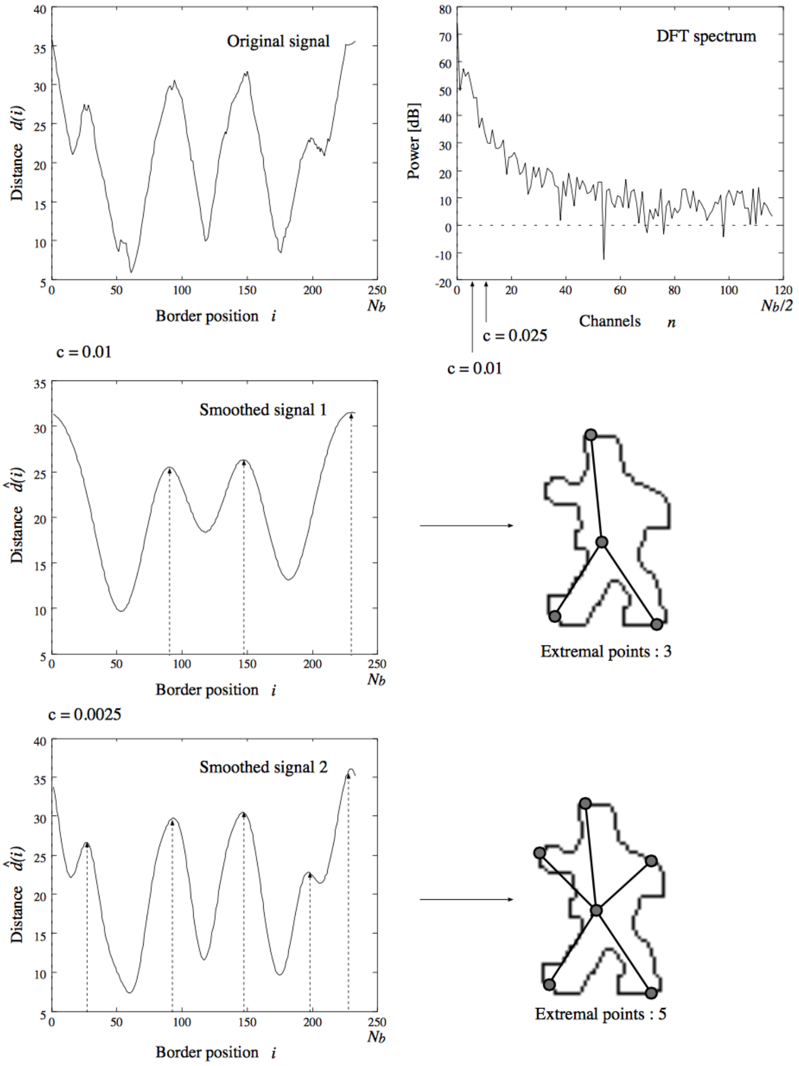


Figure 4.11: *Upper Left:* The original signal. *Upper Right:* The DFT spectrum. *Middle Left:* The smoothed signal with threshold 0.01. *Middle Right:* The smoothed signal with threshold 0.0025. *Lower Left:* The star skeleton with three local maximal. *Lower Right:* The star skeleton with all five local maximal. (This picture is from [14])

Chapter 5

Conclusions

This chapter describes the the experiment results of the proposed method. Then summaries the project.

5.1 Experiment Results

In this section we report results obtained for five different test persons. The first test person (called Person A) was repeatedly shown in previous figures. For test persons B, C, D, and E, see Fig. 5.1 for some illustrations.

We use a monocular (non-calibrated) IP camera (ACM-1511) for recording video samples. In order to simulate a standard indoor surveillance situation, the camera is mounted close to the ceiling. Recording is at 8 fps, with 1280 x 1024 image resolution.

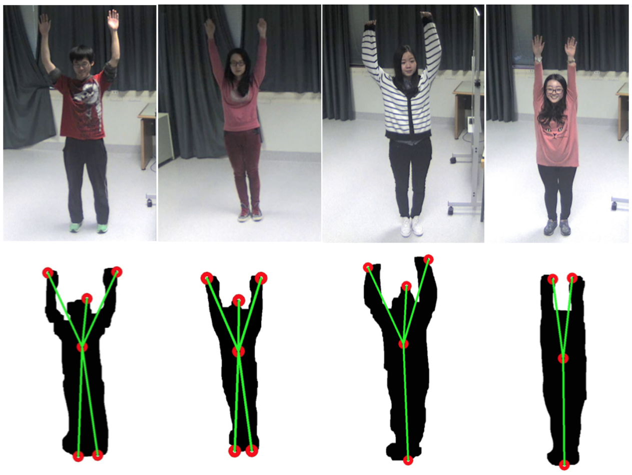


Figure 5.1: Samples of extracted silhouettes for test persons B to E, left to right.

We tested our algorithm on several video sequence for five different people. For each person, recorded sequences contained all the three poses illustrated in Fig. 4.8. Table 5.1 summaries the results of pose classification. FP (i.e. false-positive) specifies the number of detections of raised hands when there are actually no raised hands, whereas FN specifies the number of cases where we miss to detect the raised hands in a frame. We manually classify poses into "hands raised" if both hands are at the height level of the head, or above the head. The "Ratio" is finally the total number of frames minus (FP+FN), divided by the total number of frames, i.e. the percentage of correct decisions.

Table 5.1: Classification results for raised hands for five test persons

Sequence	Total frames	FP	FN	Ratio
#A	172	0	4	97.67%
#B	305	2	10	96.07%
#C	287	22	4	90.94%
#D	293	4	8	95.90%
#E	243	5	4	96.30%
Total	1300	33	30	95.15%

For example, the relatively large value FP = 22 for Person C is due to missing head silhouettes; Person C defined a "difficult case" for silhouette detection due to signal similarities between person and background.

5.2 Conclusion

This project presents a system for real-time detection of defined human poses (i.e. raising of hands) in surveillance video. A single (non-calibrated) video camera (ACM-1511) is used to record data in an indoor environment. There are two main steps in our proposed system, the extraction of human silhouettes in video data and pose classification. Silhouette extraction is refined by paying attention to the removal of shadow artifacts close to occlusion borders. For pose classification, we combined, adjusted, and implemented two existing methods (star skeleton calculation and its evaluation). We demonstrate that the proposed two-step technique is solving the given task for a large percentage of input data when recording an

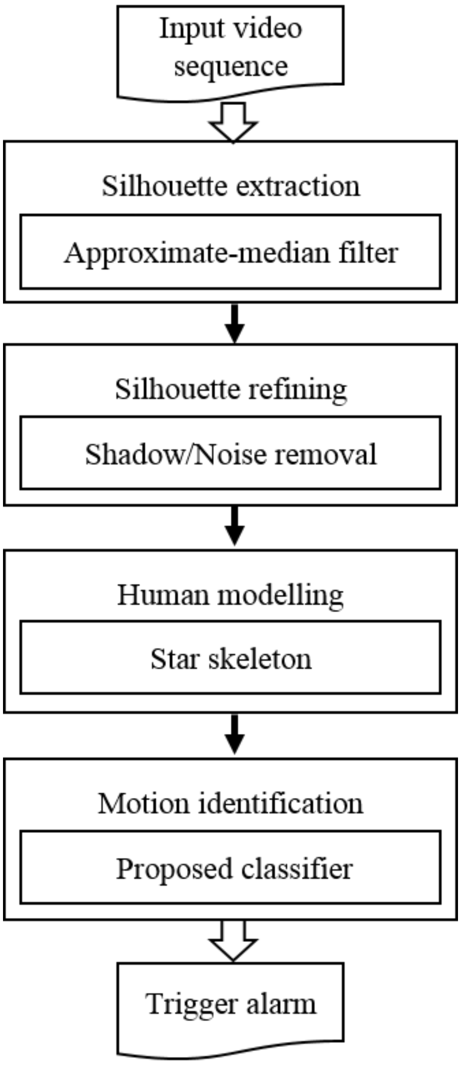


Figure 5.2: The steps of proposed system.

individual person only.

The proposed system contains four steps, starting at silhouette extraction by using the approximate-median filtering, then followed by a set of methods to refining the silhouettes, including shadow and noise removal. After that, the system used the star skeleton to matching a human model. At last step, the system proposed classifier to identify human raised hands. The Figure 5.2 shows the full steps of the system.

The system can achieve an high accuracy rate for raised hands detection of a single person. The software tested on five peoples with different features such as girl with long hair or wearing different cloth.

However, there are limitations in this system. The silhouette extracting method is very sensitive to the suddenly lightning condition change, that leads to an inaccuracy silhouette. Normally, an indoor environment should not have an suddenly light change. The lightning changing problem in our experiment is due to the automatic lightning adjustment function of the ACM-1511 IPVS camera.

Also, our proposed method can only achieve an high accuracy rate of single person raised hand detection. Detecting the raised hand for multiple person is more complicated and challenging. Simply applying the method to multiple person situation will leads error when the silhouette of persons connected to each other.

In this project, we considered the performance based on the algorithm design, not on the programming circumstance. In the future work, the performance estimation and parallel computing should involved in this system.

In conclusion, the proposed system have potential to be used in the real world, also there are many drawbacks of the system can be improved in the future.

Bibliography

[1] R. Klette: *Concise Computer Vision*. Springer, London, 2014.

[2] Z. Wang: Extracting Human Silhouettes from Video Sequences. MSc Thesis, Department of Computer Sciencem the University of Auckland, New Zealand, 2014.

[3] M. Hedayati, Wan Mimi Diyana Wan Zaki, Aini Hussain: A qualitative and quantitative comparison of real-time background subtraction algorithms for video surveillance applications. In *Journal of Computational Information Systems*, **8(2)**:493-505, 2012.

[4] N. Prabhakar, V. Vaithiyanathan, A. P. Sharma, A. Singh, and P. Singhal: Object tracking using frame differencing and template matching. In , 2012.

[5] P. KaewTraKulPong and R. Bowden: An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pp. 135–144, Springer, 2002.

[6] P. Correa, J. Czyz, T. Umeda, F. Marques, X. Marichal, and B. Macq. Silhouette-based probabilistic 2d human motion estimation for real-time applications. In Proc. *IEEE Int. Conf. Image Processing*, pages 836–839, 2005.

[7] X. Duan and H. Liu. Detection of hand-raising gestures based on body silhouette analysis. In Proc. *IEEE Int. Conf. Robotics Biometrics*, pages 1756–1761, 2009.

[8] N. J. B. McFarlane, C. P. Schofield: Segmentation and tracking of piglets in images. In *Machine vision and application*, 187-193, 1995.

[9] Z. P. Wang, B. S. Shin, and R. Klette. Accurate silhouette extraction of a person in video data by shadow evaluation. In *Int. J. Computer Theory Engineering*, **6**:476–483, 2014.

[10] O. Schreer, I. Feldmann, U. Golz, and P. Kauff: Fast and robust shadow detection in video conference applications. In *Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom*, pp. 371–375, 2002.

[11] B. E. Lee, T. B. Nguyen, and S. T. Chung: An efficient cast shadow removal for motion segmentation. In *Signal processing, computational geometry and artificial vision*, pp. 83–87 , WSEAS, 2009.

[12] A. Chowdhury, U. Chong: Real time shadow removal with k-means clustering and RGB color model. In *International Journal of Multimedia & Ubiquitous Engineering*, pp. 159, SERSS, 2012.

[13] G. J. Grevera: Distance transform algorithms and their implementation and evaluation In *Biomedical and Clinical Applications*, pp. 36–60, 2007.

[14] H. Fujiyoshi, A. J. Lipton: Real-time human motion analysis by image skeletoniztion. In *In Applications of Computer Vision Proceedings*, pp. 15–21, IEEE, 1998.

[15] Z. Guo and R. W. Hall: In *Parallel Thinning with Two-subiteration Algorithms*, pp. 359–373, ACM, 1989.

[16] T. Y. Zhang and C. Y. Suen: In *A Fast Parallel Algorithm for Thinning Digital Patterns*, pp.236–239, ACM, 1984.

[17] Arnaud Ramey: In Super-fast thiinning implementation(Zhang-Suen, Guo-Hall), https://sites.google.com/site/rameyarnaud/research/c/voronoi, 2013.

[18] P. Correa, J. Czyz, T. Umeda, F. Marques, X. Marichal, B. Macq: Silhouette-based probabilistic 2D human motion estimation for real-time applications. In *IEEE International Conference on Image Processing*, pp. 836–839, 2005.

[19] I. Haritaoglu, D. Harwood, and L. S. David. W4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis Machine Intelligence*, **22**:809–830, 2000.

[20] C. Hsieh, P. S. Huang, and M. Tang. Human action recognition using silhouette histogram. In Proc. *Australasian Computer Science Conference*, pages 11–16, 2011.

[21] N. B. Bo, P. V. Hese, D. V. Cauwelaert, P. Veelaert, and W. Philips. Detection of a hand-raising gesture by locating the arm. In Proc. *IEEE Int. Conf. Robotics Biometrics*, pages 976–980, 2011.

[22] D. Singh, A. K. Yadav, and V. Kumar. Human activity tracking using star skeleton and activity recognition using HMMs and neural network. *IJSRP*, **4**, May 2014.

[23] B. Muthukumar, S. Ravi: Tracking the human motion in real time using Star Skeleton Model. In Vol. 1, Issue 3, IJEAT, 2012.

[24] I. Haritaoglu, D. Harwood, and L. S. Davis: Real-time surveillance of people and their activities. In *Pattern Analysis and Machine Intelligence*, pp. 809-830, IEEE Computer Society, 2000.

[25] X. Chen, Z. He, D. Anderson, and J. Keller, and M. Skubic: Adaptive silhouette extraction and human tracking in dynamic environments. In *Fuzzy System*, pp.236–243, IEEE Computer Society, 2006.

[26] A. Manzanera: Human motion analysis: tols, models, algorithms and applications. Tutorial presented in ENSTA-ParisTech, Aug–5–2009

[27] M. Dahmane, J. Menuier: Real-time moving object detection and shadow removing in video surveillance. In , , 2005.

[28] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure: Robust silhouette extraction technique using background subtraction. In , 2007.