Gesture Detection in Video Surveillance

Mid-year Report for BTech451

by

Yuan Wang

Tamaki Innovation Campus
Department of Computer Science
The University of Auckland
New Zealand
June 2013

Abstract

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images. And image processing is a way to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. Detecting people in images are essential to intelligent vision-based human computer interaction, it has been widely used in many areas including video surveillance, robotics, driver assistance system, etc. The use of computer vision systems now can not only provide human detection in surveillance video, but also offers a promising solution to analyze people behavior and detecting unusual movements and gestures.

In the first semester of my BTech451 course, I collected video data using the IP camera provided by Compucon company, propose Haar-like features method to detect human faces in captured video data, and gain some programming experience, then have a basic understanding with human whole body detection using Random Forests algorithm.

Keywords: image process, human detection, Haar-like features, Random Forests.

Contents

Chapter 1

Introduction

This is an one year project for students who majoring in Bachelor of Technology(IT) degree. The project carries weight of two University of Auckland courses and divided into two parts, BTech451A in semester 1 and BTech451B in semester 2.

This report is not the final version of this project, it is for BTech451A, thus only keeps tracks on first half of the project process. Full version of this report will be provided at the end of year 2013.

1.1 Project Overview

This project focus on human detection in first part. There are several scenarios that need to be considered about, face detection can be done while having people with their front face right towards the camera using Haar-like features method, human detection would be done while there is only one people showing in the video with his whole body shape displayed using Random Forests algorithm. Multiple people detection is also an challenging problem.

The second part of the project is selected human motion or gestures classification in surveillance video data. e.g. hands-up in bank robbery, people laying down on the ground, or a special action which can be previously defined that system treated as a signal that dangerous enough to raise an alarm.

1.2 Project Goal

My project this year will focus on human detection, after having experiences using Haar-like features to detect human faces, and continue working on understanding detection of whole human bodies using Random Forests, hopefully to achieve some classification on motion and gesture detection in surveillance video data.

1.3 Company Information

This project is sponsored by the company Compucon New Zealand.

Compucon NZ is part of an International Computer manufacturing group of companies founded in 1989 in Sydney.

The NZ operation is registered as Modern Technology NZ Ltd and has established a reputation for technical excellence based on sound engineering and other knowledge based practices. All manufacturing processes are certified by Telarc ISO 9002 quality standards at our Albany assembly plant in Auckland NZ.

The Compucon team contributes to the success of our customers through our knowledge, excellence, commitment and supply of computing platforms and solutions meeting or exceeding customer expectations.

Chapter 2

Data Collection

For this project Compucon has supplied a video surveillance camera for data capturing, after installed the camera in university laboratory several video with different resolutions and scenarios are recorded.

2.1 Camera Information and Setting

The model of this camera is CA-1511, it is a mega-pixel POE(Power over Ethernet) IP camera suitable for Day and Night Indoor surveillance use, it supports either 30 frame per second in full D1 resolution, or 7 frames per second in SXGA mega-pixel (1280x1024) resolution.is equipped with Infrared LED and is capable of night vision at 0Lux at F1.6. An brief information of the camera can be found at: *http://www.compucon.co.nz/content/view/442/226/*

The software processing images from the camera is ACTi NVR3.0, since it is an IP camera, the IP addresses of both camera can computer are also need to be set correctly.

We decided to work on indoor human motion detection, multi-media lab (room 723-123) in tamaki campus is decided to be the place for recording video, since we are not supposed to damage walls or ceiling of the lab, the camera is mounted on the shelf of the projector, another position is on the table that make the camera face towards the door of the lab. It gives some limitation when recording video data since the view is not wide enough, thus when we record multi-people, it is hard to capture the whole human body shapes.

2.2 Sample images of recorded video

Figure 2.1-2.4 are some sample images in recorded video data showing different scenarios:

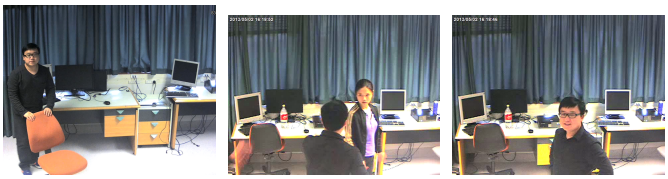


Figure 2.1: people with different directions and positions that only shows one face in view

Figure 2.2: one person hands up with different directions and poses

Figure 2.3: multi-people hands up while pretending queuing in the bank environment

Figure 2.4: people with different motions, directions under infra-red mode

Chapter 3

Face Detection

For this chapter, I have done some research on understanding Haar-like features and AdaBoost, i also trained my own classifier for face detection in programming, some results will be shown in the and of this chapter.

3.1 Build weak Classifiers using Haar-like features

Haar-like features are digital image features used in object recognition. Figure 3.1 shows some examples of used features prototypes:

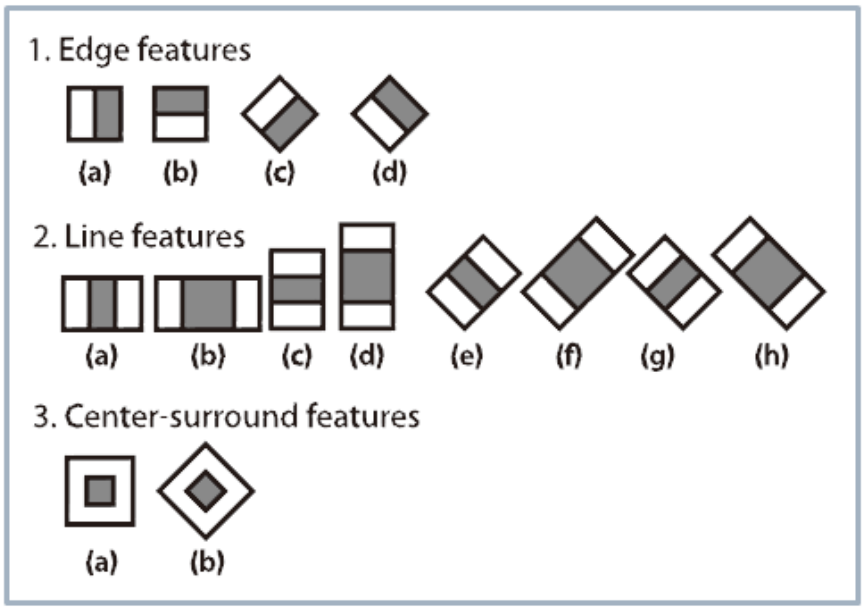


Figure 3.1:

A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.[1] Then the difference of this sum is used to classify subsections of the image. In this project, we have an image database with human faces, It is a common observation that among all faces the region of the eyes is different than the region of the cheeks. Therefore we can choose a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case), thus after selecting 2-5 rectangular features which best separate the positive and negative examples, we can define ONE weak classifier.

The main challenge here is to find a set of Haar-like features that can be combined to form the best effective weak classifier, to achieve this goal, a weak learning algorithm is designed to select one feature that best separate face or not face images, for each feature, the weak learner determines the optimal threshold classification function, therefore the minimum number of examples are misclassified. A weak classifier $h_j(x)$ that consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 \; p_j f_j < p_j \theta_j \\ 0 \;\; \text{otherwise} \end{cases} \tag{3.1}$$

here x is a 24×24 pixel sub-window of an image.[2]

3.2 Build strong classifier using AdaBoost

AdaBoost is short for Adaptive Boosting, it is a machine learning algorithm.[3] It decides about the order of applied weak classifier with weights, here the weights indicate the 'importance' of the weak classifier, and also it is used when training data. On each round, the weight of each incorrectly classified example are increased, and the weights of correctly classified examples are decreased, thus the trained classifier is able to deal with examples which eluded correct classification before.

Figure 3.2 is the structure of an AdaBoost cascade, by combining several weak classifier ($F_1, F_2, ...$) into one strong classifier we construct a cascade of classifiers which achieves increased detection performance while reducing computation time.

A cascade of classifiers is like a decision tree where at each stage the weak classifier we trained in last step is formed to detect object of interest like faces, while rejecting a certain number of the non-object patterns[4].

Each stage was trained using AdaBoost algorithm. With increasing stage number, the number of weak classifiers, which are needed to achieve the desired false alarm rate at the given hit rate, increases.

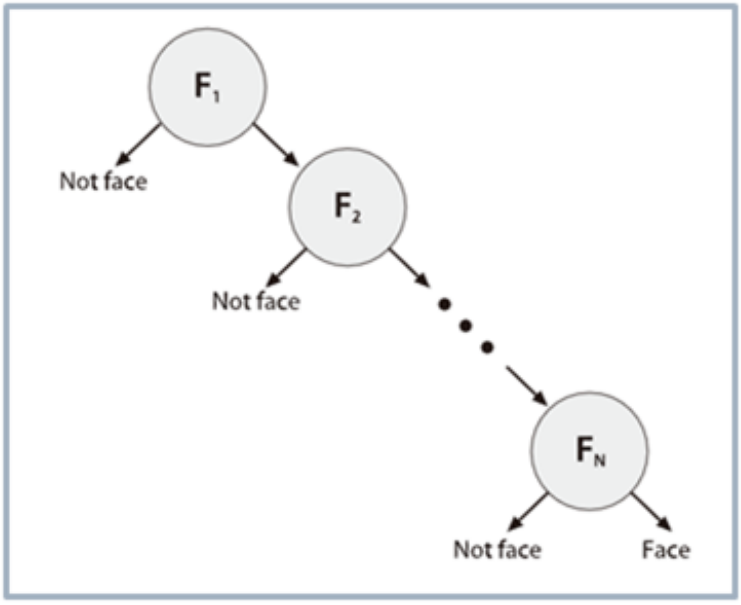


Figure 3.2:

3.3 Training classifier

A step by step tutorial is carried out by University of Auckland Computer Science Department PhD.Mahdi Rezaei[5].

3.3.1 Data sets

For training the Haar-like classifiers, in order to improve the quality of the classifier, a large set of positive and negative sample images are required.

In this project, I used 942 positive images and 1100 negative images for training. I selected 442 images from the *Caltech Human Face (front) database*[6], about 200 images from the *BioID face database*[7]. For negative images, i have used about 900 images from the *Caltech 256 dataset*[8] and other from internet.

3.3.2 Training

With the face objects and negative images ready, we can start the training phase. I set the dimension of the positive objects to be 15×15 for a balance between processing time and resolution. The training takes 60 minutes to run through 15 stages. The generated classifier xml file is 82KB in size.

3.4 The Program

I use Visual Studio 2012 and OpenCV 2.3.1 for development.After the classifiers are created, The implementation for this part is to load the images in a loop one by one. Before going for face detection, apply an appropriate filtering technique to the current image in order to improve the image quality (e.g. histogram equalization - cvEqualizeHist). At last, we apply face detection. In order to do detection the image must be converted into gray-scale image first. In order to display the detected region in green rectangle, the image to be displayed needs to be converted into colored image if it is not.

3.5 Face detection result

The result for tests are acceptable given the limited number of training images. However the performance can be improved by adding more positive images for variety of people with different ages, size etc. and more negative images to reduce false positive detection.

Figures below shows the result of face detected in program:

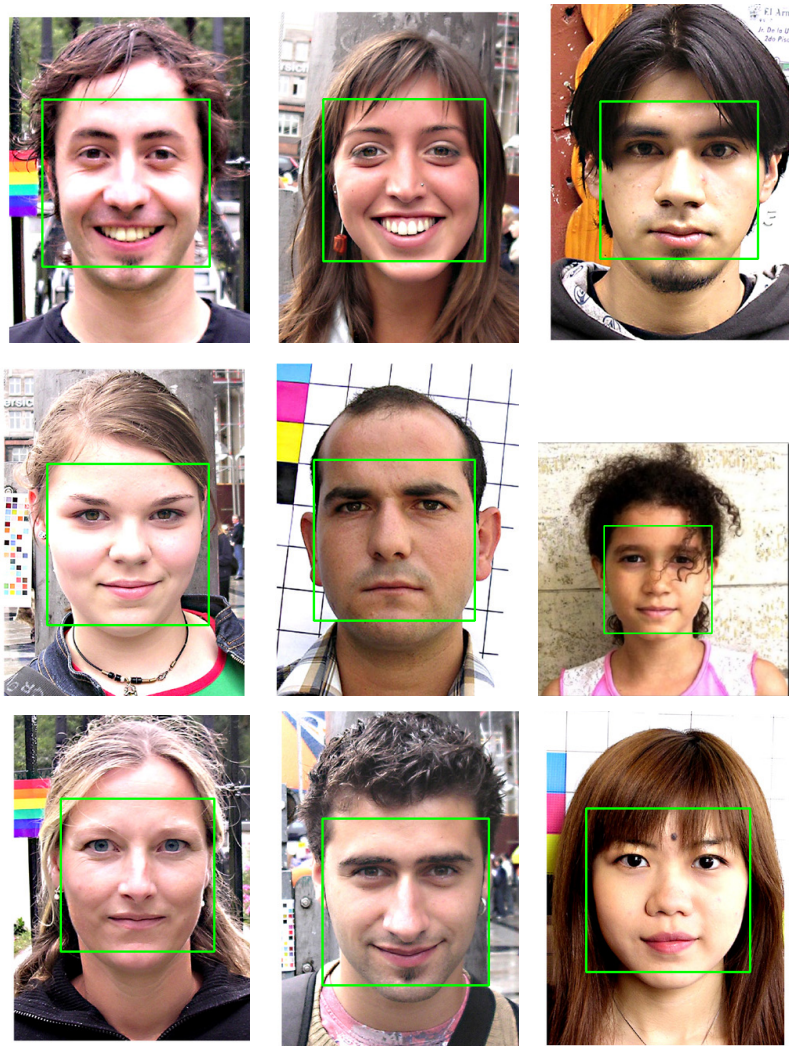


Figure 3.3: single face detection

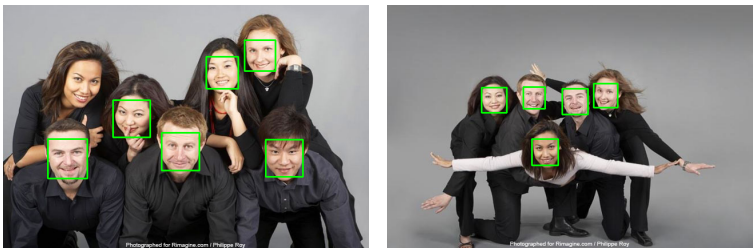


Figure 3.4: multi faces detection

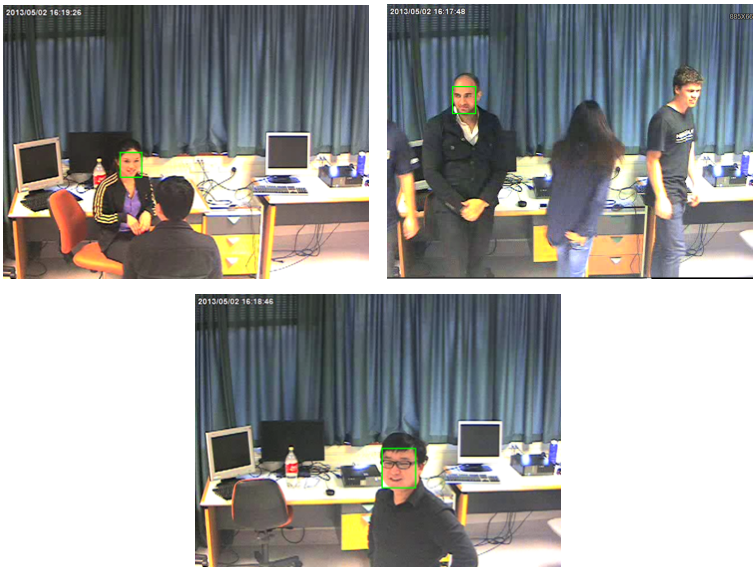


Figure 3.5: face detected in recorded video data

Chapter 4

Human Detection

For human whole body detection the method i am now currently learning is Random forests. In this chapter i would like to discuss the basic theory of it based on my understanding, and show some work done by others that similarly with the result that i hope to achieve in next semester.

4.1 Construction of single decision tree

The construction of decision tree is to build weak classifier for each node. For an image patch we can random choose 2 points p_1, p_2 and make a comparison of their grayscale I, here the decision tree is a binary tree. if $I(p_1) \geq I(p_2)$ output goes to 1, otherwise goes to 0, since this is just an easy comparison operation that contains no addition or multiplication, thus the speed of building these set of weak classifier is very quick. for an image patch size of 32×32, the number of possible combination of points is C^2_{1024}, means more than 5 million weak classifiers.

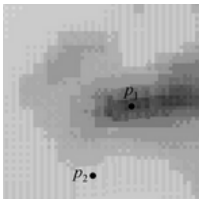


Figure 4.1:

So for the first node we can random choose 1000 of the points comparison combination and pick the best result one as the weak classifier on this node and record it. then depends on decision result the arriving image patch is passed to next level of tree and continue steps above. the depth of a tree can be about 10 to 12.

4.2 Forming Random Forests

Random forests have recently attracted a lot of attention in computer vision[10], it has been shown that assembling together several trees trained in a randomized way achieves superior generalization and stability compared to a single deterministic decision tree[11]. A typical random forest consists of a large set of decision trees[12], in this project we can have more than 500 trees build in forests. To classify human body from an input image, we can put the image patches down each of the decision trees in the forests. Each tree gives a classification and votes for this class. Then the forest chooses the classification having the most votes(over all the trees in the forests).

The forests error rate depends on two things:

- The *correlation* between any two trees in the forest. Increasing the correlation increases the forest error rate.
- The *strength* of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.[13]

Random forests have followed features:

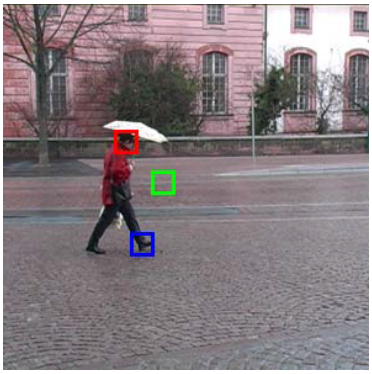
- It is very accurate comparing to other algorithms.
- It can be very efficient on large data bases.
- It gives estimates of what variables are more important in the classification.
- It can still be effective for estimating missing data and maintains accuracy.
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.

4.3 Whole human body detection using Hough Forests

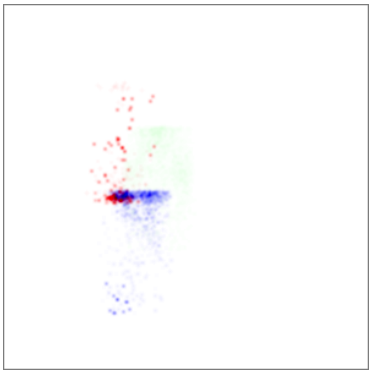
Hough Forests is based on Random Forests, in general the tree construction for hough forests follows the common random forest framework[14], that when detecting human body the detection of individual human body parts cast probabilistic votes for possible locations of the centroid of the whole human body, the detection hypotheses then correspond to the maxim of the Hough image that accumulates the votes from all parts. The whole detection process can thus be described as a generalized class-specific Hough transform[15].

Here I have done some research on using Hough forests for pedestrian detection and will have a basic explanation on how it works using Juergen Gall's work.[16]

Figure 4.2 shows an sample result of human detection from Juergen Gall article:'An Introduction to Random Forests for Multi-class Object Detection'[16], For each of the three patches emphasized in (a), the specific Hough forest casts a vote about the possible location of the human body centroid (b) (each color channel corresponds to the vote of a sample patch). Note the weakness of the vote from the background patch (green). After the votes from all patches are aggregated into a Hough image (c), the pedestrian can be detected (d) as a peak in this image.

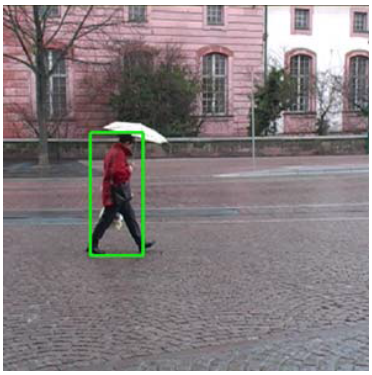


(a) Original image with three sample patches emphasized



(b) Votes assigned to these patches by the Hough forest

(c) Hough image aggregating votes from all patches



(d) The detection hypothesis corresponding to the peak in (c)

Figure 4.2:

Chapter 5

Schedule of my project

5.1 Work done so far

At the beginning of the project, after I am assigned to this topic which works on data modeling of motion detection for surveillance originally, I have chatted with both my academic supervisor and industry supervisor and make the final topic on human motion and gesture detection in surveillance video data.

During this semester I have done data capturing using IP camera provided by the company for later works on face detection, human body detection and gesture detection. I now have a basic understanding in theoretical parts of how to detect human or face using different algorithms after several weeks research, and accomplish the work on testing face detection using Haar-like features methods.

I'm now working on learning the algorithm of Hough Forests for whole human body detection and collecting experiences with constructing forest classifier.

5.2 Future works and plan

In the next semester I will target on performing good results in whole human body detection using a forest classifier, to achieve this more video data showing full human bodies may be required, also more practice on using C++ and OpenCV is required hardly.

Future works on understanding and classifying human motion and gesture are also considered depending on progress of human body detection.

Acknowledgments

I would like to thank University of Auckland Computer Science Department in Tamaki campus by the support to accomplish on video data captured. A special thank to my supervisor, professor Reinhard Klette for all support and advices on academic learning. I would also like TN Chan from Compucon by keep pushing me on working with my project weekly. I would like to thank Eric Song for his help on experiencing programming on Haar-like features. Last but not least, I would like to thank Juan Lin for her help on collecting programming experiences with hough forest classifier.

Bibliography

[1] Haar-like features in Wikipedia. http://en.wikipedia.org/wiki/Haar-like_features

[2] Paulo Menezes, Jos'e Carlos Barreto, Jorge Dias: Face tracking based on Haar-like features and eigenfaces. April 2004

[3] AdaBoost in Wikipedia. http://en.wikipedia.org/wiki/AdaBoost

[4] Viola, Paul and Michael Jones. Rapid object detection using boosted cascade of simple features. 2001

[5] Compsci775 tutorial on Haar-like classifiers using OpenCV by Mahdi Rezaei http://www.cs.auckland.ac.nz/~rklette/TeachAuckland.html/775/materials/775S22012%20Face%20&%20Eye%20Detection%20Tutorial%202.pdf

[6] Caltech Human Face (front). http://www.vision.caltech.edu/html-files/

[7] BioID face database. http://www.bioid.com/index.php?q=downloads/software/bioid-face-database.html

[8] Caltech 256 dataset. http://www.vision.caltech.edu/Image_Datasets/Caltech256/

[9] Random Forests in Wikipedia. http://en.wikipedia.org/wiki/Random_forest

[10] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. ICCV, pp. 1-8, 2007.

[11] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. Neural Computation, 9(7):1545-1588, 1997.

[12] J. R. Quinlan. Induction of decision trees. Machine Learning, 1(1):81-106, 1986.

[13] Random Forests Leo Breiman and Adele Cutler. http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm#intro

[14] L. Breiman. Random forests. Machine Learning, 45(1):5-32, 2001.

[15] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition, 13(2):111-122, 1981.

[16] Juergen Gall, Nima Razavi1,and Luc Van Gool. An Introduction to Random Forests for Multi-class Object Detection. 13 April 2012