

**AN APPROACH FOR SCENE MONITORING BASED ON BLOCK-
BASED DISTANCE MEASURE TECHNIQUE BETWEEN SEQUENCE
OF ESTIMATED HEAD POSE AND MOVEMENT DIRECTION**

By

CHOO CHE YON

A thesis submitted to the Department of IPSR,
Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Computer Science
May 2013

ABSTRACT

AN APPROACH FOR SCENE MONITORING BASED ON BLOCK-BASED DISTANCE MEASURE TECHNIQUE BETWEEN SEQUENCE OF ESTIMATED HEAD POSE AND MOVEMENT DIRECTION

Choo Che Yon

Computer vision and image processing are becoming popular in many areas such as security, business, advertising, psychology, etc. More and more smart systems have been developed that aim to aid or lessen the work load of humans, thus saving more valuable human resources for other purposes. When dealing with scene analysis, many visual attributes can be derived or extracted from the captured scenes for further analysis. Among all these visual attributes, head pose is one of the useful and important visual attributes to help in scene analysis since it usually coincides with the gaze direction.

This research is about the development of a system prototype which is capable of detecting and tracking humans that appear in the targeted scene and estimate the region of interest by estimating the head pose for monitoring purposes. There are only two modules in this project. The first module is *human detection and tracking* which is responsible for detecting (using HOG and SVM) and tracking (using optical flow) of humans that appear in the scene; whereas for the second module is *scene monitoring based on sequence of estimated head pose* which is responsible for estimating the head pose of

each tracked individual using NRBVS method, followed by monitoring process using those estimated head poses.

Experimental results show that the proposed system is able to estimate head pose of each individual who appeared in the scene using CCTV camera with less detail (blurred image). The use of estimated head poses in monitoring process yield a good result in detecting the head motion. Furthermore, more rules can be added to the proposed NRBVS method anytime to enhance the performance. It is processor friendly, thus, the proposed system is able to execute in real-time manner.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Dr. Khor Siak Wang and my co-supervisor, Dr. Wang Xin. Generously, you shared the enormous sea of knowledge, guide me, enlighten me, and motivate me in my study. Those were very much appreciated.

Without you, Dr. Khor Siak Wang, my mentor in image processing field research, I would not be able to experience this fun, excitement, and challenges in the said field. Once again, as my mentor, I would like to express my appreciation to you.

Secondly, I would like to express my gratitude to you, my precious classmates and dearest friends for the presence and help when I am doubtful and stressful. Those were very much touched and appreciated.

Third, for my lovely family, who dotes me the most, thanks for the support. I am proud to be the son of yours, and I am making you proud.

Special thanks to Dr. Lee Yun Li, who was my ex-supervisor. Thank you for the help and information shared for my master research. Those were very helpful.

Thank you, Mimos, which is an incorporation that collaborated with Utar. Thank you for providing tools and equipments for my research; thank you for the supervision and knowledge shared my Mimos's representations;

thank you for letting me in participating the data collection process in the real industrial environment, those experiences were priceless.

Last but not least, I would like to express my love to How Ming Hui, who is my girlfriend. Thanks for being there for me; thanks for the courage; thanks for the love.<3

Thank you.

APPROVAL SHEET

This thesis entitled “**AN APPROACH FOR SCENE MONITORING BASED ON BLOCK-BASED DISTANCE MEASURE TECHNIQUE BETWEEN SEQUENCE OF ESTIMATED HEAD POSE AND MOVEMENT DIRECTION**” was prepared by CHOO CHE YON and submitted as partial fulfilment of the requirements for the degree of Master of Computer Science at Universiti Tunku Abdul Rahman.

Approved by:

KHOR SIAK WANG

(Dr. Khor Siak Wang)

Date: 2 May 2012

Head of Programme/Supervisor

Department of Information Systems

Faculty of Information Communication and Technology

Universiti Tunku Abdul Rahman

**FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 2 May 2013

SUBMISSION OF THESIS

It is hereby certified that **CHOO CHE YON** (ID No: **10UEM07396**) has completed this thesis/dissertation entitled “AN APPROACH FOR SCENE MONITORING BASED ON BLOCK-BASED DISTANCE MEASURE TECHNIQUE BETWEEN SEQUENCE OF ESTIMATED HEAD POSE AND MOVEMENT DIRECTION” under the supervision of Dr. Khor Siak Wang (Supervisor) from the Department of Information Systems, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my thesis in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

CHOO CHE YON
(*Choo Che Yon*)

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name CHOO CHE YON

Date 2 May 2012

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
APPROVAL SHEET	vi
SUBMISSION OF THESIS	vii
DECLARATION	viii
LIST OF TABLES	xiii
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS/NOTATION/GLOSSARY OF TERMS	xx

CHAPTER

1	INTRODUCTION	1
1.1	Overview	1
1.1.1	Potential Applications for Various Visual Attributes	2
1.1.2	Problems and Challenges for Smart Systems	3
1.2	Problem Statements	5
1.2.1	Lack of Ability to Detect Humans in the Targeted Scene	5
1.2.2	Lack of Ability to Track Individual in the Scene	5
1.2.3	Lack of Ability to Monitor the Scene Based on the Sequence of Estimated Head Pose	6
1.3	Objectives	6
1.3.1	Detect Humans Appeared in the Scene	6
1.3.2	Tracking Individual Appeared in Scene	6
1.3.3	Monitor and Analysis on Head Poses	7
1.4	Project Scope	7
1.4.1	Module 1-Human Detection and Tracking	8
1.4.2	Module 2-Scene Monitoring Based on Sequence of estimated Head pose	8
1.5	Thesis Organization	8
1.6	Expected Research Outcome	9
1.7	Possible Applications	10
1.8	Conclusion	11
2	LITERATURE REVIEW	12
2.1	Introduction	12
2.2	Literatures for Human Detection	13
2.2.1	Appearance-Based Approach	14
2.2.2	Feature-Based Approach	17
2.2.3	Appearance and Feature-Based Approach	19
2.3	Literatures for Human Tracking	19

2.3.1	Filtering Technique	20
2.3.2	Mode Seeking Method	21
2.4	Literatures for Head Pose Estimation	22
2.4.1	Appearance-Based Approach	23
2.4.2	Feature-Based Approach	26
2.4.3	Manifold-Learning and Dimensionality Reduction-Based Approach	27
2.4.4	Hybrid Approach	30
2.5	Conclusion	31
3	SYSTEM DESIGN AND METHODOLOGY	33
3.1	Introduction	33
3.2	System Overview	33
3.3	Software and Hardware Requirement	36
3.4	Conclusion	37
4	SYSTEM IMPLEMENTATION	38
4.1	Introduction	38
4.2	Preprocessing (P1)	38
4.3	Blob Filtering (P2)	40
4.4	Motion Blob Segmentation (P3)	42
4.5	Human Detection (P4)	43
4.5.1	Training of Human classifier	44
4.5.2	Classifying of Human	49
4.5.3	Sliding Window	51
4.6	Human Tracking (P5)	53
4.6.1	Optical Flow	55
4.7	Direction of Motion Calculation (P6)	58
4.8	Head Localization (P7)	59
4.8.1	Foreground Subtraction	60
4.8.2	Determine Shoulder Line	61
4.8.3	Localize Head Position	63
4.9	Head Pose Estimation (P8)	65
4.9.1	Colour Normalization	67
4.9.2	Region Division	73
4.9.3	Feature Extraction	74
4.9.4	Voting Process	75
4.9.4.1	Nested Rule-Based Voting System (NRBVS) Engine	76
4.9.4.2	Grouping of Head Pose	77
4.9.4.3	Construction of Rules	85
4.9.4.4	Voting Mechanism	93
4.10	Head Pose Monitoring (P9)	96
4.10.1	Detecting Head Motion	98
4.10.2	Smoothing of Sequence of Head Poses	102
4.10.3	Adjustment of Sequence of Head Poses	106
4.10.4	Gaze Direction Calculation	109
4.11	Conclusion	110

5	RESULT AND DISCUSSIONS	111
5.1	Introduction	111
5.2	Testing Environment	112
5.2.1	Testing Equipment	112
5.2.2	Testing Tools	113
5.2.3	Testing Data	113
5.2.3.1	Testing Data Preparation	115
5.2.3.2	Testing Data Specification	122
5.3	System Evaluation Based On Test Cases	122
5.3.1	Evaluation Using Test Case 1	133
5.3.1.1	Evaluation of P4	134
5.3.1.2	Evaluation of P5	136
5.3.1.3	Evaluation of P6	138
5.3.1.4	Evaluation of P7	140
5.3.1.5	Evaluation of P8	141
5.3.1.6	Evaluation of P9	143
5.3.2	Evaluation Using Test Case 2	147
5.3.2.1	Evaluation of P4	148
5.3.2.2	Evaluation of P5	150
5.3.2.3	Evaluation of P6	151
5.3.2.4	Evaluation of P7	153
5.3.2.5	Evaluation of P8	154
5.3.2.6	Evaluation of P9	157
5.3.3	Evaluation Using Test Case 3	161
5.3.3.1	Evaluation of P4	162
5.3.3.2	Evaluation of P5	163
5.3.3.3	Evaluation of P6	165
5.3.3.4	Evaluation of P7	168
5.3.3.5	Evaluation of P8	168
5.3.3.6	Evaluation of P9	171
5.3.4	Evaluation Using Test Case 4	177
5.3.4.1	Evaluation of P4	178
5.3.4.2	Evaluation of P5	180
5.3.4.3	Evaluation of P6	182
5.3.4.4	Evaluation of P7	184
5.3.4.5	Evaluation of P8	184
5.3.4.6	Evaluation of P9	190
5.3.5	Evaluation Using Test Case 5	194
5.3.5.1	Evaluation of P4	195
5.3.5.2	Evaluation of P5	196
5.3.5.3	Evaluation of P6	199
5.3.5.4	Evaluation of P7	199
5.3.5.5	Evaluation of P8	199
5.3.5.6	Evaluation of P9	200
5.3.6	Summary of Evaluation	200
5.4	Conclusion	206
6	FUTURE WORKS AND CONCLUSION	208
6.1	Introduction	208

6.2	Contribution	208
6.3	Future Work	209
6.4	Conclusion	212

REFERENCES **214**

APPENDICES **228**

A.	Publication – A novel NRBVS approach for long distance head pose estimation	228
B.	Publication – Letter of acceptance for manuscript in Appendix A	235

LIST OF TABLES

Table		Page
3.1	Summary of system processes	36
3.2	Software and hardware requirements	36
4.1	Comparison of skin colour detection result using original and colour balanced image	70
4.2	Grouping of head poses	77
4.3	Summary of head pose group characteristic (skin)	87
4.4	Summary of head pose group characteristic (dark)	87
4.5	Candidate selection for voting session 2	95
4.6	Possible outcome from detecting head motion	99
5.1	Profile of actors	116
5.2	Predefined scheme of act	121
5.3	Characteristics of testing videos	122
5.4	Summary of evaluation of human tracking algorithm for test case 1	138
5.5	Summary of result from P9 for test case 1	145
5.6	Summary of evaluation of human tracking algorithm for test case 2	151
5.7	Summary of result from P9 for test case 2	158
5.8	Summary of evaluation of human tracking algorithm for test case 3	165
5.9	Summary of result from P9 for test case 3	173
5.10	Summary of evaluation of human tracking algorithm for test case 4	182
5.11	Summary of result from P9 for test case 4	192
5.12	Summary of evaluation of proposed system	200

LIST OF FIGURES

Figure		Page
1.1	Location of overhead wide-angle camera	3
1.2	Example of captured scene	3
2.1	Summary of different techniques used when dealing with the scene monitoring	13
3.1	Overall system model	34
4.1	Process flow of P1	39
4.2	Different sizes of motion blob	41
4.3	Segmented motion blob	42
4.4	Human classifier training process	45
4.5	Example of positive samples	46
4.6	Example of negative samples	47
4.7	Blocks (green) and cells (red) in the detection window	47
4.8	Process flow of human detection	50
4.9	Example of sliding window's process (first and second step)	52
4.10	n^{th} sliding window	53
4.11	Scaled sliding window	53
4.12	Process flow of human tracking	54
4.13	Original image from video source	56
4.14	Motion backprojection image	56
4.15	8 possible movement directions of tracked human	58
4.16	(a) Input image of P7 (head and shoulder); (b) True head region (result of P7)	60

4.17	Motion blob of head shoulder image (from figure 4.16a)	61
4.18	Detection of shoulder line (a) temporary shoulder line at last row of image; (b) temporary shoulder line at second last row; (c) detected shoulder line	62
4.19	Valid shoulder line region (red rectangle)	62
4.20	Starting, optimum and neighbour state of head location searching	64
4.21	Distance between top motion pixel and shoulder line	64
4.22	Localization of head location	65
4.23	8 categories of head pose	66
4.24	P8 process flow	66
4.25	Colour balancing using gray world assumptions (a) before (b) after	68
4.26	7 regions of true head region and region numbering	73
4.27	Input and output of voting process	75
4.28	Comparison of 8 categories of head poses (skin colour)	78
4.29	Comparison of 8 categories of head poses (dark colour)	78
4.30	Comparison of head poses with categories 1, 3, 5, and 7 (skin colour)	79
4.31	Comparison of head poses with categories 1, 3, 5, and 7 (dark colour)	80
4.32	Graph patterns of head pose group A (skin colour)	81
4.33	Graph patterns of head pose group A (dark colour)	82
4.34	Graph patterns of head pose group B (skin colour)	82
4.35	Graph patterns of head pose group B (dark colour)	83
4.36	Graph patterns of head pose group C (skin colour)	83

4.37	Graph patterns of head pose group C (dark colour)	84
4.38	Graph patterns of head pose group D (skin colour)	84
4.39	Graph patterns of head pose group D (dark colour)	85
4.40	Possible outputs from each voting session	86
4.41	Example rule generated	89
4.42	Example rule generated	90
4.43	Example rule generated for candidate head pose 1 & 2	91
4.44	Example rule generated for candidate head pose 1, 2 & 3	92
4.45	Voting process flow	94
4.46	P9 process flow	101
4.47	Example of smooth curve	103
4.48	Example of non-smooth curve	103
4.49	Smoothed curve	104
4.50	Smoothing process flow	105
4.51	Head pose over time graph	107
4.52	Head pose over time graph	107
4.53	Adjusted head pose over time graph	108
4.54	Adjusted head pose over time graph	108
5.1	Configuration of testing equipment	113
5.2	Example scenes with entry and exit points (in red arrow)	115
5.3	Frames from testing video of test case 1	134
5.4	Screen shots of program execution on testing video of test case 1	134
5.5	Human detection processing time for test case 1	135

5.6	<i>ETWPF</i> evaluation over time graph for human tracking algorithm on test case 1	136
5.7	Error of motion calculation for P6	138
5.8	Screen shots of system execution which each block was formed	139
5.9	Estimated head pose for test case 1	141
5.10	Actual head pose for test case 1	142
5.11	Error of head pose estimation for P8	142
5.12	Smoothed estimated head poses	144
5.13	Frames from testing video of test case 2	147
5.14	Screen shots of program execution on testing video of test case 2	148
5.15	Human detection processing time for test case 2	149
5.16	<i>ETWPF</i> evaluation over time graph for human tracking algorithm on test case 2	150
5.17	Error of motion calculation for P6	152
5.18	Screen shots of system execution which each block was formed	152
5.19	Estimated head pose for test case 2	154
5.20	Actual head pose for test case 2	155
5.21	Error of head pose estimation for P8	155
5.22	Smoothed estimated head poses	157
5.23	Frames from testing video of test case 3	161
5.24	Screen shots of program execution on testing video of test case 3	161
5.25	Human detection processing time for test case 3	162
5.26	<i>ETWPF</i> evaluation over time graph for human tracking algorithm on test case 3	164
5.27	Error of motion calculation for P6	166

5.28	Screen shots of system execution which each block was formed	167
5.29	Estimated head pose for test case 3	169
5.30	Actual head pose for test case 3	169
5.31	Error of head pose estimation for P8	170
5.32	Smoothed estimated head poses	172
5.33	Frames from testing video of test case 4	177
5.34	Screen shots of program execution on testing video of test case 4	177
5.35	Human detection processing time for test case 4	179
5.36	<i>ETWPF</i> evaluation over time graph for human tracking algorithm for subject A on test case 4	180
5.37	<i>ETWPF</i> evaluation over time graph for human tracking algorithm for subject B on test case 4	180
5.38	Error of motion calculation for P6	182
5.39	Screen shots of system execution which each block was formed	183
5.40	Estimated head pose for subject A for test case 4	185
5.41	Estimated head pose for subject B for test case 4	185
5.42	Actual head pose for subject A for test case 4	186
5.43	Actual head pose for subject B for test case 4	186
5.44	Error of head pose estimation for P8 for subject A	188
5.45	Error of head pose estimation for P8 for subject B	189
5.46	Smoothed estimated head poses for subject A	191
5.47	Smoothed estimated head poses for subject B	191
5.48	Frames from testing video of test case 5	194
5.49	Screen shots of program execution on testing video of test case 5	195

5.50	<i>ETWPF</i> evaluation over time graph for human tracking algorithm on test case 5	197
------	---	-----

LIST OF ABBREVIATIONS

Etc	Et cetera, “and other things”
HOG	Histogram of oriented gradient
SVM	Support vector machine
NRBVS	Nested rule-based voting system
CCTV	Closed circuit television
HOT	Histogram of template
LSVM	Linear support vector machine
CBCL	Center for biological and computational learning
MIT	Massachusetts Institute of Technology
TSL	Temporary shoulder line
RGB	Red, green and blue
BC	Best case
NC	Normal case
WC	Worst case
ATM	Automated teller machine
SVD	Singular value decomposition
KNN	K-nearest neighbour
LDA	Linear discriminant analysis
PCA	Principle component analysis
CCA	Canonical correlation analysis
EGM	Elastic graphs matching
SCI	Sparsity concentration index
ANN	Artificial neural network
HOSVD	High order singular value decomposition

MICA	Multilinear independent component analysis
GE	Graph embedding
LLE	Local linear embedding
LPP	Locality preserving projection
PSS	Pose-specific subspace
NL	New location
LK	Lucas-kanade
FPPW	False positive per window
FNPW	False negative per window
FPR	False positive rate
FNR	False negative rate
TPR	True positive rate
TNR	True negative rate
ETWPF	Euclidean distance error of tracked window per frame
aETWPF	Average of Euclidean distance error of tracked window per frame
aFPR	Average of false positive rate
aTPR	Average of true positive rate
D&T	Dalal and Triggs

CHAPTER 1

INTRODUCTION

1.1 Overview

Computer vision and image processing are becoming popular in many areas such as security (Lin, Chang, & Pai, 2009), business (Lablack, Zhang, & Djeraba, 2008), advertising (Lablack, Zhang, & Djeraba, 2008), psychology (Tsechpenakis, Metaxas, Adkins, Kruse, Burgoon, Jensen, Meservy, Twitchell, Deokar, & Nunamaker, 2005; Shan, Tsechpenakis, Metaxas, Jensen, & Kruse, 2005; Lablack & Djeraba, 2008), etc. More and more smart systems have been developed that aim to aid or lessen the work load of humans, thus saving more valuable human resources for other purposes.

When dealing with scene analysis, many visual attributes can be derived or extracted from the captured scenes for further analysis. Among all these visual attributes, head pose is one of the useful and important visual attributes to help in scene analysis. Some of the potential applications are described in section 1.1.1.

1.1.1 Potential Applications for Various Visual Attributes

A lot of potential applications can be developed based on the derived or extracted visual attributes. For example, an application which detects the region of interest of a customer walking into a shop, can be used for business purpose which helps to improve the shop layout by placing their new or promotion products over the region of interest of customers; An application which calculates the number of people who attracted to a particular advertisement (such as poster, banner, etc.) can measure the effectiveness of the advertisement; An application which to analyse the behaviour of an individual who appeared in the scene, can be used for security purpose, etc.

Those smart systems described need the use of overhead wide-angle cameras in order to obtain the information of the entire scene. This kind of camera is usually mounted at the corner just below the ceiling and it is best if the camera is capturing the whole focus area as shown in figure 1.1 and figure 1.2. However, even though with the proper mounting of cameras and good video quality, there are still many pertinent problems or challenges that impede the good accuracy rate and performance. Some of these challenges are explained in the following section.



Figure 1.1 Location of overhead wide-angle camera

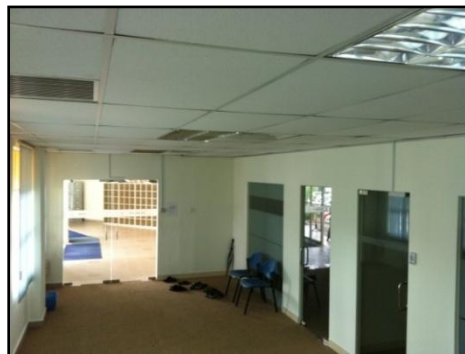


Figure 1.2 Example of captured scene

1.1.2 Problems and Challenges for Smart Systems

The captured video scene from CCTVs may not produce video scene of good quality which will further cause the captured scenes to become blurred, objects become smaller, and what make it worst is the inconsistency brightness and contrast of the input video caused by the variant weather conditions (such as sunny day, cloudy day, etc.). This boosts the difficulty to interpret the signal information received by the smart system. For this kind of scenario and environment setting, estimating head pose is a challenging task. Blurred images obtained can cause the detection of face features (such as eye, ears, etc.) to fail and so for head pose estimation. Colour information is more

reliable compared to face features, shapes, etc. Therefore, estimating head pose in this environment setting has to be further researched and developed.

Before the head pose estimation, the system has to be able to identify human from other objects. Many scholars have done researches and a lot of workable solutions for human detection and tracking have been developed (Zhang & Liang 2010; Tang & Goto, 2010; Beleznai & Bischof, 2009; Bhuvanewari & Abdul, 2009; Denman, Fookes, & Sridharan, 2009; Tang & Goto, 2009b; Tang & Goto, 2009a; Thombre, Nirmal, & Lekha, 2009; Thanh, Ogunbona, & Li, 2009; Ding, Xu, Cui, Sun, & Yang, 2009; Chakraborty, Rudovic, & Gonzalez, 2008; Jia & Zhang, 2007; Dalal & Triggs, 2005; Beleznai, Frühstück, Bischof, & Kropatsch, 2004). However, most of them are only capable of producing good result under extreme controlled environment such as constant lighting condition, no occlusion for human detection, etc. (Thanh, Ogunbona, & Li, 2009; Tang & Goto, 2009b). Such constraints limit the usability of the smart systems in real world. To make the system more versatile, the human detection has to be complemented by human tracking. Therefore, to make the real time smart system a reality, a robust yet fast human detection and tracking approach has to be developed.

This research project titled “an approach for scene monitoring based on block-based distance measure technique between sequence of estimated head pose and movement direction” is aimed to develop a system prototype which is capable of detecting and tracking humans that appear in the targeted scene

and estimate the region of interest by estimating the head pose for monitoring purposes.

1.2 Problem Statement

For monitoring purpose, many CCTVs have been installed in strategic locations. However, these CCTVs lack many intelligent functions to automate the monitoring process. The major missing functions are described below.

1.2.1 Lack of Ability to Detect Humans in the Targeted Scene

Most surveillance systems installed in public places such as shopping complexes, airport, office, etc. do not have the functions or lack the ability to detect the presence of humans. Such detection is crucial to help in alerting the users on possible occurrence of activities such as snatching, loitering, etc.

The system must be able to identify human from other objects in order to continue for further analyse of the humans in the scene such as the way they look, the path they took, etc..

1.2.2 Lack of Ability to Track Individual in the Scene

Tracking is another challenge in computer vision. It is essential for surveillance system to know the entry and exit locations of a person in the

scene. Without a proper labelling of each individual, surveillance system would be pretty much useless such that it does not know where a detected person is coming from and the direction he or she is heading to.

1.2.3 Lack of Ability to Monitor the Scene Based on the Sequence of Estimated Head Pose

Monitoring process cannot be done effectively without head pose information. Without this information, the system will not be able to know the region of focus of an individual in the scene which is an important feature in many aspect such as surveillance, advertising, etc..

1.3 Objective

1.3.1 Detect Humans Appeared in the Scene

To detect the presence of humans in the targeted scene.

1.3.2 Tracking Individual Appeared in Scene

A proper tracking of humans is essential for surveillance system. This objective aims to track individuals that appear in the scene.

1.3.3 Monitor and Analysis on Head Poses

The proposed project is to estimate the head pose of individuals appear in the scene. The estimated head pose can be collected and analysed for monitoring purposes to know where the individual is looking at or what has attracted his or her attention for a period of time.

When the camera used is not in high quality recording, the head information retrieved could appear to be blur and small. To overcome such limitation, a new method will be proposed to estimate the region of interest of each individual.

1.4 Project Scope

This research project is to detect and track humans (no occlusion) that appear in the scene. In addition, the sighted region of each individual is estimated and the head movement is being monitored to continuously extract the region/object of focus. Basically, this proposed system consists of two modules:

Module 1: Human detection and tracking.

Module 2: Scene monitoring based on sequence of estimated head pose.

1.4.1 Module 1 – Human Detection and Tracking

This module is mainly for detecting and tracking of people that appear in the scene. The detection is only covered up to humans with visible head and shoulder. Objects occlusion and crossover are excluded from this research as this is the support module (main focus is in module 2). Once an individual is detected and calculated, it is constantly being tracked and the head and shoulder region of the detected individual will be sent to module 2 for further analysis.

1.4.2 Module 2 – Scene Monitoring Based on Sequence of Estimated Head Pose

This module aims to estimate the head pose of each individual. All the estimated head pose will be collected for monitoring purpose to continuously extract the region/object of focus. This is the very first stage (prototype) of research on the novel head pose estimation technique – NRBVS (refer to chapter 4, section 4.9.4.1), hence not all the real world variants are covered. Here, several constraints were highlighted. Subjects must have hair, and it should be short (normal male hair style) and dark coloured. Furthermore, they should be having uniform hair style (normal male hair style, not fully or partially (>10%) covering face). Last but not least, the subjects should have Asian skin colour (light brown, some samples can be found in chapter 5, section 5.2.3.1 - Profile of actors).

1.5 Thesis Organization

This thesis is organized into 6 chapters. Chapter 1 will be the introduction to the research project, overview, problem statements, objectives, project scope, and research of this project will be stated.

Chapter 2 will focus on the literature review. Literatures related in the area of research that being studied will be recorded in this chapter. A discussion of literatures will be stated.

Chapter 3 is about the system design and methodology of the proposed system. This chapter will review about how the system's structure, architecture, process flow, functionalities of processes and also software and hardware requirements of the system.

Chapter 4 is the system's implementation which includes a discussion of the structure of the system, detail of implementation of modules and the analysis of hardware and software requirements.

Chapter 5 will discuss the result of the proposed system. The result of performance evaluation of the system will be shown and discussed in this chapter.

Chapter 6 will discuss the future work and conclude this research project.

1.6 Expected Research Outcome

This research work is expected to develop software modules that enable the detection and tracking of humans in captured scenes through CCTVs in real life environment. A novel head pose estimation and analysis approach will also be introduced which capable of estimating the head pose of individuals and analysing the sequence of estimated head poses of individuals with blurred video frames. Developed modules would work together to form a system prototype which is capable of detecting human from other objects and estimates the sighted region of the detected person and monitor the movements of head of each individual.

1.7 Possible Applications

The developed software modules, when installed in CCTVs are able to provide many benefits in diverse application areas. Some of them include:

- Snatch thief detection system, which detects the possible suspects of snatch thief by identifying the possible motions before snatching activity such as looks around the surrounding, etc.
- Shop lot layout enhancement system, which identifies the region of interest of customers walking into the shop where will be the suitable place to display promotion or new products.

- Advertisement efficiency evaluation system, which evaluates the efficiency of an advertisement (such as banner, poster, etc.) by the counting the number of viewers.
- Examination inspection system, which checks the examinees who turn his/her head around (could be cheating) in the examination hall.
- Surveillance system, which detects the possible intruders and gives alert to the appropriate personnel.

1.8 Conclusion

Proposed system consists of two modules including human detection and tracking and scene monitoring based on sequence of estimated head pose. These two modules can be used in most of the smart systems in many fields such as surveillance, advertising, psychology, etc. With the successful implementation of this project, it would bring benefit to the society.

As for scene monitoring based on sequence of estimated head pose process, it is to continuously check the focus of attention of an individual in order to know his/her object of interest when he/she is in the scene. The sequence of head pose will also be analysed for detecting possible suspicious head pose (ulterior motives of an individual) such as look around during exam in examination hall, look around in a bank, etc. It is essential for the system to know the focus of attention of an individual in order to obtain his/her object of interest. For instance, in advertising field, to check which location is the best

to put poster, etc. This information is useful in such a way that we can detect the possible security threat such as snatching activities and possible victim in the scene if further analysis is commenced; or the business company can maximize the effectiveness of the advertisement and product placement to increase the business profit (Isti & Annury, 2005).

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Recent research in computer vision and image processing has increasingly focused on detecting and recognizing humans in scene contents (Shabani, Ghaemina, & Shokouhi, 2010; Ding, Xu, Cui, Sun, & Yang, 2009). Due to the various postures of humans, movements, and thousands of activities that can be performed, the challenges for detecting human and understanding the human activities are extreme. For instance, a security system without human detection or the ability to analyse human activities is equivalent to futile. To confront all those extremes, lots of precious researches had been done by scholars throughout the world. As highlighted in chapter 1, the current research work comprises 2 major modules, namely human detection and tracking and Scene monitoring based on sequence of estimated head pose. In this chapter, all the relevant research works by other authors will be summarized. The following diagram provides a general view on the different techniques used when dealing with the scene monitoring.

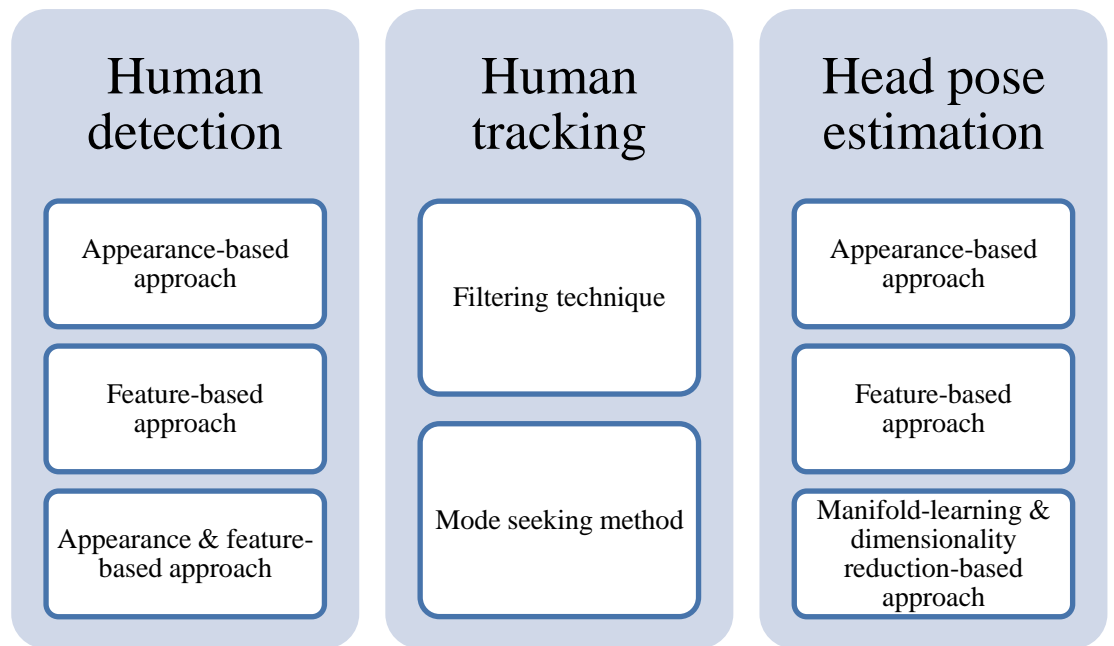


Figure 2.1 Summary of different techniques used when dealing with the scene monitoring

2.2 Literatures for Human Detection

One of the most important system components in any intelligent systems which related to human (such as human-computer interactions, surveillance systems, etc.) is human detection. Without a proper human detection module, any further processing of the system is impossible. Recently, human detection has become popular among researchers because of the inherent challenges. For instance, occlusion of humans, posture variance of humans, size variance of humans, crowded scenarios, etc. Most of the proposed algorithms work well under controlled environment and would experience high degree of degradation in performance with the presence of any of the challenges above.

To detect the presence of humans, 3 approaches are normally used, namely appearance-based approach, feature-based approach, and appearance and feature-based approach. The difference between the first 2 approaches is that the appearance-based approach uses the whole human to train the classifier and try to search for the whole human within any given testing image; for feature-based approach, instead of using the whole human, it uses the distinctive features of a human such as hands, head, legs, etc. to detect human. Some of the approaches are also based on the combination of both appearance-based and feature-based method. Integrating both approaches could result in strength combination of both methods for better human classification.

2.2.1 Appearance-Based Approach

The benefit of using appearance-based approach is the simplicity of the method which usually uses one classifier to differentiate human from other objects and would have the global picture of the whole human. However, due to the invariant shape of humans, there will always be background pixels within the region of interest that could affect the classification result. In year 2005, Dalal and Triggs (2005) proposed the Histogram of Oriented Gradient (HOG) features and Support Vector Machine (SVM) classifier for human detection which has widely regarded as a standard benchmark tests by many researchers doing similar research work. The major drawbacks of this approach are the slow speed of processing due to the dense scan while detecting and the large feature size.

To overcome the speed limitation of human detection, a fast detection approach was proposed by Jia and Zhang (2007) which used the HOG features and cascade classifier for detecting humans. Tackling the crowded scenes scenario, Beleznai and Bischof (2009) proposed a fast human detection method using shape-based matching and motion cues. Proposed method comprised of two major parts. For the first major component, Beleznai and Bischof (2009) used the template-based matching technique on the pre-processed contour image to locate the possible human. The second major component was the local shape descriptor which inferred the human locations in images of absolute difference between current image and background. The output from the two components was combined to locate humans. Another template matching approach that included matching and verification stages was proposed by Thanh, Ogunbona, and Li (2009) which showed promising result in the literature. For the matching stage, a set of predefined template images that described human postures were used to find the best matching description for the image within a detection window. According to the authors, the best matching template would be the one having the shortest distance to the image within the detection window. The matched template would then be passed on to the verification stage where the template would be checked to see whether it reached high degree of confidence that the image contains human. If the template has high credibility and high matching score, then the image was classified as human.

More recently, a more powerful and discriminative feature, Histogram of Template (HOT) was proposed by Tang and Goto (2010) that cooperated

with a SVM classifier for human detection. Instead of using gradients (as proposed by Dalal and Triggs (2005)), Tang and Goto (2010) used pre-defined templates for feature calculation. All templates were in the form of 3×3 grid and there were 3 pixels in each template. According to the authors, if the value of the pixels satisfied certain predefined function, they would conclude that the central pixel meets that particular template. Finally, the count of pixels which met a template would be collected to construct a histogram of template (HOT) for human detection. In addition, the experimental result showed that the HOT feature is more discriminative yet having smaller feature size than HOG feature.

Despite all the powerful classifier and feature representation, pre-processing of the image is far more significant in most image processing approach. Zhang and Liang (2010) proposed a new method to detect moving objects based on background subtraction method and used the shape analysis method for human detection. The shape analysis made use of the features of motion region for detection. If the object area was larger than the pre-defined threshold and the aspect ratio of the object region satisfied the pre-defined ratio, then the moving object would be considered as a moving human. Experimental result shows that this approach was fast and simple and it was able to remove shadows.

2.2.2 Feature-Based Approach

Feature-based methods refer to the used of distinctive features such as head, body, hand, leg, etc. for detecting humans. Usually, they equipped with multiple classifiers in which each classifier responsible for detecting one designate part of human. The detection results from each classifier would then be combined to form the final result which is human.

Compared to appearance-based approach, feature-based approach is able to solve the occlusion problem in a more effective and better way. It ensures the view invariant and human invariant for human detection. However, with the advantages listed above, additional processing is required such as feature extraction and classification process for part based human detection which detects human parts instead of detecting the whole human body. Chakraborty, Rudovic, and Gonzalez (2008) designed three example-based detectors that specifically aimed to detect three distinctive parts of humans including head, arms and legs using HOG features and SVM classifiers. The detected parts were then combined at a later time for human detection. However, due to the heavy computation of feature extraction and classification, it gradually slowed down the speed of the proposed algorithm.

Several efficient improvements on human detection based on Dalal and Triggs' algorithm had been made by Baranda, Jeanne, and Braspenning (2008). The algorithm aims to detect humans by detecting human upper body to reduce the occlusion in indoor environment. The first improvement was

using linear-SVM (LSVM) as classifier where each dimension of the training and testing vectors was scaled to be between 0 and 1. For the second improvement, it was to reduce the number of HOG feature vectors for each object representation by removing parts of the feature vectors that possessed less discriminative power. Experimental result showed that the proposed method could reduce the feature length by 50% and led to comparable performance and better efficiency in terms of speed and memory usage.

Another approach aimed to increase the overall speed of detection was from Ding, Xu, Cui, Sun, and Yang (2009) who proposed a cascade SVM approach for detecting head-shoulder for human detection using HOG features. First stage of SVM rejected most of the background region with second stage of SVM detected the head-shoulder part of human. The experiments proved the robustness and accurate detection even in crowded scenes.

To further address the speed and occlusion problems, another part-based approach for human detection was proposed by Tang and Goto (2009a) who used HOG features and boosting SVM classifier in which each classifier worked only to detect specific human parts such as head and shoulder, torso, leg, left of body and right of body. The cascade of classifier was designed so that for an object to be detected as human, it must pass all the cascades. In addition, area information was used for measuring the relative depth of humans to detect the human's location for occlusion detection. In this way,

even some part of the human was not detected, it still could pass the classifier and detected as occluded human.

2.2.3 Appearance and Feature-Based Approach

Some of the human detection approaches combined both appearance and feature-based methods. For instance, Tang and Goto (2009b) detected standing and moving humans with two different methods. For standing human, the appearance based feature, HOF feature and SVM classifier were used; whereas for detecting moving human, the motion based feature was extracted from the optical flow field that was calculated from the two consecutive frames. It could capture the characteristics of human motion well. The motion-based feature was about the relative motion of human body. It focused on local motion of human such as motion of legs and arms. This approach was tested and the result showed that the combination of the two features used achieved higher detection rate when compared to other features. In addition, this feature was hardware friendly; acceleration of the hardware was possible to boost the detection rate.

2.3 Literatures for Human Tracking

Other than human detection, tracking is also one of the popular research areas for video processing and it is the fundamental problem for computer vision based research (Thombre, Nirmal, & Lekha, 2009;

Ghaemina, Shabani, & Shokouhi, 2010; Bhuvaneswari & Abdul, 2009; Rashid, Remya, & Wilscy, 2009; Shabani, Ghaemina, & Shokouhi, 2010; Denman, Fookes, & Sridharan, 2009; Zhang, Sun, Guang, Wang, Xie, & Shang, 2010; Huang & Sun, 2010). Detecting and tracking of people in a video sequence is important for real-time system such as automated video surveillance and monitoring systems. However, robust human tracking is very challenging due to the fast motion, occlusion, illumination variation, background clutters, real-time restriction, etc.

2.3.1 Filtering Technique

Among various methods, filtering techniques, such as kalman filter (Thombre, Nirmal, & Lekha, 2009) and particle filter (Ghaemina, Shabani, & Shokouhi, 2010) is one of the favourable methods for visual tracking. This filtering technique is a set of mathematical equations that provides efficient computational means to estimate the state of a process by minimizes the mean square error. Basically, filtering technique is a prediction of possible location for object of interest in the following frames based on a measurement model. It is very powerful in the sense that it can estimate the past, present and even future states (Thombre, Nirmal, & Lekha, 2009). But before the prediction, the object of interest and the dynamics have to be defined. However, without good prior information of the object of interest, these filtering techniques might not be as reliable as it stated. Hence, mode seeking methods such as mean-shift (Bhuvaneswari & Abdul, 2009; Rashid, Remya, & Wilscy, 2009; Shabani, Ghaemina, & Shokouhi, 2010), optical flow (Denman, Fookes, & Sridharan,

2009), and cam-shift (Zhang, Sun, Guang, Wang, Xie, & Shang, 2010; Huang & Sun, 2010) have become more useful in this situation.

2.3.2 Mode Seeking Method

Mean-shift algorithm is similar to filtering techniques when matching for the next location of the object of interest, which is to obtain the minimum distant between the current information and its possible locations in the next frame. What makes it different from filtering techniques is that mean-shift algorithm will locate the object of interest at each frame rather than initialize at start and leave the tracking to the filtering algorithm.

Despite the strength of mean-shift algorithm, it requires an initial detection of the moving objects (by using motion extraction such as background subtraction, etc.) while optical flow can be used to segment overlapping objects using velocity (occlusion) by continuously performs motion segmentation process and computes optical flow simultaneously (Denman, Fookes, & Sridharan, 2009).

There are still some flaws in mean-shift algorithm (Zhang, Sun, Guang, Wang, Xie, & Shang, 2010; Huang & Sun, 2010), mean-shift is a nonparametric density estimation algorithm, it iteratively calculates the probability density along the direction which the gradient of probability increases and finally converges to the peak of the probability density. But, it may run into trouble when the object is similar to the background or when the

occlusion occurs. It is the same for optical flow which is very sensitive to the colour. Moreover, mean-shift could not adjust the tracking window size.

To enhance the robustness in human tracking, Zhang, Sun, Guang, Wang, Xie, and Shang (2010) made use of the cam-shift instead of mean-shift for tracking. Cam-shift is an improved algorithm of mean-shift algorithm. It calls mean-shift iteratively and adaptively changes window size with target. However, the cam-shift algorithm tends to lose target or reiterate tracking for overlapped targets. Furthermore, when the object of interest has more than one hue, the tracker will tend to track the most significant part and leave the small part untracked, similarly for complex background. With the successful implementation of cam-shift in tracking, Huang and Sun (2010) proposed the combination of cam-shift and kalman filter technique for human tracking in dynamic scene. First, kalman filter is used to obtain the center position of the object of interest, and set the surrounding to be the search space for cam-shift algorithm to find the peak, which is the predicted location of object of interest in next frame.

2.4 Literatures for Head Pose Estimation

Head pose is one of the useful and important visual attributes to help in scene analysis since it usually coincides with the gaze direction (Vatahska, Bennewitz, & Behnke, 2007). It is essential for the system to know the focus-of-attention of an individual in order to obtain his/her object of interest. For

instance, a robber is constantly focusing on his/her target that is currently withdrawing cash from an automated teller machine (ATM). This information is crucial in such a way that we can predict the possible snatch thief and possible victim in the scene if further analysis is launched.

Head pose estimation methods can generally be grouped into four main categories: appearance-based methods, feature-based methods, manifold-learning & dimensionality reduction-based approach and hybrid methods.

2.4.1 Appearance-Based Approach

Appearance-based methods formulate the head pose estimation problem as a pattern classification problem on image feature space (Dong, Tao, Xu, & Oliver, 2009). It uses the whole face or head image rather than some distinctive features such as nose-tips and eyes for head pose estimation. Number of classes defines the accuracy of the pose estimation that can be achieved. Most of the appearance-based methods suffer from the redundancy of the information caused by pose, illumination, expression, occlusion and background. Different testing subjects would also result in different experimental results. However, despite those weaknesses, appearance-based method is less expensive in computation compared to feature-based and dimensionality reduction based method in such a way that, it does not require additional steps such as detection of distinctive features (nose-tip detection, eye detection, etc.) or reduce the dimension of features by mapping it into the feature subspace.

Given the success of the face detection algorithm (Rowley, Baluja, & Kanade, 1998; Tsai, Cheng, Taur, & Tao, 2006; Chen, Hsu, & Chien, 2007; Viola & Jones, 2001a; Osuna, Freund, & Giroso, 1997), Huang, Shao, and Wechsler (1998) introduced a detector array for face pose estimation in 1998. The detector array used 3 SVMs for detecting 3 discrete yaw angles to determine the pose. A more recent version of detector array, 5 FloatBoost classifiers (Viola & Jones, 2001b) was introduced by Zhang, Hu, Liu, and Huang (2007) for head pose estimation using multiple cameras. Each detector in the array was attuned for the detection of a single view of face. These approaches require much effort in training and data preparation stage. A part from that, the main drawback of these approaches might produce systematic problem if 2 or more detectors in the detector array were attuned to very similar pose. An image is assigned as a positive data for a detector must be a negative data for another detector. It is not clear whether a prominent detection approach can be produced from the training process when positive and negative data are similar.

In year 2008, Lablack, Zhang, and Djeraba (2008) used a template-based method adopting the singular value decomposition (SVD), Gabor Wavelets as features, K-nearest neighbor (KNN) and support vector machine (SVM) as classifiers for head pose estimation. A thorough comparison between classifier and feature used was carried out for performance evaluation. The advantage of template-based method is that only positive samples were required for training stage and training set could be expandable

any time. However, this method only estimated a discrete head pose and the more templates added to the training set, the more sufficiency suffered.

Asides from the template-based method, Ma, Shan, Chen, and Gao (2008) proposed a novel way to estimate the head pose by estimating the head yaw rotations based on a asymmetry of 2-D facial appearance. Similar to other appearance-based methods; the initial step was to extract features from the face image. To eliminate the influence of lighting and noise, multiple scale 1-D Gabor filters was applied followed by conducting the Fourier transform on the input signal. Next, the transformed image signals were combined as asymmetry features. Linear Discriminant Analysis (LDA) was finally applied to reduce the feature dimension for better classification. Head pose estimation is not only for security purpose, Zhang, Zheng, Mu, and He (2009) proposed a method for driver assistance system which uses dual histograms of Isophote features and KNN classifier for head poses estimation. The Isophote feature was invariant to illumination and contrast which made it work well in outdoor scene.

Recently, sparse representation has been used to improve the performance of head pose estimation. Ma and Wang (2010) proposed a novel method that used the block-based sparse representation classifier in order to reduce the influence of background for head pose estimation. Similar to Ma, Shan, Chen, and Gao (2008), the feature extraction was based on the principle component analysis (PCA) for dimensionality reduction and LDA for discriminant analysis. Lastly, the sparsity concentration index (SCI) was used

to predict the head pose. The experimental result showed the effectiveness of this method.

2.4.2 Feature-Based Approach

Feature-based methods refer to the use of distinctive features such as nose-tips, eye, ears, etc. for head pose estimation. It usually comes with a face or head detector to locate the location of face or head of an individual followed by feature detection such as eye detection or nose-tip detection to extract those distinctive features. This kind of technique requires the clarity of the video source. It is not suitable for long distance estimation and blurred video stream.

The head pose is estimated based on the information of those features (Vatahska, Bennewitz, & Behnke, 2007; Tu, Fu, & Huang, 2009). For the method by Vatahska, Bennewitz, and Behnke (2007), supervised learning approach was implemented in which the head pose was estimated based on the three rotational angles (roll, yaw, and pitch) which were outputs of an artificial neural network (ANN) classifier that used the position of the distinctive features (eyes, ears, mouth and nose-tip) as inputs. In addition, kalman filters were used to track each facial feature and each rotational angle over time for robustness real-time implementation.

Similarly, Tu, Fu, and Huang (2009) also used feature-based approach for head pose estimation. Instead of detecting all the face features used by

previous authors (Vatahska, Bennewitz, & Behnke, 2007; Tu, Fu, & Huang, 2009) Tu, Fu, and Huang (2009) uses the nose-tip of the subject's face in coarse-to-fine manner in Laplacian pyramid as the feature after the subject's face location was found. The head pose was estimated simultaneously using tensorposes model which was built based on high order singular value decomposition (HOSVD) and multilinear independent component analysis (MICA), and naïve PCA subspace models.

Other than facial features, geometrical information could also be useful in head pose estimation. Human perception of head pose relies on the cues such as the deviation of nose angle and deviation of the head from bilateral symmetry (Wilson, Wilkinson, Lin, & Castillo, 2000). Raytchev, Kimura, Yoda, and Sakaue (2010) make use of the geometrical configuration of the head and real-time supervised learning algorithm for head pose estimation. The geometry-based head pose estimation algorithm was used to obtain the pose then the obtained pose parameter was used as supervised information during learning stage which realizes the real-time learning of the model for head pose estimation.

2.4.3 Manifold-Learning and Dimensionality Reduction-Based Approach

To obtain a better representation of face images, Raytchev, Yoda, and Sakaue (2004) and Yun and Huang (2006) suggested the use of high-dimensional feature space of face image data as a set of geometrical related

points lying on a smooth low-dimensional manifold in feature space. This suggestion led to the use of manifold-learning & dimensionality reduction-based approach for head pose estimation where linear or non-linear embedding of the face images was used to delete irrelevant information. The best thing about this approach was the simplification of the overall classification process and resulting in obtaining a more accurate classification. Among the literatures that applied this approach, Raytchev, Yoda, and Sakaue (2004) proposed an Isomap-based non-linear manifold learning procedure for data representation of faces and head pose estimation. The extended Isomap model used was able to map high-dimensional data into low-dimensional feature space and more faithful subspace embedding for view representation. From this representation, a pose parameter map relating the input faces images to view angles was learnt and used for estimation of head pose. On the other hand, Yun and Huang (2006) estimated the head pose using a supervised graph embedding (GE) analysis. First construct the neighbourhood weighted graph in the sense of supervised locally linear embedding (LLE), followed by calculating the unified projection in a closed-form solution based on the GE linearization. New data calculated was then projected into the embedded low-dimensional subspace and finally a KNN classification for head pose estimation. The experimental result showed that with small number of training set, GE achieved high accuracy for head pose estimation.

Biased manifold embedding had also been proposed by Balasubramanian, Ye, and Panchanathan (2007), that made use of the pose labels before determining the low-dimensional embedding. In biased manifold

embedding framework, the poses which were similar were maintained nearer to each other; while those which poses were farther were placed farther irrespective of the identity of the individual in the low-dimensional embedding. This biased manifold embedding framework proved the improvement of performance of the pose estimation process significantly.

In year 2008, Hu and Huang (2008) tested different subspace learning methods including PCA, LDA, Locality Preserving Projection (LPP), and Pose-Specific Subspace (PSS) for head pose estimation. The result shows that the PSS subspace learning yields the best among others.

Based on the non-linear embedding method (Hu, Huang, & Ranganath, 2005), Lin, Chang, and Pai (2009) made some improvement on it to estimate the head pose by skipping the embedding process using the Isomap to eliminate the computation needed for the evaluation and normalization of the unified embedding space. Another improvement made was to replace the training process for interpolative mapping function with a supervised training scheme.

Considering about the feature subspace, Liu, Lu, and Luo (2009) defined a new and more compact representation of images for head pose estimation. The head image was processed to enhance the facial feature and remove the unrelated information by skin colour model and Laplacian of Gaussian transform. The processed image was then used to construct an eigenpose subspace in which the decision function was found.

Studies by Foytik, Asari, Youssef, and Tompkins (2010) showed that the PCA fails to identify the true relationship between the observed space (feature subspace) and the pose variable; and LDA neglects the continuation properties of poses variation and treats the poses separately with a discrete multi-class approach. A better method to estimate head pose is proposed by Foytik, Asari, Youssef, and Tompkins (2010) which used Canonical Correlation Analysis (CCA) in which the pose variation is treated as important component for head pose estimation. The feature is represented by a manifold in feature space. Furthermore, a Gabor Filters were used to promote pose sensitivity as additional input for the estimation process.

2.4.4 Hybrid Approach

Hybrid approach for head pose estimation make used of multiple independent techniques and fused the estimation from each techniques used into one result. In this case, the system gains information from multiple cues that could increase the accuracy.

In year 2001, Sherrah and Gong (2001) introduced a method which includes appearance-based template matching with geometric cues and particle filtering technique for pose estimation and tracking. Other similar works which make used of the tracking technique in pose estimation includes the work by Ba and Odobez (2004) which incorporate the use of appearance-based and tracking method. This method requires the prior knowledge about the head orientation, followed by tracking of the pose for variation detection.

Their work was later refined for multiple cameras and far-field imagery (Ba & Odobez, 2007).

Incorporate tracking technique in pose variation detection could help in head pose estimation. However, prior information is required which is the position and the orientation of the head. The tracking technique could tell even a slight movement that had been made by the head. The main drawback of incorporate tracking technique in head pose estimation is the bad prior information given by previous process to the tracking technique. Bad information could lead to false detection or estimation for subsequent frames in video scene.

Other than incorporate tracking techniques in head pose estimation, Wu and Trivedi (2008) and Wu and Pedersen (2004) combined the use of feature-based method, Elastic Graphs Matching (EGM)(Kruger, Potzsch, & von der Malsburg, 1997; Lades, Vorbruggen, Buhmann, von der Malsburg, Wurtz, & Konen, 1993) with the manifold embedding estimation technique for head pose estimation.

2.5 Conclusion

Literature reviewed provides lots of useful information for this project. Strengths and weaknesses of the methods by the scholars are analysed and a comparison among the methods is done. Based on the strengths and

weaknesses of each method, a suitable method can be chosen to implement for this project.

Considering the accuracy, robustness, speed of computation, as well as computer resource usage, method by Ding, Xu, Cui, Sun, and Yang (2009) was chosen for detecting humans and method by Denman, Fookes, and Sridharan (2009) was chosen for tracking of detected humans.

For human detection, the chosen method effectively reduced the feature size which results in faster feature extraction and classification process while preserving the discriminating power of the original method by Dalal and Triggs (2005). In addition, the chosen method could detect partially occluded human which is useful in real world environment.

For human tracking, some popular tracking methods such as kalman filter, particle filter, etc., are unable to cope with situation where there is irregular movement pattern (e.g. sudden change of direction). On the other hand, method by Denman, Fookes, and Sridharan (2009) which adopted optical flow algorithm is very sensitive to motion which makes it suitable to track moving objects in the targeted scene as any slight movement can be identified.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 Introduction

Systems planning and design are essential before the implementation stage of the proposed system. Other than defining the structure, process flow and characteristics of the proposed system; it is also able to help in identifying the potential problems that could possibly occur in the design of system.

This chapter includes the information of the system design, showing the overall process flow of the system with detailed implementation given in chapter 4, functionality of processes, as well as the software and hardware requirements of the proposed system.

3.2 System Overview

This system is divided into two modules, namely “human detection and tracking” and “Scene monitoring based on sequence of estimated head pose”. These modules have their own functionalities which contribute to meet the objectives of this project. The overall system model and the breakdown of the modules are shown in figure 3.1.

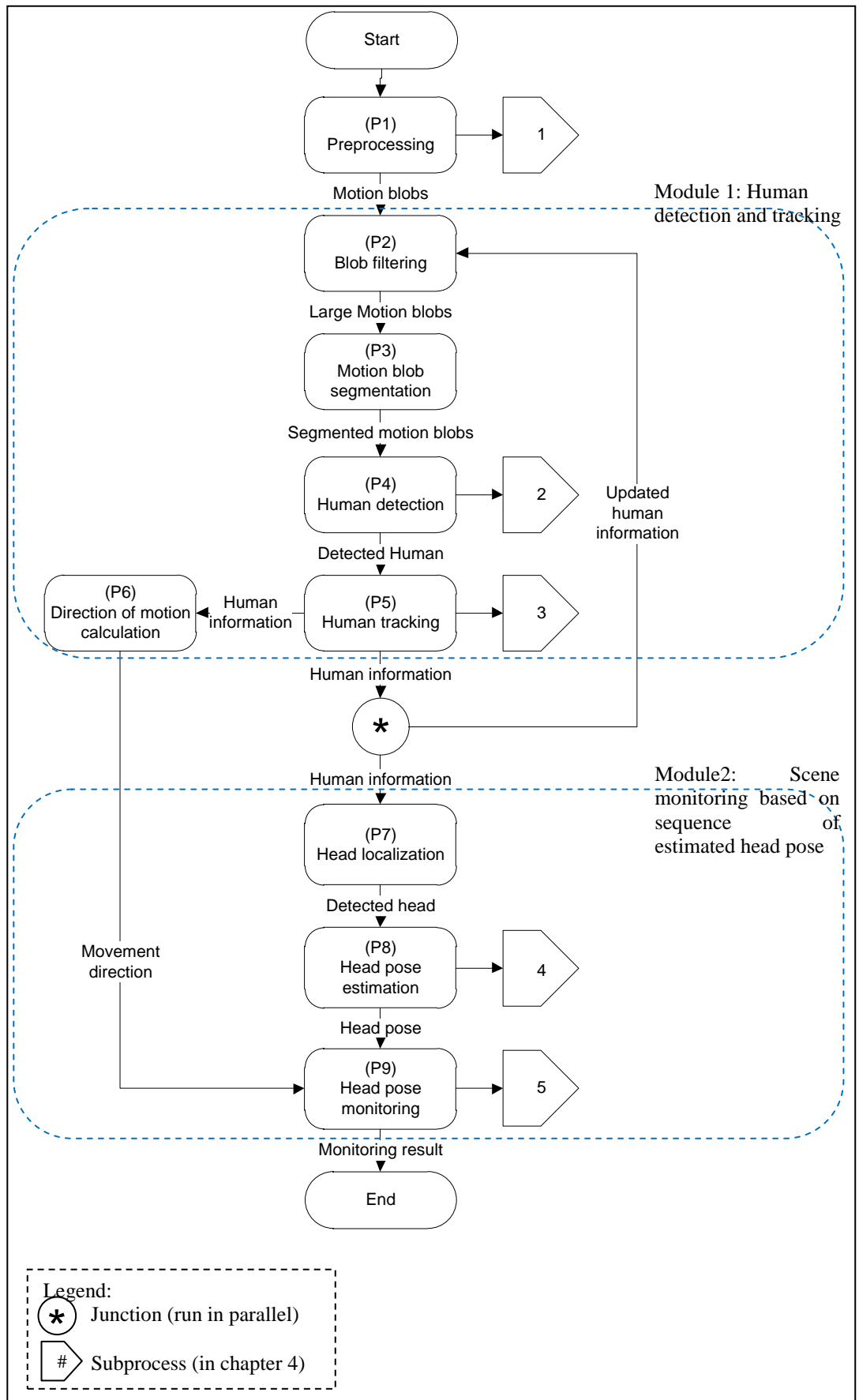


Figure 3.1 Overall system model

The proposed system comprises 9 processes namely “preprocessing”, “blob filtering”, “motion blob segmentation”, “human detection”, “human tracking”, “direction of motion calculation”, “head localization”, “head pose estimation”, and “head pose monitoring”. These processes are denoted as “P1”, “P2”, “P3”, “P4”, “P5”, “P6”, “P7”, “P8”, and “P9” respectively.

The design of system is focused on speed and functionality which brings out the best performance for the proposed system. First, the system must acquire input (P1) and eliminates unimportant or unnecessary processing as much as possible (P2 and P3) before P4 since P4 requires high computer resources.

After P4, each detected humans are assigned to a tracker to eliminate confusions on the data acquired by the following processes. The tracker will ensure that the information acquired is bound to the tracked human only.

With tracker as information manager, the system can proceed to acquire information for head pose monitoring (P9). P9 requires the information of movement direction (P6) as well as the head poses of the tracked human (P8). Since P8 needs the head location in order to estimate the head pose, P7 is processed before P8. Hence, the system process flow is designed and arranged as shown in figure 3.1.

The details of the implementation of each processes is discussed in the next chapter. The functionalities of each process are summarized in table 3.1.

Table 3.1 Summary of system processes

Process name	Symbol	Description
Preprocessing	P1	Preprocess the acquired video frame taken from a camera source. Processes in (P1) include resizing frame, gray scale conversion, cropping of motion blob, etc. on the acquired video frame.
Module 1: Human detection and tracking.		
Blob filtering	P2	Filter out motion blobs which size are too small for being a human in the targeted scene.
Motion blob segmentation	P3	Segment the motion blob into regions.
Human detection	P4	Perform human detection (based on human parts) in the segmented motion regions.
Human tracking	P5	Track the location of detected humans in the scene by finding the strong edge points defined in (P5) in the subsequent frames.
Direction of motion calculation	P6	Calculate the direction of motion for (P9) based on the information of (P5).
Module 2: Scene monitoring based on sequence of estimated head pose.		
Head localization	P7	Localize the head position in the detected human blob which obtained in (P4).
Head pose estimation	P8	Estimate the head pose of the head position localized in (P7).
Head pose monitoring	P9	Monitor the sequence of head poses (estimated in (P8)) to obtain his/her sighted region.

3.3 Software and Hardware Requirement

Table 3.2 Software and hardware requirements

Name	Description
Software	
Windows XP 32-bit/64-bit and Windows 7 RC Ultimate 32-bit/64-bit	A recommended working platform for Microsoft Visual Studio to work efficiently.

To be continued...

...continued

Table 3.3 Software and hardware requirements

Name	Description
Software	
Microsoft Visual Studio 2008	Software to build, compile and debug the program.
OpenCV 2.0	An image processing library which provides necessary function for image processing.
WinAVI Video Converter v9.0	Used to convert the captured data into supported AVI format as only AVI format video is supported by the OpenCV 2.0 library.
Hardware	
Intel Core 2 Duo or faster processor	Faster processing speed is required as real time video processing requires high processing power.
1GB RAM (Recommended 2GB RAM or higher)	Necessary for temporary image processing storage.
Web Camera which can support up to 640 x 480 resolution and 15 frames per second and above	Used to capture the video frame, the higher the resolution, the clearer the picture but the processing speed is slower.
Monitor	Used to display the result.
Keyboard and Mouse	Input devices for controlling the system by sending signal from the user.

3.4 Conclusion

This chapter explains about the proposed system design, the system structure, process flow, as well as the software and hardware requirements of the proposed system. Detailed description of the methodology of each module will be discussed in the following chapter.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 Introduction

This chapter will discuss the implementation of the system in detail including a detail discussion of the processes of the system about how the processes work, what are the input and output of the processes, etc.

4.2 Preprocessing (P1)

The main objective of P1 is to detect the presence of motions and segment the detected motions into regions. This is to reduce the computation process of P4 by reducing the human detection search space so that the P4 can operate only on the motions blob itself. In this proposed system, static background subtraction method is used for detecting motions. The process flow of P1 is illustrated in figure 4.1.

The first step in P1 is to resize the input image frame to standard size (320×240 pixels). Note that the system will only process half of the total captured frames, i.e. out of 2 consecutive frames, only the first will be processed (based on the assumption that there will be no significant changes of

information in two consecutive frames. It is logical to implement the preprocessing stage based on the assumption above since human eyes can process 10-12 separate images per second and the frame per second used here is 30 (Wikipedia, 2011; Answers.com, 2011; 100fps.com, 2011)). Next is to convert the image into gray scale format. This gray scale conversion is to reduce the effect of chrominance influence in background subtraction method to obtain the motions.

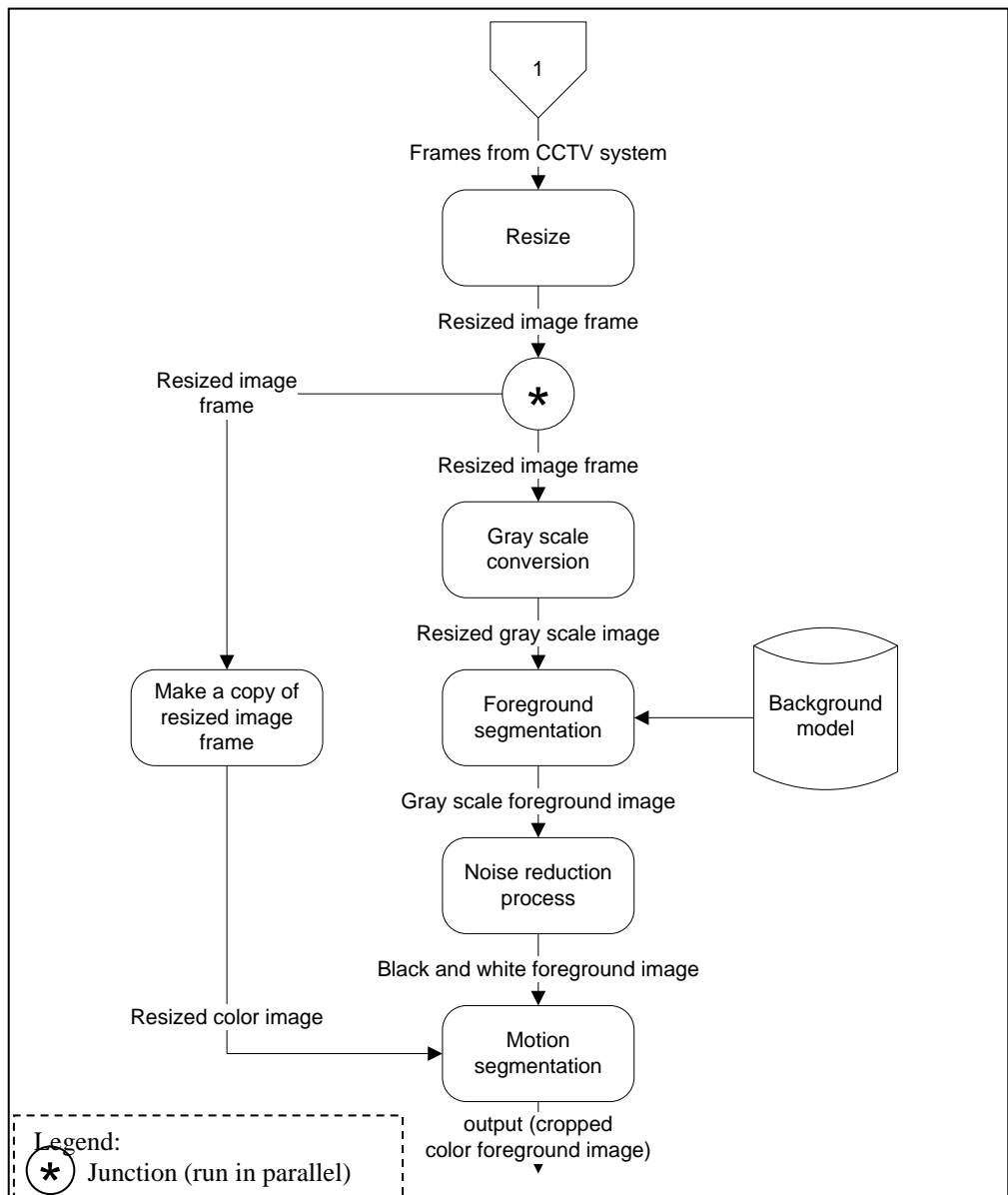


Figure 4.1 Process flow of P1

After the acquisition of first frame, also known as background image and input image i.e. subsequent frames are ready (both in gray scale format and standard size), then background subtraction algorithm is performed to segment the foreground which is the motions. However, the segmented foreground may not be the motion regions which are useful to the system. Such motions include noises produced by illumination change, shadow or the noises due to the low image quality captured through CCTV system. These noises may introduce false detection and they may even slow down the overall computational speed of the proposed method. To minimize the noises, the segmented foreground image is undergoing a simple noise reduction process, i.e. binarization and the final foreground region is obtained. Finally, the output of P1 (coloured foreground region) is the coloured motion blobs retrieved from the copied colour input image. These cropped colour foreground image may contain any moving objects that were not present in the background image.

4.3 Blob Filtering (P2)

The output from P1 would be a collection of blobs in different sizes from a given frame. Among those sizes, only the large and medium motion blobs will remain and proceed to the next process P3 and the small motion blobs will be discarded. The large and medium motions may contain humans, hence they are useful for the system. Whereas when the motion blob size is too small (which means the object is too far away or it could just be some

noises), the proposed methodology might fail to function. For instance, a small motion blob with 10 pixels in a 320×240 pixels image, it would be too small to identify whether it is a human, non-human, or noises. In this case, just ignore the motion blob would save the resources rather to identify whether it contains human and which might introduce false alarm. Some examples of the motion blob sizes are shown as in figure 4.2.

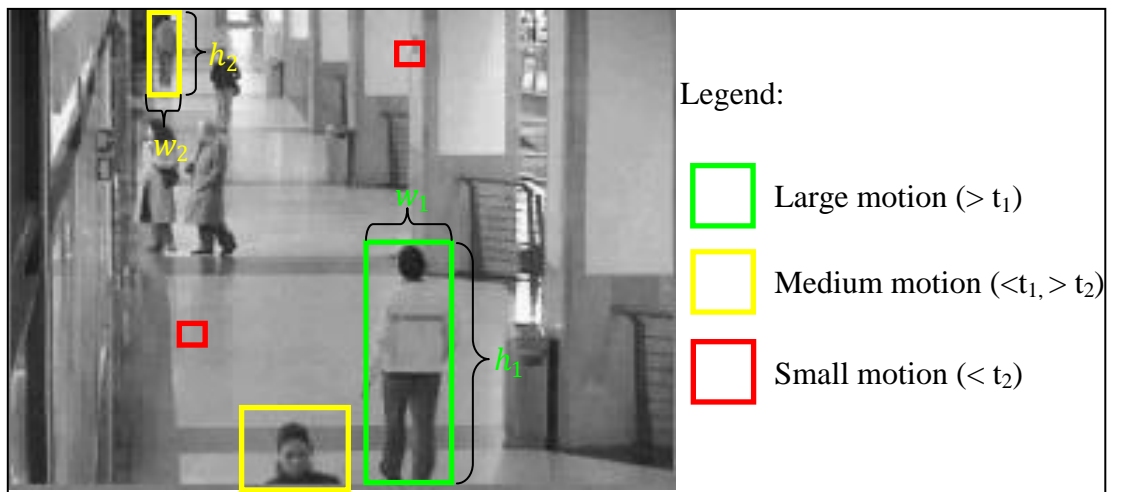


Figure 4.2 Different sizes of motion blob

In order to discard those irrelevant or unused blobs, the following parameters are defined.

$$width_{t1} = \frac{w_1}{2} \quad (4.1)$$

$$height_{t1} = \frac{h_1}{2} \quad (4.2)$$

$$width_{t2} = \frac{w_2}{2} \quad (4.3)$$

$$height_{t2} = \frac{h_2}{2} \quad (4.4)$$

where w_1 and h_1 is the width and height of a complete human figure captured from the closest distance; where w_2 and h_2 is the width and height of a complete human figure, normally captured from the furthest distance from the camera.

From the formulas 4.1 to 4.4, notice that the width and height of t_1 and t_2 are half (but not full) the size of humans in the scene as shown in figure 4.2. This is to account for variant size of humans that may appear in the targeted scene.

4.4 Motion Blob Segmentation (P3)

With all the irrelevant blobs removed from P2, the retained blobs will be further processed here. Those blobs with individual size $> t_2$ and $< t_1$, they will be halved into segments in order to facilitate the processing in P4.

The two segments are illustrated in figure 4.3 where the upper part (red rectangle) of the blob is used for locating head and shoulder (because P4 identifies human by detecting head and shoulder part only and head and shoulder part exists in upper body).



Figure 4.3 Segmented motion blob

However, if the individual blob size $> t_1$, the size is retained since it may contain more than one object.

4.5 Human Detection (P4)

P4 is essential for this proposed system as it works to filter out all other irrelevant objects i.e. non-human objects and hence highlighting the useful information which enables other sub-systems to function that could strive towards reaching the intended objectives. Therefore, histogram of oriented gradient (HOG) is used as features for human detection due to its powerful discriminative ability (Ding, Xu, Cui, Sun, & Yang, 2009) and support vector machine (SVM) is used as human classifier.

In P4, from a detected human figure, only “head and shoulder” is used for human detection. There are two reasons for choosing “head and shoulder” for P4. The first reason is because the “head and shoulder” of a human can be considered as the most discriminative visual attribute (compared to human torso, human hands, etc.) when using HOG for human detection (Ding, Xu, Cui, Sun, & Yang, 2009). Human torso is excluded due to the reason that the torso is nearly rectangle in shape and it is less discriminative to be used as a feature in HOG since rectangular objects are commonly found in any captured environment. Similarly, human upper limbs are also excluded because they can be easily hidden or merged with the torso part if the person wearing a long

sleeve shirt. In addition, the variations of arm poses make them hard to detect. The same goes for human legs.

As for the second reason for choosing “head and shoulder” part for P4 is to shorten the processing time required for identifying humans that appear in the scene. Compared to detecting the whole human, detecting only a portion of the entire human figure is faster (due to the smaller HOG feature size), and it can also solve partially humans occlusion problem when part of the body is blocked by others (Ding, Xu, Cui, Sun, & Yang, 2009).

The training of human classifier and the use of human classifier in detecting the presence of humans are described in the following sub-sections.

4.5.1 Training of Human Classifier

There are two human detectors in this proposed project which are of type linear. One is trained to recognize small human from the motion blobs (32×32 dimension) and another is trained for larger human size (64×64 dimension). The terms “small SVM” and “large SVM” are referred as SVM classifier trained for 32×32 dimension and 64×64 dimension respectively.

The use of two classifiers can enhance the human detection rate. Relying on a single classifier (32×32 dimension) would produce more false detection result especially when motion blob is large; and relying on classifier of size 64×64 dimension will result in poor detection rate especially on

smaller motion blob. Training of both small and large SVM is exactly the same except for the input feature size. The steps for training the SVM classifier is shown in figure 4.4.

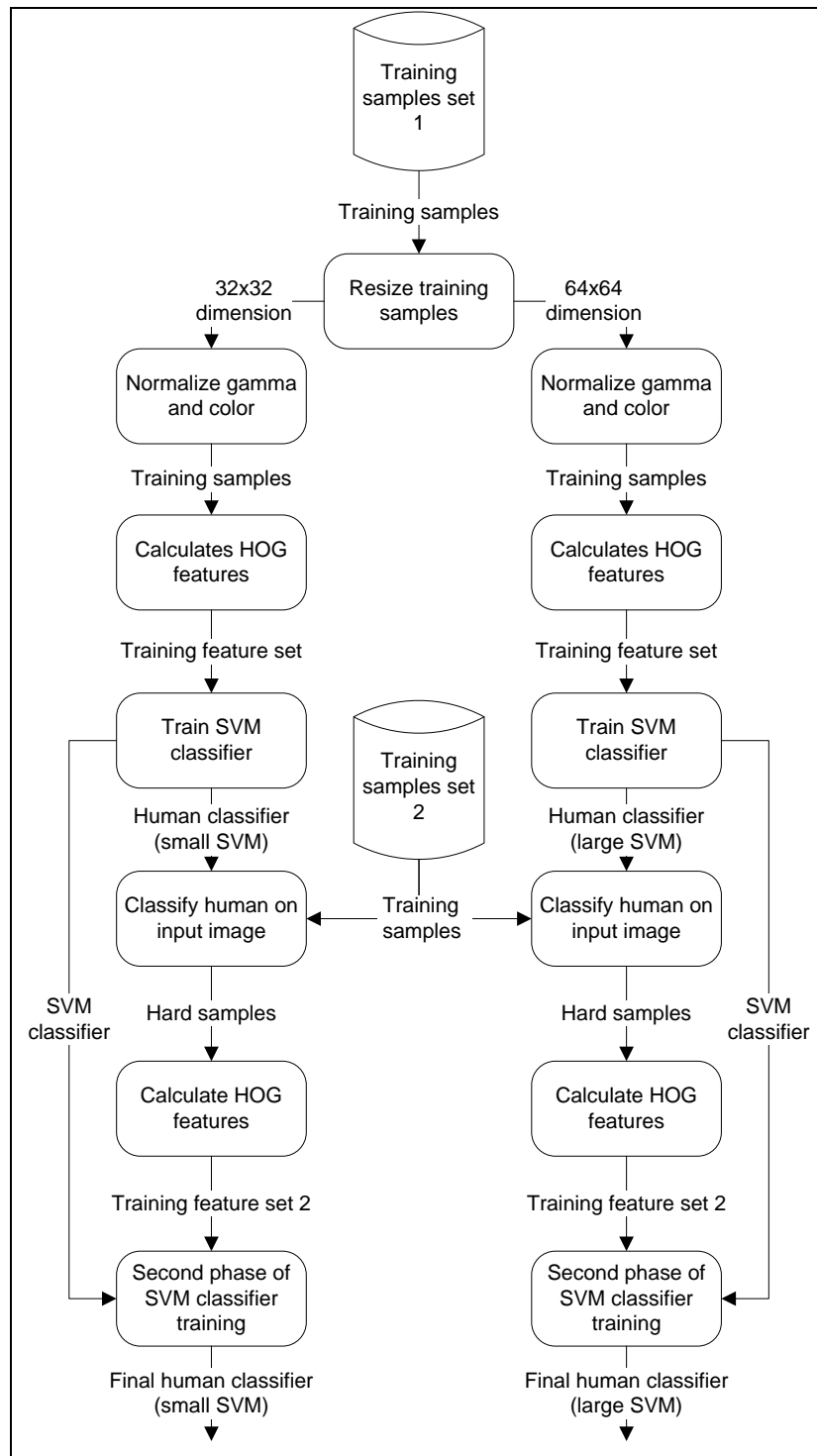


Figure 4.4 Human classifier training process

Two set of training samples are used for training purpose. The first set of training samples is used to generate the first generation of SVM classifier and the second set of training samples is for enhancement purpose. The first set of training samples includes 7,500 positive samples (image of human head-shoulder) obtained from Massachusetts Institute of Technology (MIT) Center for Biological and Computational Learning (CBCL) pedestrian dataset, INRIA pedestrian dataset, and self-captured dataset; and 10,000 negative samples (image of non-human such as tree, car, building, table, etc.) which are obtained from 1,000 random images from INRIA pedestrian dataset (10 random regions in each image). The second set of training samples are 100 images which do not contain human such as scenery, building, etc. Some of the sample sets are shown in figure 4.5 and 4.6.

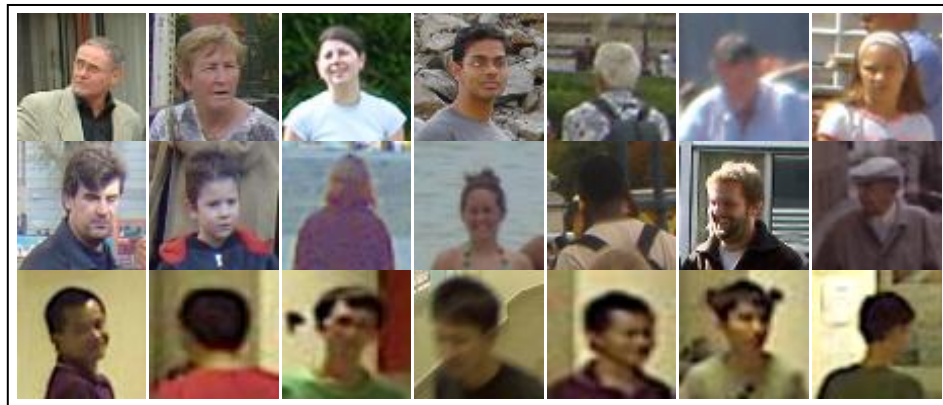


Figure 4.5 Example of positive samples

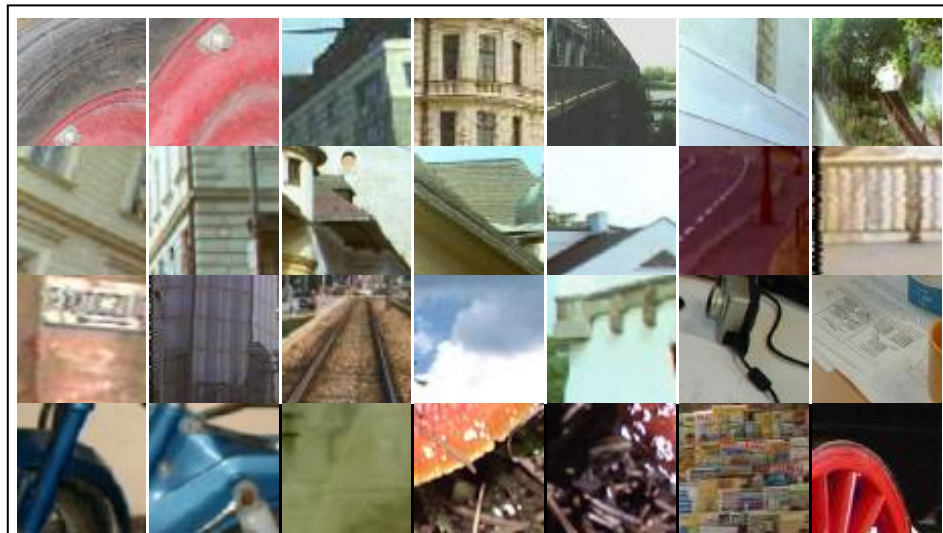


Figure 4.6 Example of negative samples

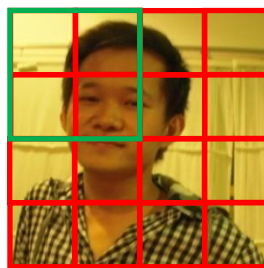


Figure 4.7 Blocks (green) and cells (red) in the detection window

Before training the SVM classifier, the training samples are being normalized for their gamma and colour to reduce the illumination and chrominance disturbance for detecting human (such as shadow, over exposure, etc). Next, the normalized training samples are converted into HOG features form. These features are collected from the blocks of a dense overlapping grid of blocks covering the detection window image. Therefore, each individual cell is shared among several blocks with different normalizations. Thus, the cell appears several times in the final features. In practice, since the head and shoulder part of a human is nearly a square as in figure 4.7, hence the detection window is set to square as well.

For small SVM, the sample images that are smaller than 48×48 dimension are resized to 32×32 dimension. Each image is represented as 3×3 blocks (where each block has 50% overlapping with its adjacent block) and each block is represented as 2×2 cells (where each cell is 8×8 pixels) where each gradient is calculated from cells with 9-bins (gradient bin range from $0 - 180$ degree where each bin has a range of 20 degree). In addition, there is no padding added when calculating the HOG feature values (since padding already added in the training sample images). Therefore, the HOG feature size for this sample image is $9 * 4 * 9 = 324$ -D feature vector. Typically each cell is shared between blocks since there are 50% overlapping of blocks. However, the cell may have different value in different blocks since each block has its own normalization level for the contrast level.

Similarly, for large SVM, the sample images that are larger or equal to 48×48 dimension are resized to 64×64 dimension. Each image is represented as 7×7 blocks with 50% overlapped and each block is represented as 2×2 cells where each cells is calculated with 9-bins. Therefore, the HOG feature size for this size of input image is $49 * 4 * 9 = 1764$ -D feature vector.

The first generation of SVM classifier is built after the preparation of HOG feature vectors for initial set of training samples. The training of SVM is done with the help of SVMLight library (Joachims, 1999). With the successful building of large and small SVM classifiers, the next step is to detect humans on the second set of training samples. Any regions classified as human will be

cropped and used as hard samples for second phase of SVM classifier training. 3,000 hard samples are collected during the stage.

The final training feature vectors size for small SVM is nearly 0.3 Gigabytes in text file format and nearly 1.6 Gigabytes for large SVM. Similarly, with the help of SVMlight library, the final SVM classifiers are built (large and small SVM).

4.5.2 Classifying of Human

There are several steps involved in detecting humans in the proposed system. The process of detecting humans is shown as figure 4.8.

With the use of sliding window (explained in next section), the system obtains local information (small region of motion blobs) which acts as the input for P4. First of all, the input image will be resized into 32×32 dimension or 64×64 dimension depending on the size of motion blobs. Every motion blob smaller than 48×48 dimension (50% bigger than 32×32 and 50% smaller than 64×64) will be resized to 32×32 dimension; every motion blob between 48×48 and 64×64 dimension will be resized to 64×64 dimension; and no resizing process if the motion blob is larger than 64×64 dimension.

Later on, the resized input image will undergo gamma normalization (power law compression) process that could reduce the colour of the image,

hence reduce the influence of illumination effects since colour texture strength of the image is typically proportional to the illumination (Human Feature Extraction in VS Image Using HOS Algorithm, 2007).

After the gamma normalization process, the gradients are computed based on the edge contour of the image. After the gradients are computed, they are collected based on the cells and blocks (as shown in figure 4.7 in sections 4.5.1).

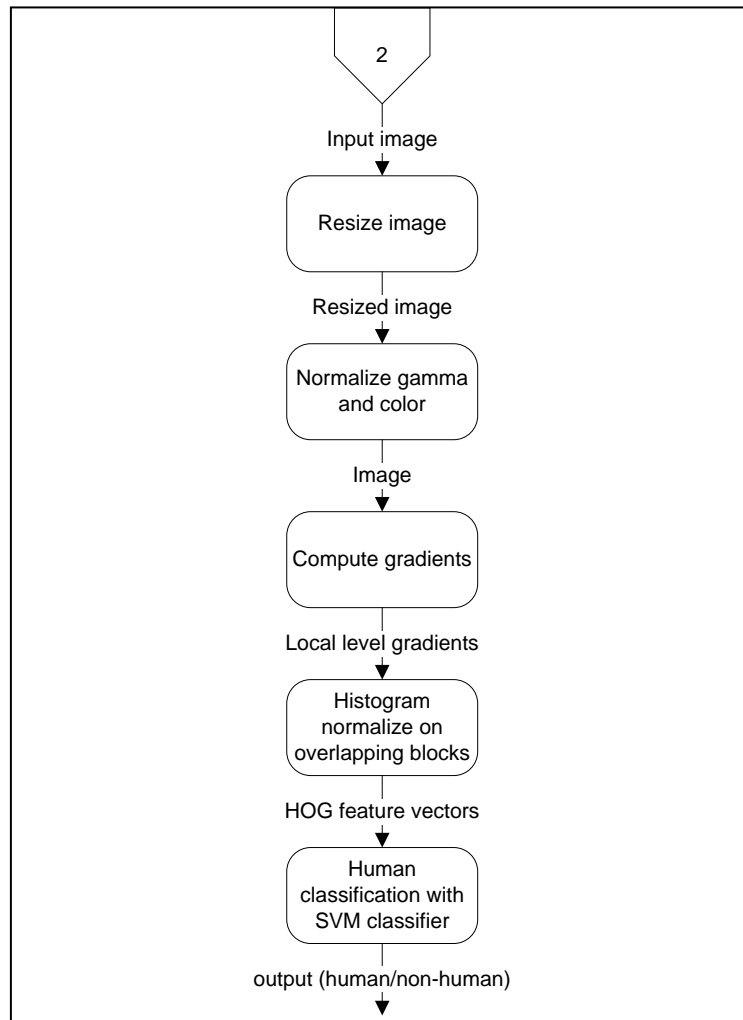


Figure 4.8 Process flow of human detection

Each detection window is represented by $n \times n$ blocks (depends on the input image's dimension) with 50% overlapped. Each block is further divided into 2×2 smaller regions called cells where each cell is represented as 8×8 pixels and finally a 9-bin HOG features is extracted from the cell. The final HOG feature (gradients collected) size is variant depending on several factors. Those factors include the detection window's dimension, bin (gradient grouping range), cell size and block size.

Next, those overlapped blocks are being normalized and finally the output (HOG feature values) is fed into the SVM classifier for human classification.

4.5.3 Sliding Window

Since human could appear in any locations within a captured scene, locating the head and shoulders could be challenging. A common way of detecting the location of head and shoulder is the use of sliding window, which is a scalable squared box used to move around, in a predefined way in the motion blob.

For P4, the sliding window is used to focus on different regions of the motion blob and calculate HOG feature values for the human classification.

For detecting "head and shoulder", the first sliding window is a square (since head and shoulder part of human is nearly square in shape). Similar to

the human ratio, the square size is obtained from a set of training images consisting of standing and walking humans. The sliding window starts from the top left corner of the image (as shown in figure 4.9) and ends at the bottom right corner of the image (as shown in figure 4.10). Whenever the sliding window reaches the bottom right corner of the image, the sliding window is scaled into a bigger size (5% size increment for each scaling process) and start again at the top left corner of the image (as shown in figure 4.11). The scaling stops when the size of the square reaches maximum size of human head and shoulder that is allowed in the scene.

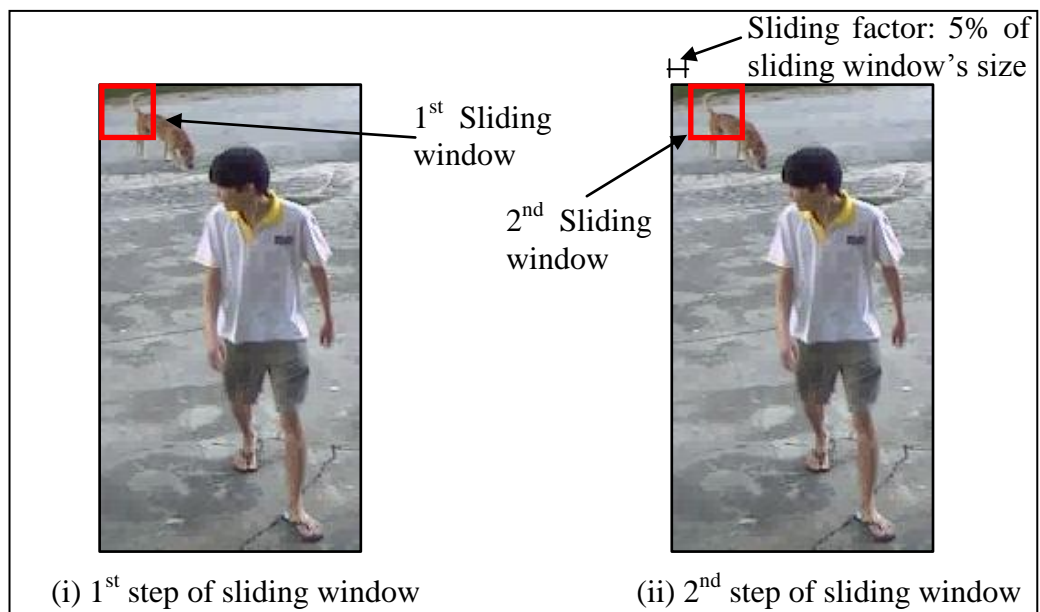


Figure 4.9 Example of sliding window's process (first and second step)

Even though the sliding window enhances the detection power of the system, the drawback is obvious, which is slowing down the overall processing speed. However, it is worth as gaining significant detection power and yet slightly reduces in speed.



Figure 4.10 n^{th} sliding window



Figure 4.11 Scaled sliding window

4.6 Human Tracking (P5)

P5 ensures that the information gathered from the scene does not create any confusion for other modules. There can be more than one human in the scene at any moment, therefore tracking each of them enables the system to correctly tag the information gathered with the right human captured in the scene. Figure 4.12 shows the steps in tracking humans.

In practice, the system only needs to track the object of interest (humans) in the scene. Therefore, the system only obtains the tracking features in the regions with the presence of such objects during the feature finding stage.

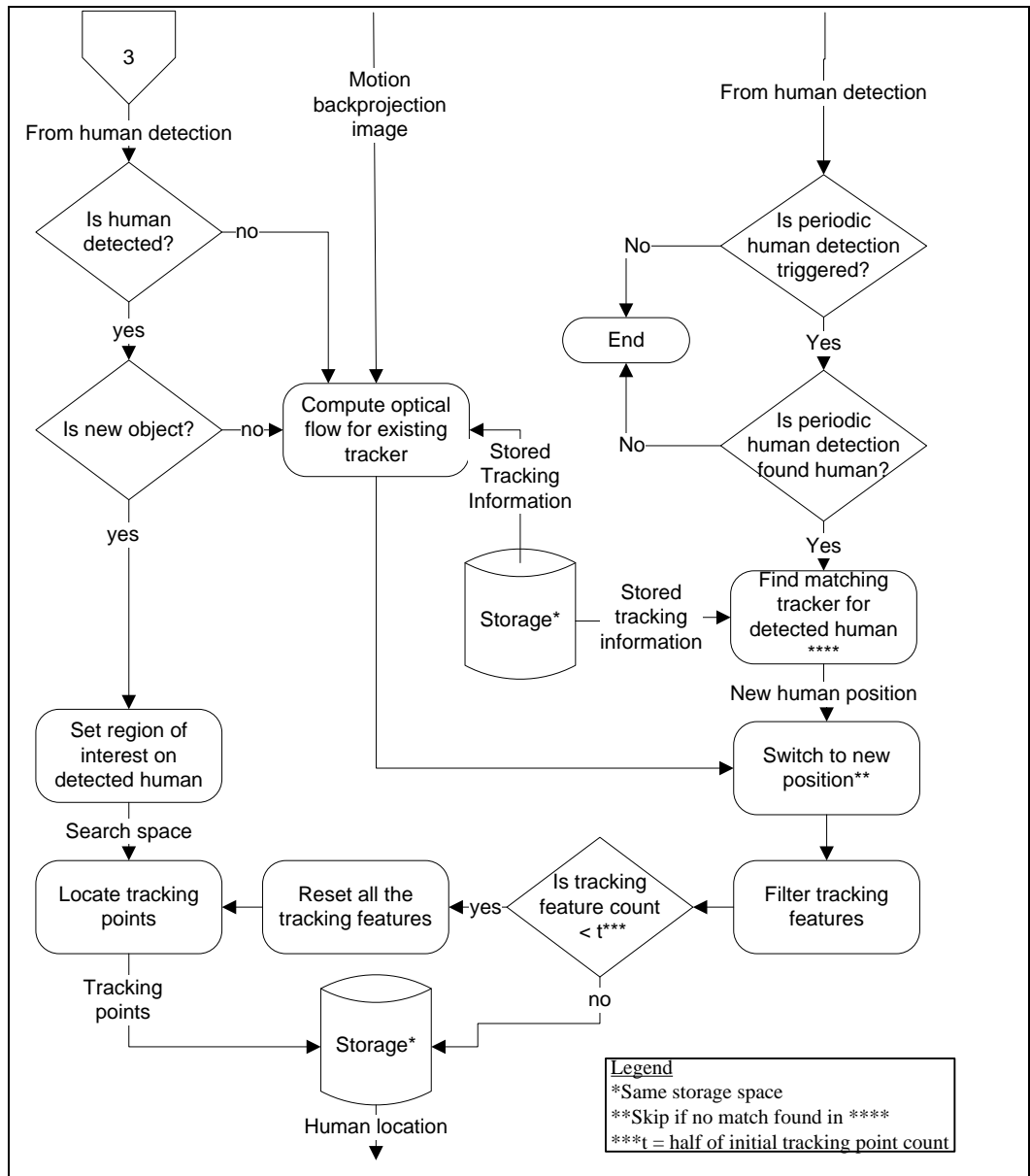


Figure 4.12 Process flow of human tracking

Initially, the system will check whether the input human is already tracked. If this is the case, then the system will calculate the new location of the human using LK optical flow algorithm.

There exists some scenario where most of the tracked points of the tracker are left on the background pixels (due to the similar foreground and background pixels) rather than following the object of interest. In this case, we compute human detection periodically (1 second interval); if there is human detected, a simple matching process will be commenced to find the matching tracker for the detected human and update the tracker information.

If the input human is a new human (detected human who just entered the scene), then the system will start to get the tracking features from the region of interest and pass this information of the tracked human to the subsequent processes.

4.6.1 Optical Flow

LK optical flow algorithm (Denman, Fookes, & Sridharan, 2009) is implemented in P5 that could help locate new location of the tracking features which can be grouped together to form the new location of the tracked human in a new input frame. The feature used in optical flow tracking is strong edge points which can be found on the edge pixels of an image. These points are highly differentiable from their surrounding pixels based on their pixel colours.

To complement with the optical flow algorithm, the motion backprojection image is used instead of a whole image frame from the camera source. This is to force the optical flow algorithm to operate only on the regions with the presence of motion. This could minimize the error which might occur during the process of P5 by reducing the tracked features search space. Example of original image and motion backprojection image is illustrated in figure 4.13 and 4.14.



Figure 4.13 Original image from video source



Figure 4.14 Motion backprojection image

Every time the system calculates the new location of the tracked points using LK optical flow algorithm, the system will create a rectangle (same size as the object in previous frame) that could bind as many tracked points as

possible. Any points that are out of bound will be ignored from the list of tracked points. This is to reduce the effect of outlier which could cause the tracking deviate from the original target. If the remaining active tracked points are less than a threshold “ t ”, then the system will reinitialize tracked points to that human to prevent the tracking process fail due to insufficient tracking information, but before that, a human detection process will reconfirm the location of the tracked human. In the system, the threshold “ t ” is half of the initial amount of tracked points found during feature finding stage. In addition, the rectangle will automatically scale itself according to the scene in which the rectangle is scaled into smaller size when it is farther from the camera; and scaled into bigger size when it is nearer to the camera.

While tracking, the system will constantly check whether the tracked location is within the motion region (if the tracked location is not within the motion region, it could be a background), and try to adjust the tracked location based on the motion. For instance, if the motion blob is slightly out from the tracked location, the tracker will adjust its tracked location by shifting it towards the motion blob by 10% of the distance between them. This is to make sure that the tracker is nearer to the moving object which could reduce the chances for the optical flow algorithm to go wrong that deviates from the original tracking object (which is moving human).

4.7 Direction of Motion Calculation (P6)

While humans are being tracked in P5, the information will be used to calculate the direction of motion of the tracked humans. For every t new locations of the tracked human calculated from P5, the motion direction of the human will also be calculated. The value t is fixed to one second for ease of calculation. The output of P6 will be used by P9.

The motion direction is being classified into 9 directions (8 directions and 1 directionless) as shown in figure 4.15. For every two consecutive locations of the human (from P5), local direction can be calculated using formula 4.5, 4.6, and 4.7, and the direction (for that particular second) can be calculated using formula 4.8.

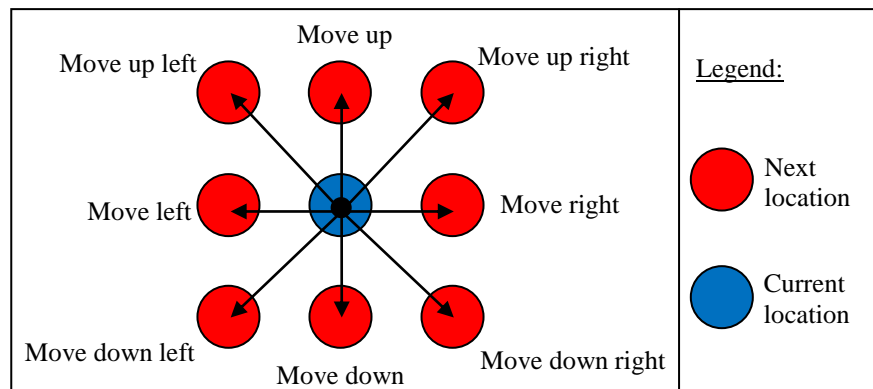


Figure 4.155 8 possible movement directions of tracked human

$$x = x_{latter} - x_{former} \quad (4.5)$$

$$y = y_{latter} - y_{former} \quad (4.6)$$

$$\text{Localdirection} = \left\{ \begin{array}{ll} \text{Non} & , \text{ if } x = 0 \ \& \ y = 0 \\ \text{Down} & , \text{ if } x = 0 \ \& \ y > 0 \\ \text{Up} & , \text{ if } x = 0 \ \& \ y < 0 \\ \text{Right} & , \text{ if } x > 0 \ \& \ y = 0 \\ \text{Down right} & , \text{ if } x > 0 \ \& \ y > 0 \\ \text{Up right} & , \text{ if } x > 0 \ \& \ y < 0 \\ \text{Left} & , \text{ if } x < 0 \ \& \ y = 0 \\ \text{Down left} & , \text{ if } x < 0 \ \& \ y > 0 \\ \text{Up left} & , \text{ if } x < 0 \ \& \ y < 0 \end{array} \right. \quad (4.7)$$

$$\text{Direction} = \max_{\text{occurrence}}(\text{localdirection}) \quad (4.8)$$

4.8 Head Localization (P7)

The input, which is a set of images containing both heads, shoulders and bodies (from P5) will be fed into this process (P7) to further identify the actual head positions. The following figure shows a sample image that is input into this process with its expected deliverable.

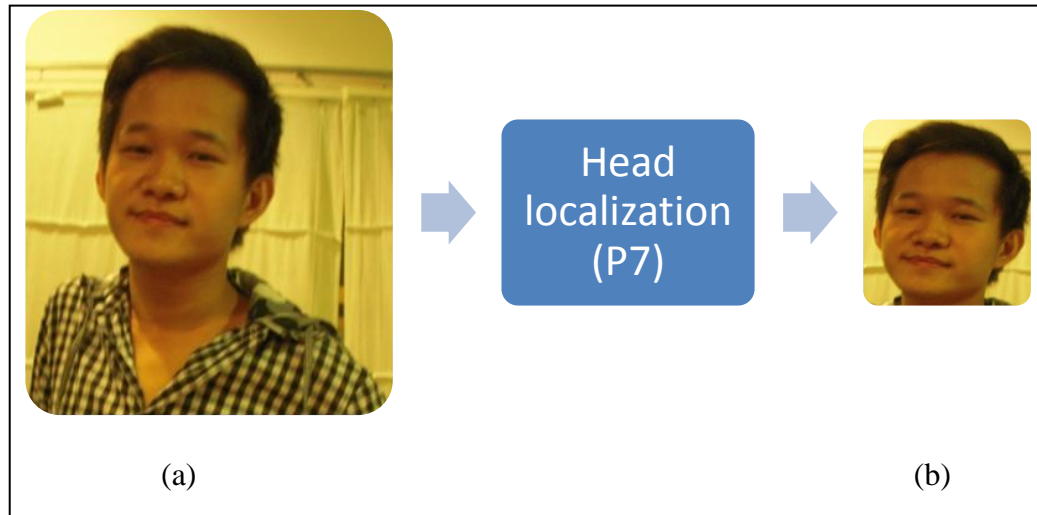


Figure 4.166 (a) Input image of P7 (head and shoulder); (b) True head region (result of P7)

In order to get the expected output as above, the following steps are followed:

- Step 1. Foreground extraction.
- Step 2. Determine shoulder line.
- Step 3. Localize head position.

Each of these steps will be explained in details in the following subsections.

4.8.1 Foreground Subtraction

From the input image (figure 4.16a), with both the coordinate values (top left corner of image) and image size, cropping can be easily done using the input from P1. The resulting image is shown in figure 4.17.



Figure 4.177 Motion blob of head shoulder image (from figure 4.16a)

4.8.2 Determine Shoulder Line

From the motion blob (shown in figure 4.17), the temporary shoulder line starts as the last row of the image. The total number of motion pixels in the last row of image, α is noted (the value α will be used for estimating the location of shoulder line). Next, the temporary shoulder line is moving upwards by one pixel and a count of all the pixels that lie on this temporary shoulder line will be obtained. This computed count will be fed into formula 4.9 to check whether the actual shoulder line is reached. This process repeats until the actual shoulder line is detected. The shoulder line is detected when the current total number of motion pixels and α satisfy the formula 4.9.



Figure 4.188 Detection of shoulder line. (a) temporary shoulder line at last row of image, (b) temporary shoulder line at second last row, (c) detected shoulder line

$$\frac{\text{total number of motion pixels that lie on the } TSL}{\alpha} \times 100\% \leq \beta\% \quad (4.9)$$

Where TSL is the temporary shoulder line and β is a threshold value (to be obtained through training set)

Once the shoulder line satisfies the formula 4.9, a verification step has to be performed to ensure that it is within a bound regions between the two threshold values (upper and lower boundary of the rectangle), which are obtained through the training set.

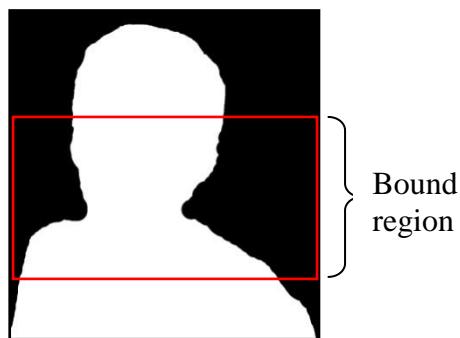


Figure 4.1919 Valid shoulder line region (red rectangle)

They could be input images where shoulder lines fail to be detected. For instance, if the input images contain human figures where the heads or shoulders are not easily noticed. If all the temporary shoulder lines fail to satisfy the formula 4.9, an ideal shoulder line will be used. The said ideal shoulder line is obtained from a set of training images. Since the motion blob is not always in good shapes (due to background subtraction, noises, etc.), the temporary shoulder line might fail to satisfy the formula 4.9. The ideal shoulder line is to be calculated based on formula 4.10.

$$\textit{ideal shoulder line} = t \times \textit{image height} \quad (4.10)$$

Where t is a threshold value (to be obtained through training set).

4.8.3 Localize Head Position

After obtaining the final shoulder line, the next step is to localize the head position. Here, locating the head position is formulated as a searching problem, and to solve this problem, a fast and reliable searching technique, hill climbing algorithm is used. In hill climbing algorithm, it usually starts at a random state, and compared to its neighbouring states for better solution. To localize the head position in the image, starting from a random is a pain due to the large search space. To reduce the search space, the starting location is set at the coordinate (a, b) , where a is half of width of the image and b is the y -coordinate of the top motion pixel (a, y) (as shown in figure 4.20). The neighbours are the pixels next to current state and a better state is that of

higher motion pixel (higher in terms of smaller y -coordinate of the image) across the x -axis.

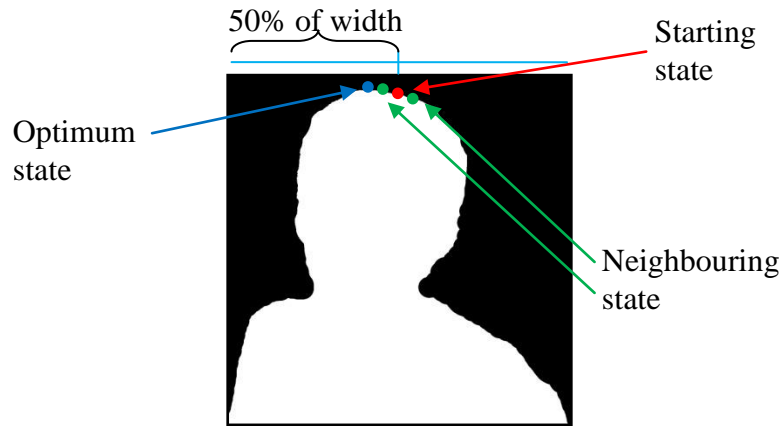


Figure 4.200 Starting, optimum and neighbour state of head location searching

This search is expected to find the top most motion pixels (as shown blue dot in figure 4.20). By finding this pixel, the head position can be estimated by using the distance between the top most pixel and the shoulder line (difference between y -coordinate) as shown in figure 4.21.

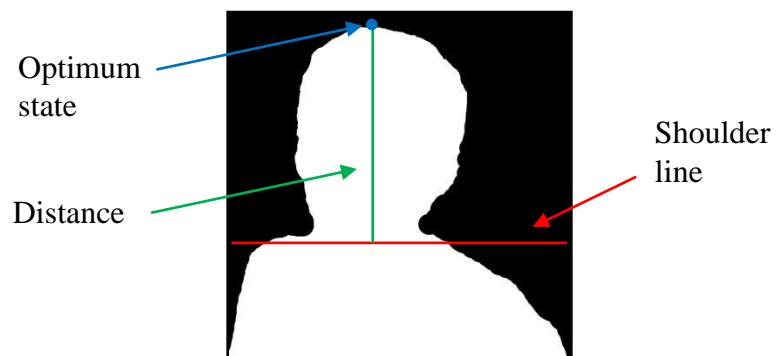


Figure 4.211 Distance between top motion pixel and shoulder line

By obtaining the distance between the top motion pixel and shoulder line, the head position can be found. The distance is used as the size of the head. Figure 4.22 shows the detected head with the distance as height and centered at (c, d) , where c is the x -coordinate of the top most motion pixel and d is the y -coordinate of the top most motion pixel plus half of the length (center of estimated head).

At last, the correct position of the head has been determined and this will serve as an input to the next process (P8).

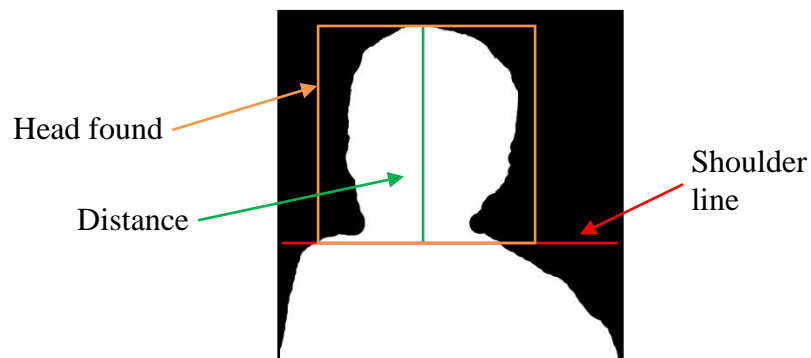


Figure 4.222 Localization of head location

4.9 Head Pose Estimation (P8)

Once the head location is found (from P7), it is then passed to P8 as input for head pose estimation. This process is to decide on the direction where the head is facing. The classifier used in P8 is NRBVS engine (refer to section 4.9.4.1) and the feature used is percentage of skin and dark colour within each divided regions (refer to section 4.9.3 for feature extraction).

The head pose is categorized into 8 different categories of postures as shown in figure 4.23. The estimation of head pose is based on the novel NRBVS approach which makes use of the colour information (dark and skin colour) of the head image. Here, it is assumed that all human detected have black or dark hair colour and Asian skin colour (example as in figure 4.24). Figure 4.24 shows the process flow of P8.

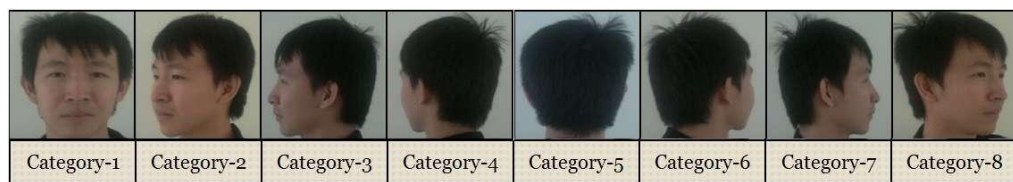


Figure 4.233 8categories of head pose

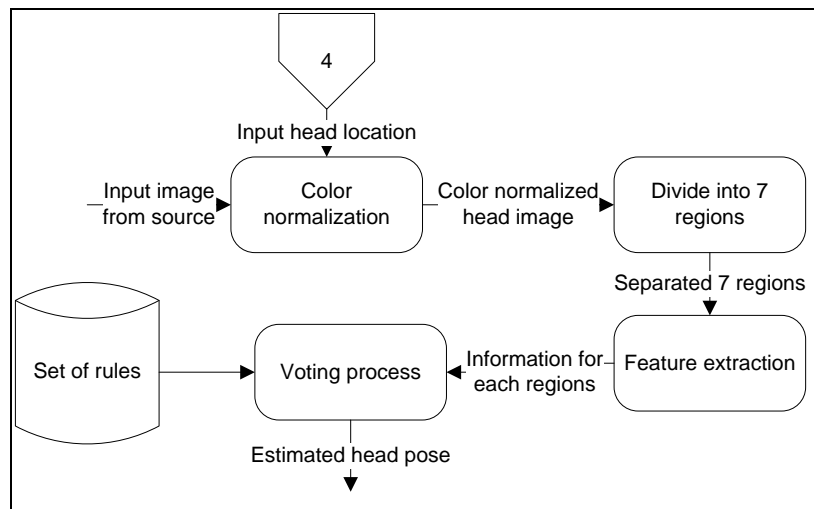


Figure 4.244 P8 process flow

The algorithm used for estimating the head pose in P8 is very sensitive to colour change. Some of the factors that affect colour change are under exposure, over exposure, etc. on the image itself. Therefore, a colour normalization technique (refer to section 4.9.1) is used to balance out the colour of the image before the estimation process.

4.9.1 Colour Normalization

When implementing colour normalization algorithm, certain assumption is made about the nature of colour of an image. The assumption is the gray world assumption. Gray world assumption states that, given an image with sufficient amount of colour variations, the average value of red, green, and blue components of the image should average out to a common gray value. This assumption is generally valid since in any real world scene, it would be a lot of different colour variations which are random and independent. Therefore, given a large enough amount of samples, the average of colour of the image should tend to converge to the mean value, which is gray (Buchsbaum, 1980).

This proposed system makes use of the gray world assumption for re-acquiring the original colour for the input image frame by recalculating the new colour (red, green, and blue values) for each pixel using the formula 4.11, 4.12, and 4.13.

$$ave(r, g, b) = \frac{\sum_{i=0}^{width * height} (r, g, b)}{width * height} \quad (4.11)$$

$$overall\ gray = \frac{ave(r) + ave(g) + ave(b)}{3} \quad (4.12)$$

$$scale\ factor(r, g, b) = \frac{overall\ gray}{ave(r, g, b)} \quad (4.13)$$

where r, g, b are the red, green, and blue colour value of a particular pixel and $width$ & $height$ is the width and height of the image.

Initially, the average of red, green, and blue colour values is calculated and their average value is used to determine the overall gray value for the image. Each colour value is then scaled according to the amount of its deviation from the overall gray value. The scale factor can be determined simply by dividing the overall gray value of the image by the average value of the corresponding colour component. Notice that there are two inputs for colour balancing process; one is input image from source (whole scene) and input head image (head image of detected human). Image from source is used to obtain the scale factor (because the algorithm needs to find the average colour from the whole scene) and the scaling is only applied to the input head image. In this way, the system can save the processing time by focusing on the head image (head of the human, to be used for head pose estimation) instead of the image from the source. The demonstration of the colour normalization technique using gray world assumption is illustrated in figure 4.25.










Figure 4.255 Colour balancing using gray world assumptions (a) before (b) after

Table 4.1 below shows the skin detection result (using skin detector discussed in section 4.9.2 *Region Division*) for both images before and after applying colour balancing technique.

From table 4.1, notice that a lot of background pixels were identified as skin pixels. The skin colour detector works better with images that applied the colour balancing technique. This is important as P8 is depending on the result from skin colour detector.









Table 4.1 Comparison of skin colour detection result using original and colour balanced image

Original Image (O)	Colour balanced image (M)	Skin colour detection result	
		Using (O) as input	Using (M) as input
			
			

To be continued...

...continued

Table 4.1 Comparison of skin colour detection result using original and colour balanced image

Original Image (O)	Colour balanced image (M)	Skin colour detection result	
		Using (O) as input	Using (M) as input
			
			

To be continued...

...continued

Table 4.1 Comparison of skin colour detection result using original and colour balanced image

Original Image (O)	Colour balanced image (M)	Skin colour detection result	
		Using (O) as input	Using (M) as input
			
			

4.9.2 Region Division

With this normalized image, the input head image is divided into 7 regions and the region numbering is as shown in figure 4.26. The image is divided in this way is due to the fact that the most important information is normally located at the center i.e. face location. Asides from the center of the image, left and right parts (numbered as 2, 3, ..., 7) of the image are also useful in which they are suitable for estimating the side views of the head.

The true head region is needed for the NRBVS engine due to the use of colour information as a feature. Non-true head region might result in false estimation of head pose. Therefore, if the head region is not found, the system will start to predict the head pose (based on the previous and the following estimated head poses) instead of proceeding to head pose estimation process.

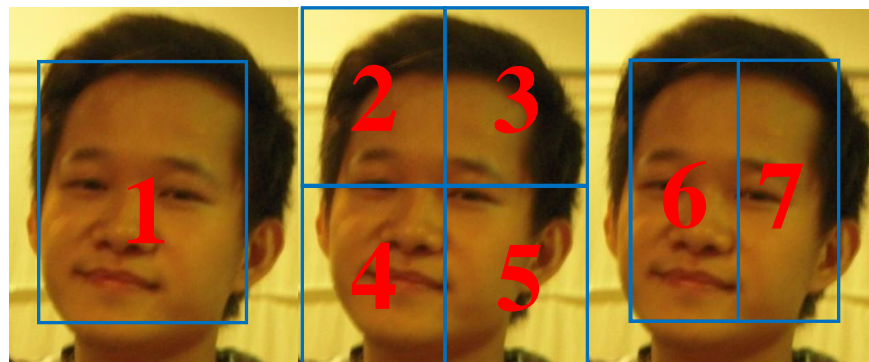


Figure 4.266 7 regions of true head region and region numbering

4.9.3 Feature Extraction

The feature used in P8 is the percentage of skin and dark colour pixels within each region (defined in section 4.9.2) and it is extracted based on the motion pixels (white colour pixels) as shown in figure 4.17. The skin or dark colour pixels will be included as feature if and only if it is a motion pixel.

The percentage of skin and dark colour within each region can be calculated using formula 4.14 and 4.15. Then, all the percentage calculated is grouped together to form the feature which represents the input image.

$$\text{percentage of skin color in } A = \frac{\text{no.of skin pixels in } A}{\text{no.of motion pixels in } A} \quad (4.14)$$

$$\text{percentage of dark color in } A = \frac{\text{no.of dark pixels in } A}{\text{no.of motion pixels in } A} \quad (4.15)$$

where A is the 7 regions defined.

This proposed system adopts the skin detection algorithm by Maggie Mae's Site (2009) which makes use of the RGB filter on RGB colour space. Three simple restrictions are included to identify skin pixels as shown in formula 4.16, 4.17, and 4.18.

$$(r, g, b) > (95, 40, 20) \quad (4.16)$$

$$\max(r, g, b) - \min(r, g, b) > 15 \quad (4.17)$$

$$|r - g| > 5 \ \& \ r > g \ \& \ r > b \quad (4.18)$$

where r , g , and b are the red, green, and blue values of particular pixels in RGB colour space respectively.

The system will check all the pixels within a given input image whether it satisfies the three formulas (formula 4.16, 4.17, and 4.18); if it does satisfy, that pixel will be classified as skin pixel.

If the pixel is not classified as a skin pixel, the system will check whether the brightness value of the pixel exceeds 10% of full brightness (that is 26 out of 256). If the pixel's brightness value is less than 26, then it is classified as dark colour pixel.

4.9.4 Voting Process

With the input images from previous process, i.e. Feature extraction, this voting process will determine the best estimated head pose category from the 8 categories of head pose defined earlier. As indicated in the diagram below, the voting process is governed by a set of rules.

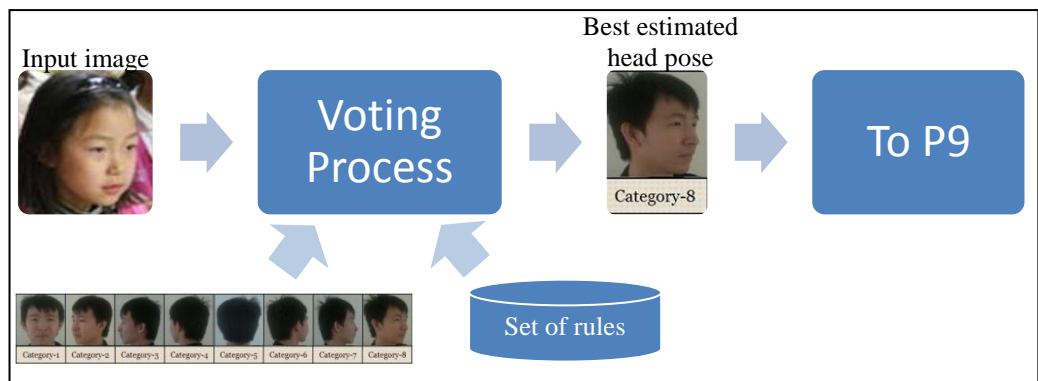


Figure 4.277 Input and output of voting process

The following aspects will detail how the voting process works:

- Nested rule-based voting system (NRBVS) engine
- Grouping of head pose
- Construction of rules
- Voting mechanism

4.9.4.1 Nested Rule-based Voting System (NRBVS) Engine

NRBVS engine is used to classify an input image's head pose based on the pixel's colour information. Initially, the input image is divided into 7 regions (refer to section 4.9.2, figure 4.26) and the percentage of the skin and dark colour pixels within each region are calculated (refer to section 4.9.3 for feature extraction). Next, the voting of head pose based on colour information is proceeded according to a set of rules that had been defined in the training session. The predefined rules govern the choice of vote. Finally, the head pose with highest vote is chosen as the head pose of the given input image.

For this NRBVS engine to yield good result, a proper training is necessary. This training is meant for constructing a set of rules (mentioned earlier in previous paragraph) used for the voting purposes. To construct these rules, 200 head images (self-captured, 25 images for each head pose, 8 head poses in total) are used. A total of 32 rules are produced from the training (16 rules for each voting session, refer to section 4.9.4.3 for construction of rules).

The whole voting process is divided into 2 sessions where the first session of voting is to classify the input image into one of the four main groups of head pose (refer to section 4.9.4.2) and the second session is to classify the input image as one of the head pose in that specific group of head pose.

4.9.4.2 Grouping of Head Pose

As mentioned in the previous section, all the head poses (8 of them) are divided into four different groups based on the similarity (in terms of skin and dark colour distribution) of the head poses. The purpose of this division is to narrow down and reduce the classification complexity (due to the similarity of head pose within a group and the significant differences between groups) and hence for a better and closer estimation of head pose for the input image. The groups of head poses are stated in table 4.2, which are constructed based on graph analysis.

Table 4.2 Grouping of head poses

Group	Head poses
A	Category 1, 2, & 8
B	Category 2, 3, & 4
C	Category 4, 5, & 6
D	Category 6, 7, & 8

There are 3 head poses in each group where some of them appear in 2 different groups and some of them only appeared within the same group. The head poses in each group share much similarity such as distribution of skin and dark colour. For instance, group C consists of head poses which the center

region of the head image consists of more than 80% of dark colour. The following figures are some graphs that show the information (obtained from average of training samples) on how the table 4.2 can be derived.

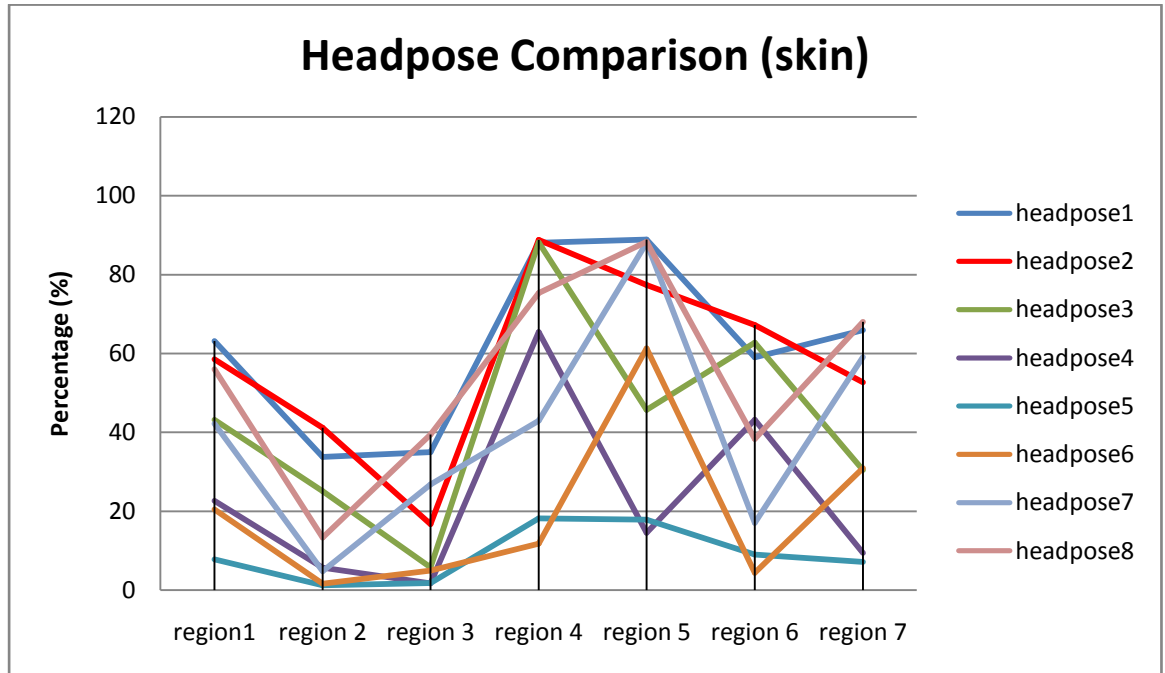


Figure 4.288 Comparison of 8 categories of head poses (skin colour)

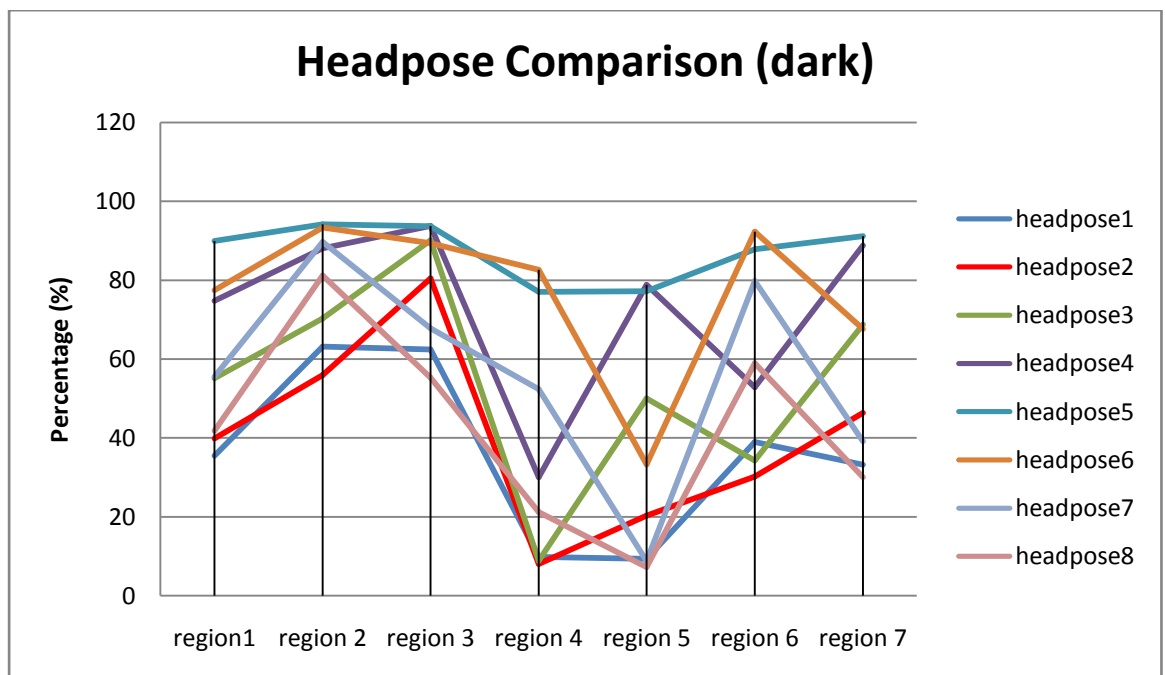


Figure 4.2929 Comparison of 8 categories of head poses (dark colour)

The main reason why there are only 4 groups of head pose is due to the uniqueness of some of the head poses. From figure 4.28 and 4.29, notice that some head poses are quite similar to another head pose (e.g. head pose category 1 and 2) which are supposed to be in the same group; and some of the head poses are different from other head pose (e.g. head pose category 1 and 5) which are suitable to be the base of a group. Based on observation, the most suitable or the head poses with the significant differences are selected as the base of the group. The selected head poses are category 1, 3, 5, and 7. To illustrate in a clearer manner, the 4 selected head poses (category 1, 3, 5, and 7) are compared in the figure 4.30 and 4.31.

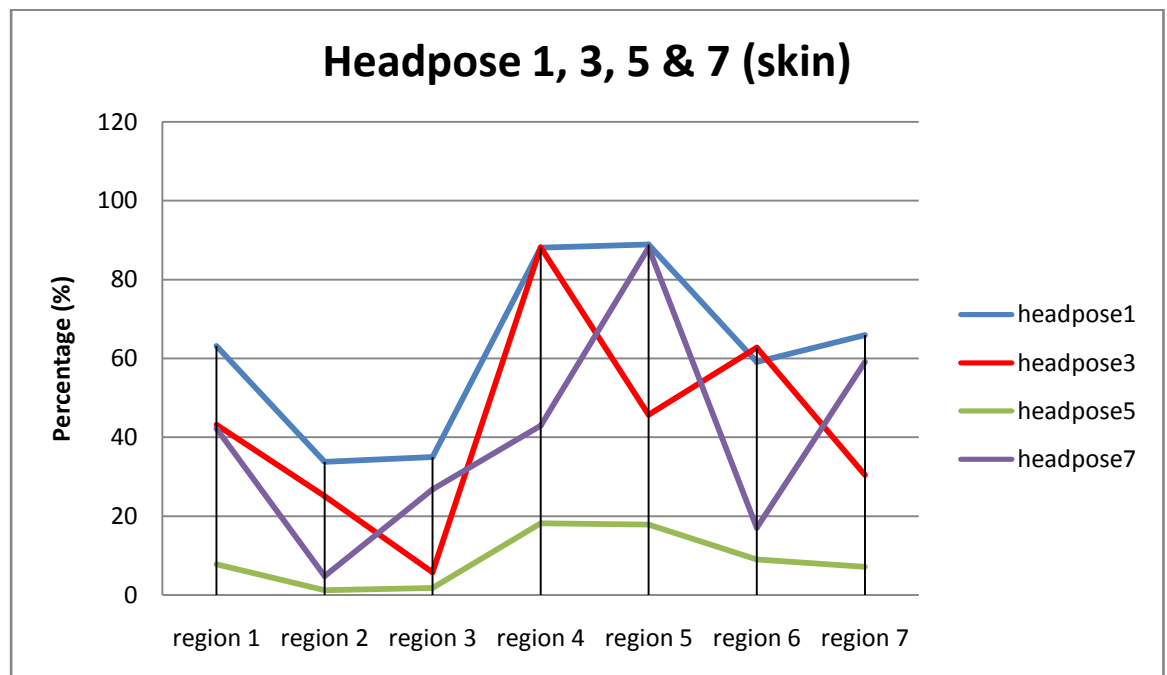


Figure 4.300 Comparison of head poses with categories 1, 3, 5, and 7 (skin colour)

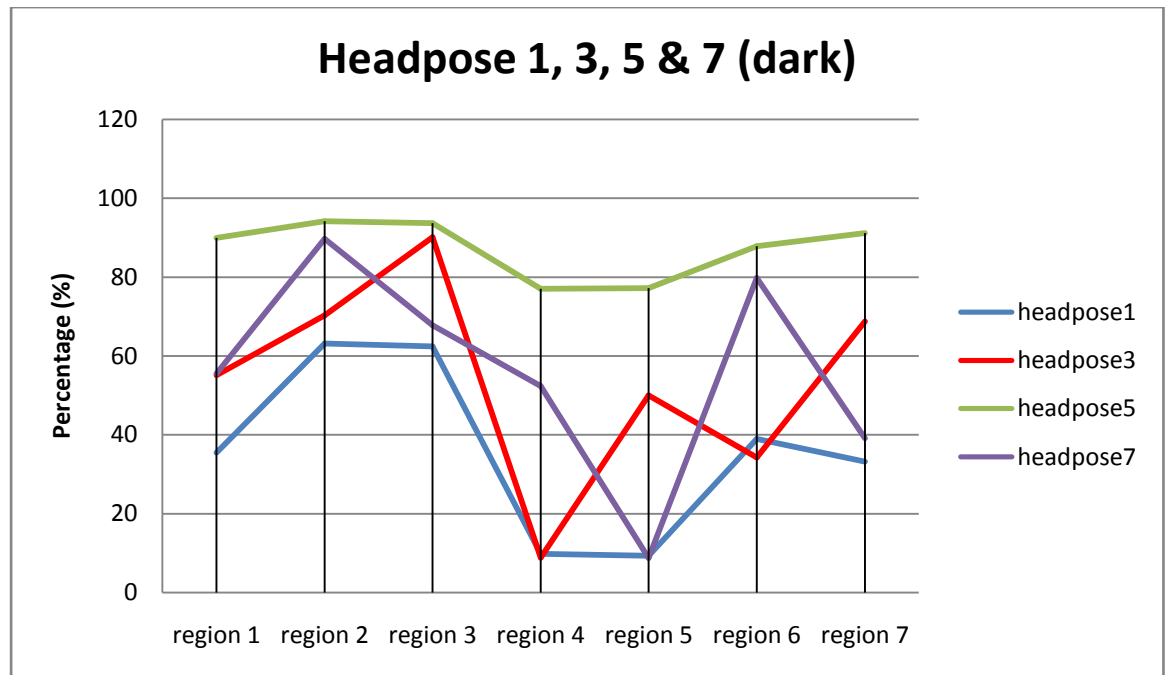


Figure 4.311 Comparison of head poses with categories 1, 3, 5, and 7 (dark colour)

In figure 4.30 and 4.31, it is clear that there are some significant differences among one another (e.g. for head pose category 5, all regions consist more than 70% of dark colour where others are not). This is why and how the groups are determined.

Note that the adjacent head poses are actually interrelated. Part of the characteristics of a head pose can be found on its neighbouring head pose (e.g. characteristics of head pose category 2 can be found on head pose category 1 and 3). For instance, based on the figure 4.28 and 4.29, the graph patterns of the head pose category 1 and 2 are similar or with subtle variation, but overall patterns of the information are similar. Therefore, head pose category 1 and 2 should be in the same group. Same goes to head pose 2 and 3, etc. By comparing non-base head pose (head pose category 2, 4, 6, and 8) to its

neighbouring base head pose (head pose category 1, 3, 5, and 7), the groupings of the head pose are done. The following figures show the graph patterns of each group of head pose categories.

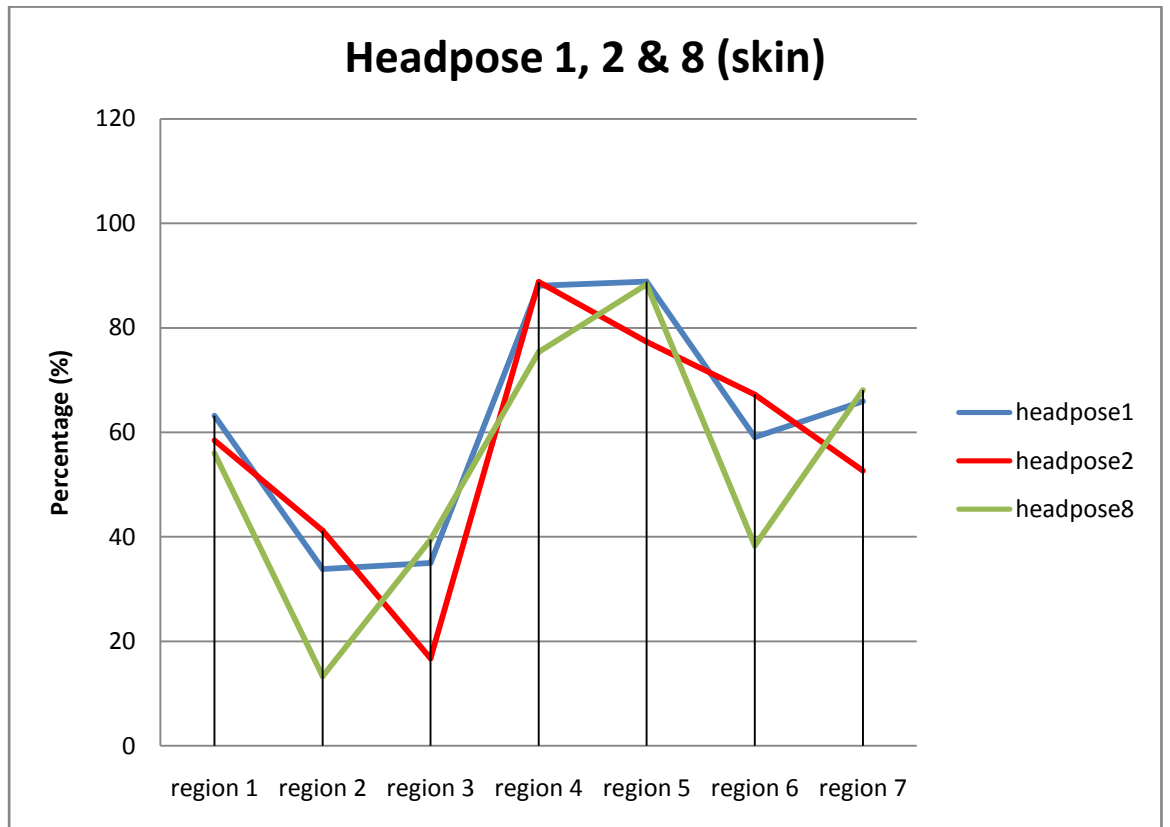


Figure 4.322 Graph patterns of head pose group A (skin colour)

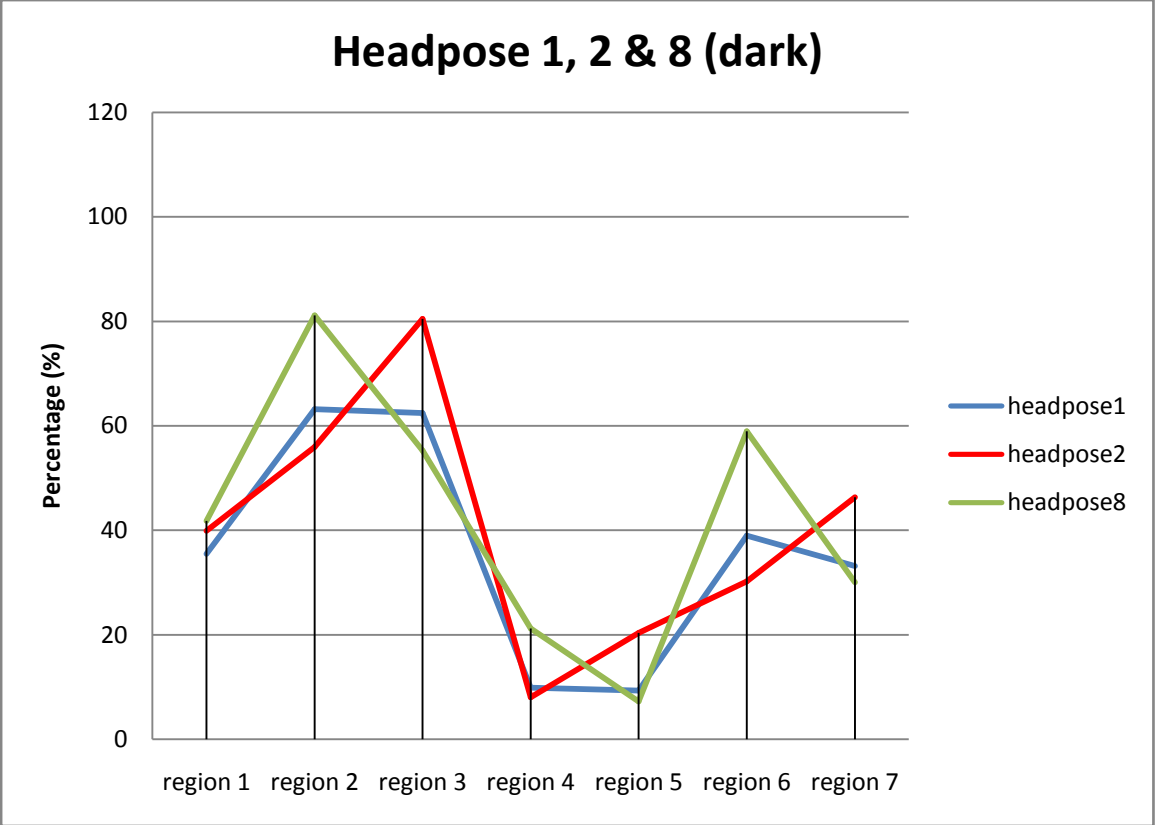


Figure 4.333 Graph patterns of head pose group A (dark colour)

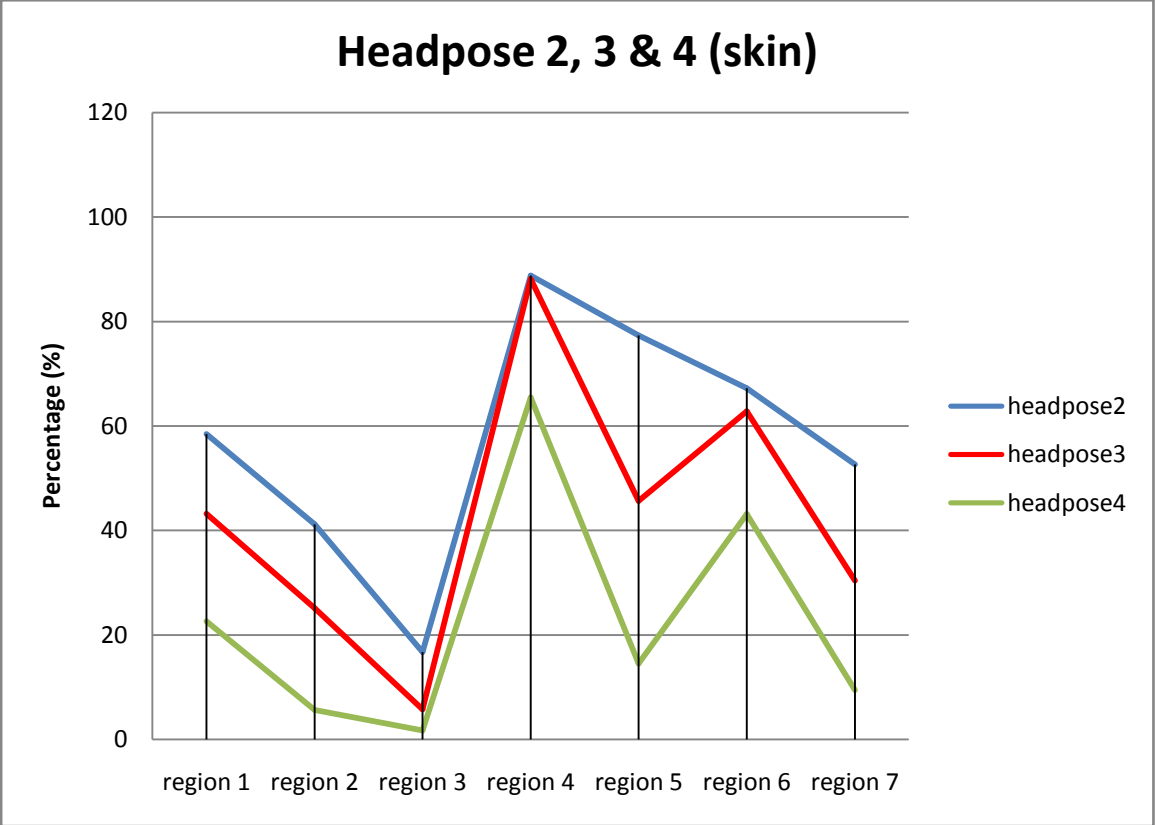


Figure 4.344 Graph patterns of head pose group B (skin colour)

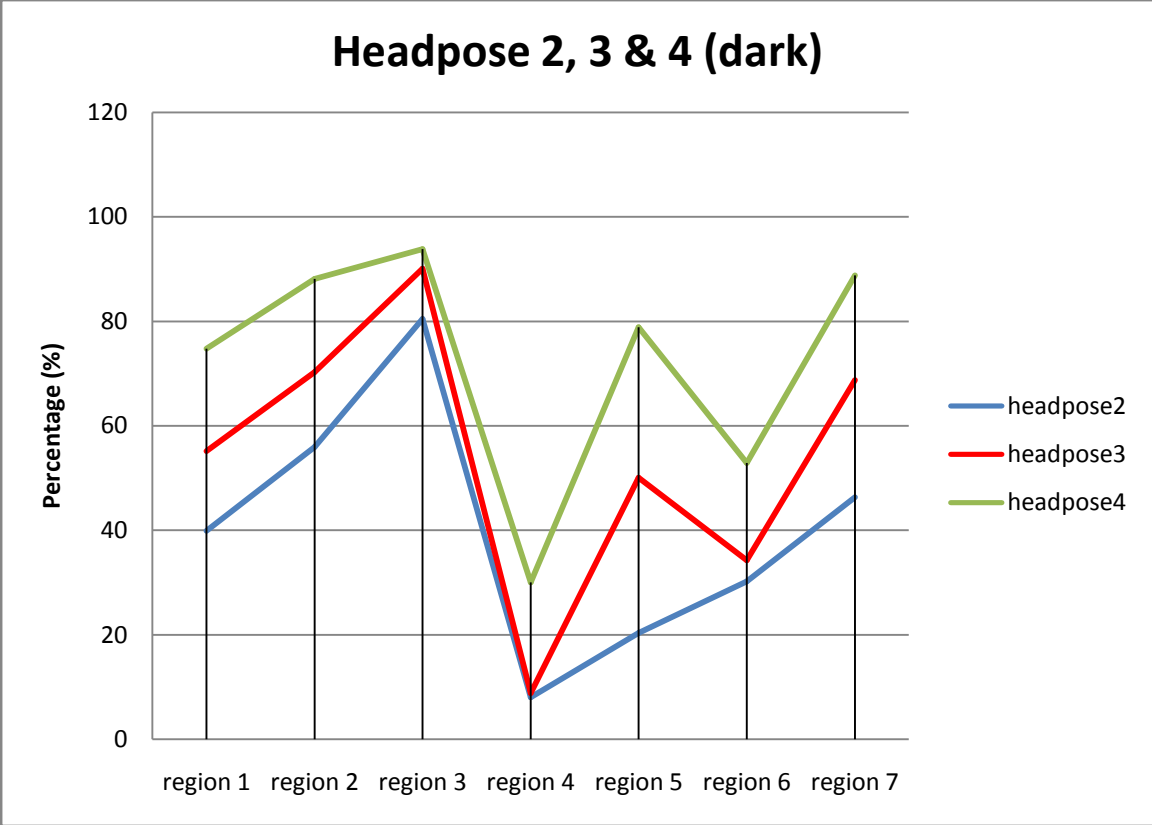


Figure 4.355 Graph patterns of head pose group B (dark colour)

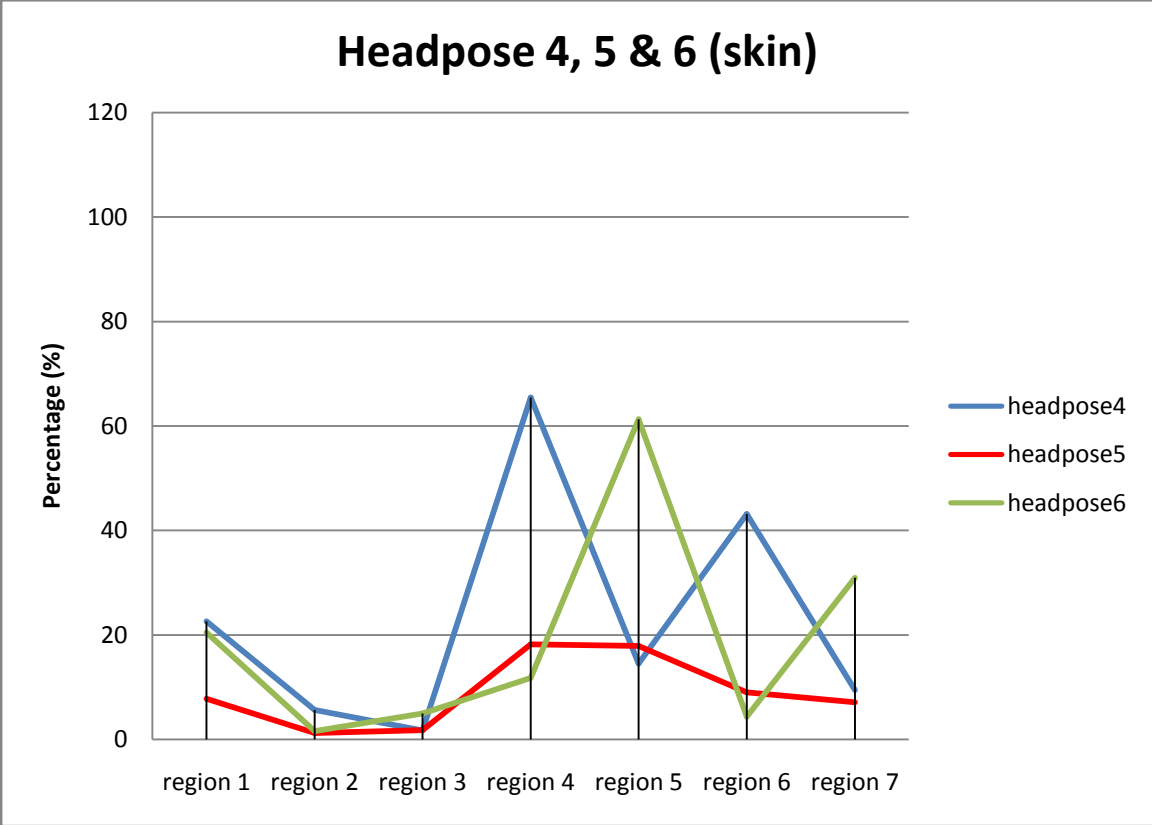


Figure 4.366 Graph patterns of head pose group C (skin colour)

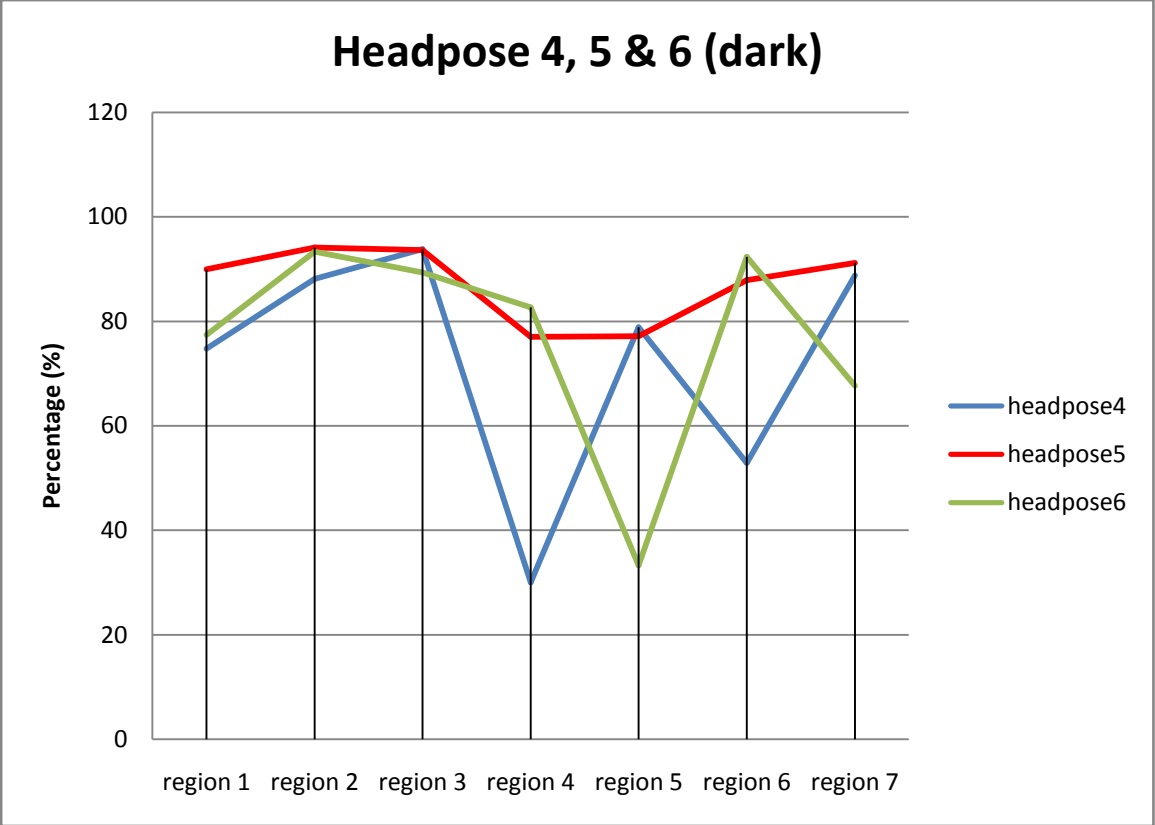


Figure 4.377 Graph patterns of head pose group C (dark colour)

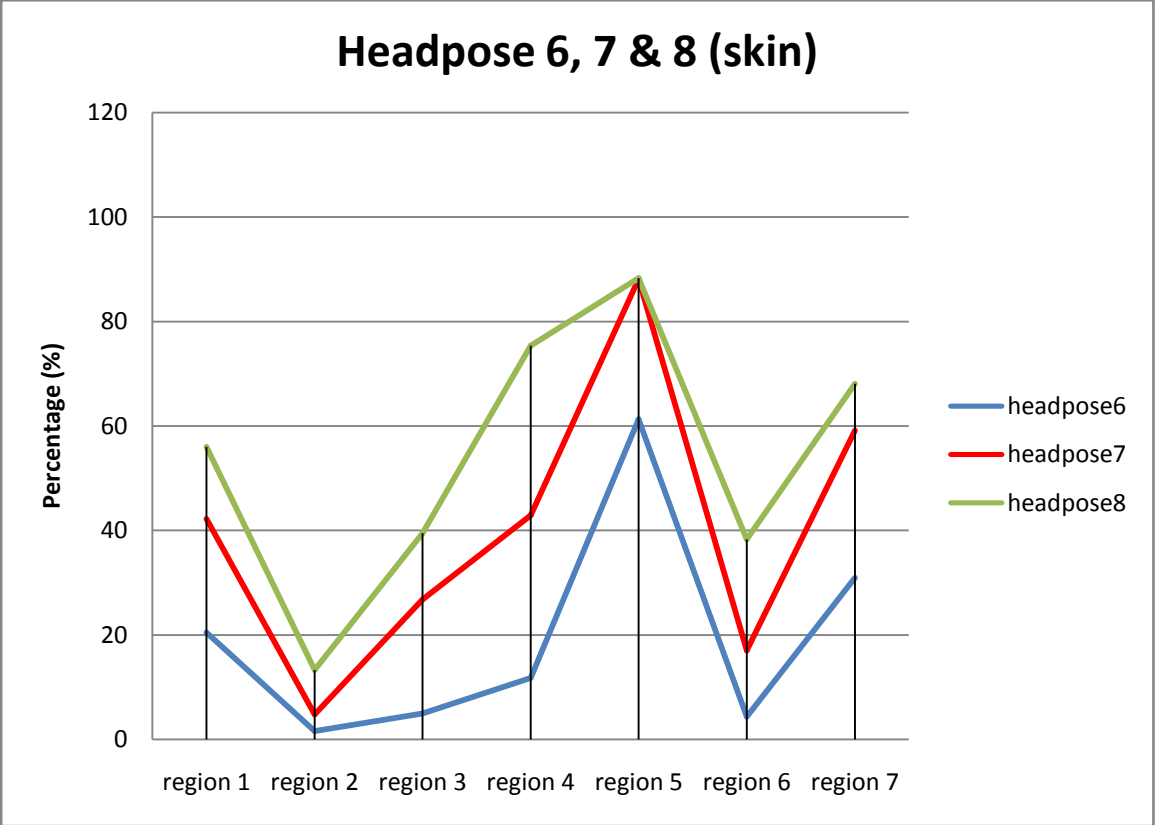


Figure 4.388 Graph patterns of head pose group D (skin colour)

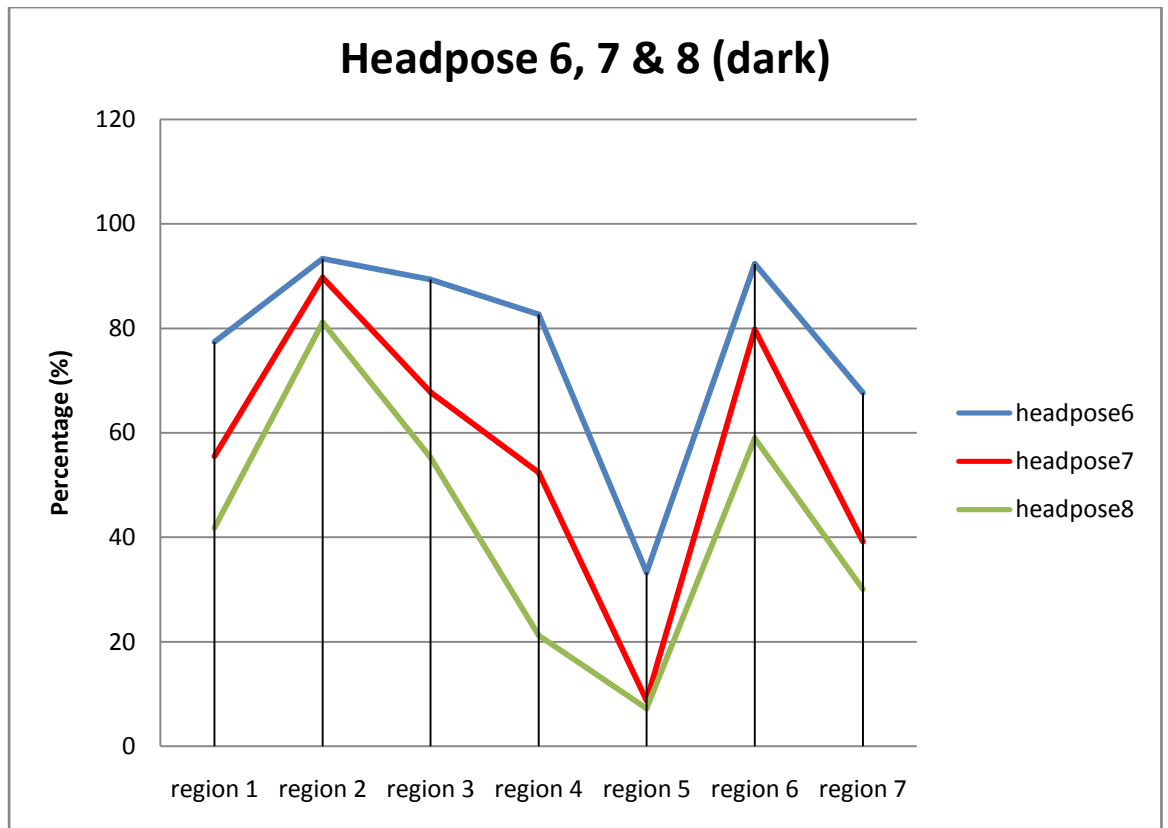


Figure 4.3939 Graph patterns of head pose group D (dark colour)

4.9.4.3 Construction of Rules

Rules are the main core of this NRBVS engine that represents the absolute statements for the whole voting process. The idea of this voting system with the use of rules is actually some kind of inference in artificial intelligent in which the rules are said the facts or statements that are always correct and may not be disobeyed. This kind of inference is normally used in artificial intelligent system for decision making (Engelmore & Feigenbawn, 1993). In this research work, the rules are viewed as absolute instructions for certain actions to be done based on the given information.

There are a total of 32 rules constructed in this research work, 16 rules for each session of voting. The objectives of rules in voting session 1 and 2 are different. In voting session 1, the rules are meant for determining the main group of head pose (where the input image belongs to) based on the given input image. The expected output from the voting session 1 is the group of head pose in which the given input image belongs to. Next, the voting session 2 will further decide which head pose category the input image belongs to. In this section, the construction of rules in each session will be shown.

	Group	Head poses
Determined through voting session 1	A	Category 1, 2, & 8
	B	Category 2, 3, & 4
	C	Category 4, 5, & 6
	D	Category 6, 7, & 8

Determined through voting session 2

Figure 4.400 possible outputs from each voting session

Rules in voting session 1

The rules in voting session 1 are constructed based on the similarities between each head pose within the same group and the differences between groups of head pose.

From figures 4.32 to 4.39, observe that the pattern of the curve in each graph is similar since they are in the same group. From those observed characteristics, the outcome of this graph analysis can be summarized in table 4.3 and 4.4.

Table 4.3 Summary of head pose group characteristic (skin)

Region	Percentage of skin colour (%)			
	Group A	Group B	Group C	Group D
1	≥ 40	$\geq 10 \text{ \& } \leq 70$	≤ 30	$\geq 10 \text{ \& } \leq 70$
2	$\geq 10 \text{ \& } \leq 50$	≤ 50	≤ 15	≤ 30
3	$\geq 10 \text{ \& } \leq 50$	≤ 30	≤ 15	≤ 50
4	≥ 60	≥ 50	$\geq 0 \text{ \& } \leq 70$	-
5	≥ 60	-	$\geq 0 \text{ \& } \leq 70$	≥ 50
6	$\geq 30 \text{ \& } \leq 80$	≥ 30	$\geq 0 \text{ \& } \leq 50$	≤ 50
7	$\geq 30 \text{ \& } \leq 80$	≤ 70	$\geq 0 \text{ \& } \leq 50$	$\geq 20 \text{ \& } \leq 80$
	Comparison between region			
2 & 3	$2 \approx 3^{**}$	$2 > 3$	$2 \approx 3^*$	$2 \leq 3$
4 & 5	$4 \approx 5^*$	$4 > 5$	$4 \approx 5^{**}$	$4 < 5$
6 & 7	-	$6 > 7$	-	$6 < 7$

Remark:

*15% threshold

**special case, only valid for specific head pose

Table 4.4 Summary of head pose group characteristic (dark)

Region	Percentage of dark colour (%)			
	Group A	Group B	Group C	Group D
1	$\geq 25 \text{ \& } \leq 50$	$\geq 30 \text{ \& } \leq 90$	≥ 70	$\geq 30 \text{ \& } \leq 90$
2	$\geq 40 \text{ \& } \leq 90$	≥ 40	≥ 80	≥ 70
3	$\geq 40 \text{ \& } \leq 90$	≥ 60	≥ 80	≥ 30
4	≤ 30	≤ 40	$\geq 20 \text{ \& } \leq 90$	-
5	≤ 30	-	$\geq 20 \text{ \& } \leq 90$	≤ 50
6	$\geq 20 \text{ \& } \leq 70$	$\geq 20 \text{ \& } \leq 70$	$\geq 40 \text{ \& } \leq 100$	≥ 40
7	$\geq 20 \text{ \& } \leq 60$	> 30	$\geq 40 \text{ \& } \leq 100$	$\geq 20 \text{ \& } \leq 80$
	Comparison between region			
2 & 3	$2 \approx 3^{**}$	$2 < 3$	$2 \approx 3^*$	$2 > 3$
4 & 5	$4 \approx 5^*$	$4 < 5$	$4 \approx 5^{**}$	$4 > 5$
6 & 7	-	$6 < 7$	-	$6 > 7$

Remark:

*15% threshold

**special case, only valid for specific head pose

Refer to the remark, some of the characteristics are not shared among head pose within a group but belong to a specific head pose only. However, this information is essential for the voting, therefore it is included as part of the characteristics that is shared (not in real) by head poses within the same

group. With the characteristics obtained, rules can be generated for classifying groups. One of the rules generated is illustrated in figure 4.41.

The blue rectangles in figure 4.41 contain instructions or statements that must be followed for decision making; whereas for the red rectangles, they contain decisions made based on the instructions (in blue rectangle) and given information (input head image). The rule structure is from top to bottom. It starts at the first top blue rectangle and ends with a red rectangle.

Refer to the percentage of skin colour of region 1 in table 4.3 and first top blue rectangle in figure 4.41, the only group of head pose that has lower than 10% skin colour is group C. Therefore, the first decision is to vote for group C if the input image satisfies the first statement. Then, check again for next possible statement. From table 4.3 again, it is observable that groups B, C, and D will have the percentage from skin colour between 10 – 30% and this has become the second statement for this rule. In a similar way, all the statements and decisions (for both skin colour and dark colour information) can be derived and they form a complete rule. The rule structure is similar to a tree structure where the root will only have zero or two branches constructed using some conditional if-else choices (nested if-else) in the programming language. This is how the name nested rule-based (from NRBVS engine) is derived.

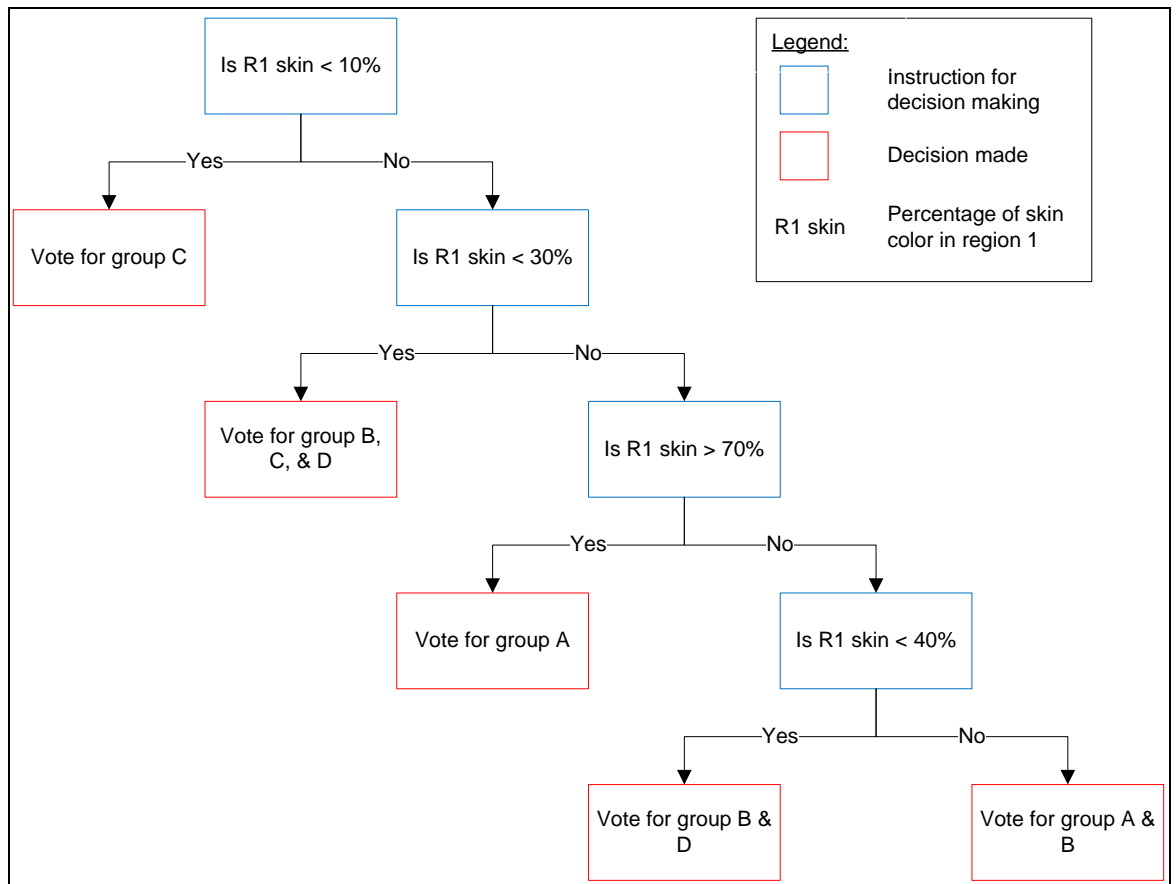


Figure 4.411 Example rule generated

Other than forming rules based on percentage of skin or dark colour, the characteristics between regions can also be used to form a rule. Figure 4.58 shows one of the rules that are formed using the comparison characteristics between regions.

For relationship type of statement, there are only three possible outcomes. It is either two regions are similar in percentage of skin or dark colour or one region contains higher percentage of skin or dark colour. In figure 4.58, the rule is formed using this approach. All the rules in voting session 1 are formed using similar approach described above.

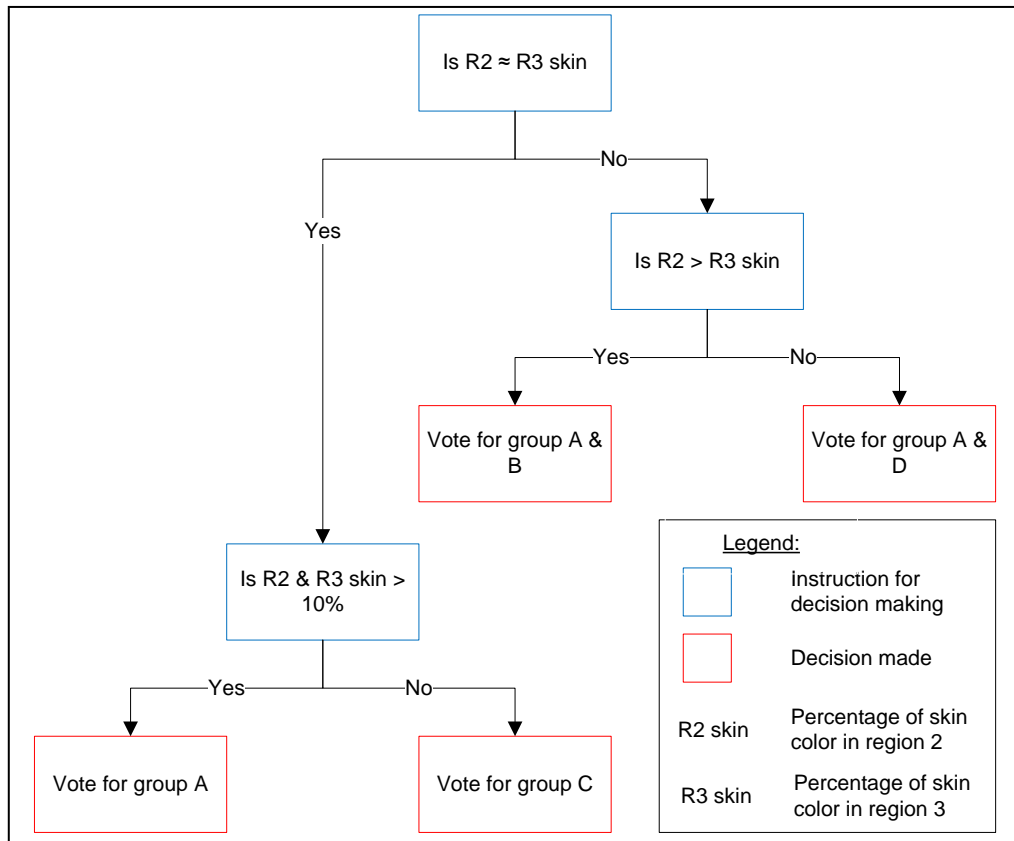


Figure 4.422 Example rule generated

Rules in voting session 2

In voting session 2, it is similar to that of session 1 but instead of classifying the input image into groups, it classifies into specific head pose category that the estimation engine believes.

The rules in voting session 2 are special. Each rule is designed for only specific combination of candidates (possible head pose), since the output from voting session 1 is the possible head pose (2 and above). For example, with group A having the highest vote count and group B having the second highest vote count to that of group A. In this case, the possible head pose would be head pose category 1 and 2 (refer to section 4.9.4.4 for candidate selection) and only the rules that are specifically designed to classify whether the input

image is head pose category 1 or 2 will be used. Figure 4.43 shows the rule for the example above.

The method to construct the rules in voting session 2 is similar to that in session 1. Obverse the unique differences between the candidate and forming of rules in similar way. In some situation where there are three possible head poses, the rules can be built in a simpler way just by reusing the rules for differentiating two possible head poses. For example, the possible head poses are head pose category 1, 2, and 3. The rules for these three possible head poses are shown in figure 4.44.

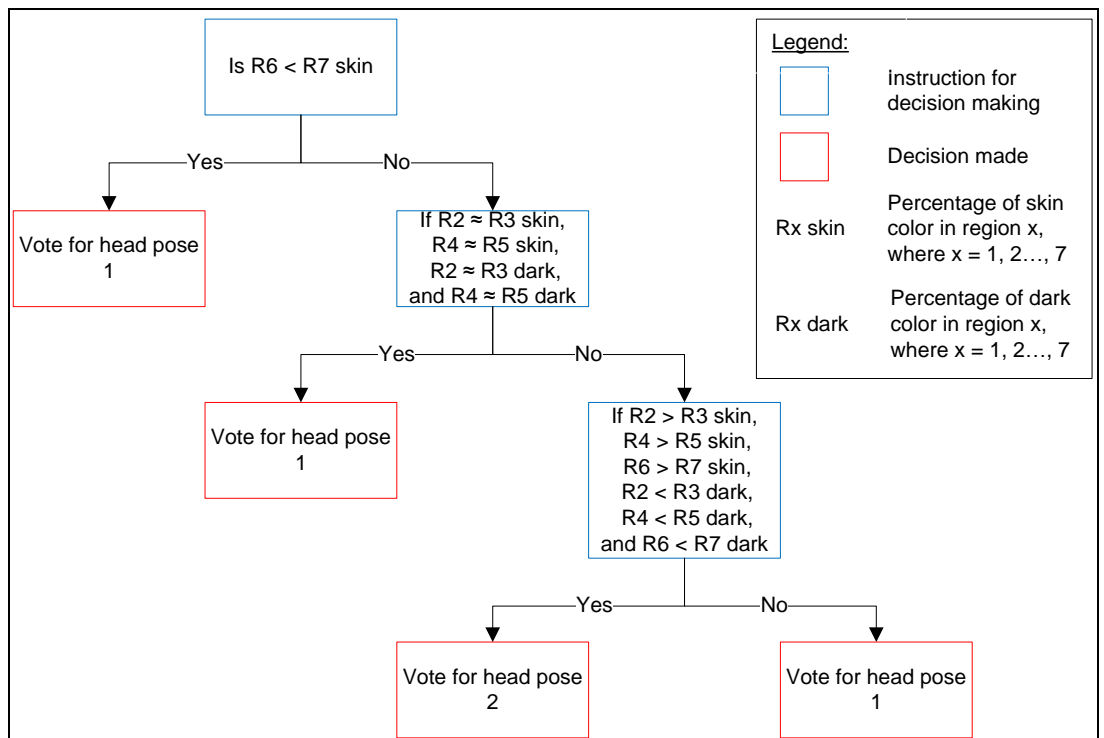


Figure 4.433 Example rule for candidate head pose 1 & 2

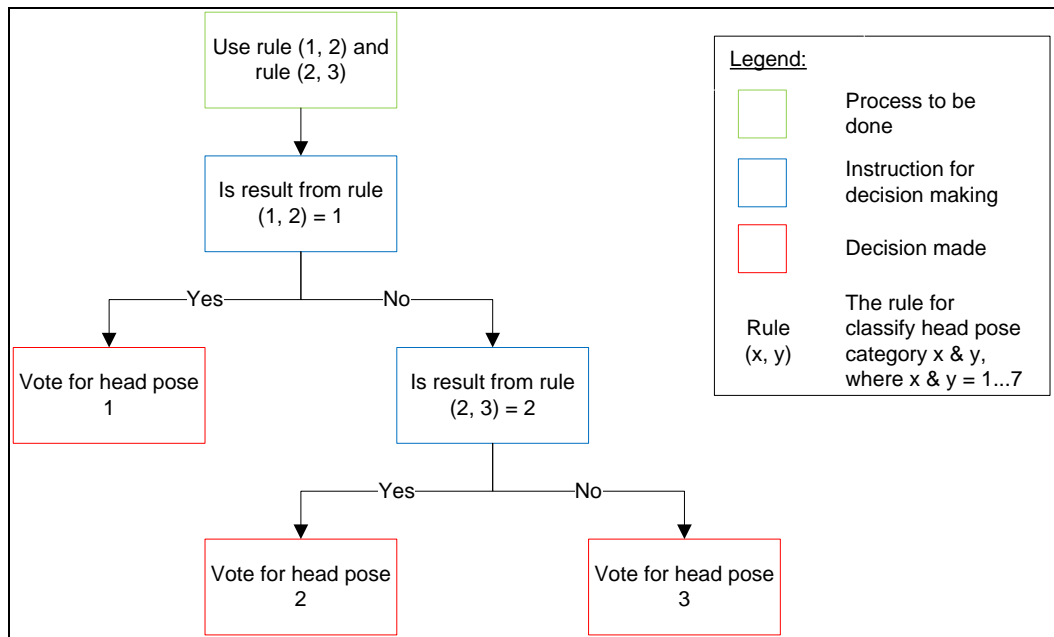


Figure 4.444 Example rule generated for candidate head pose 1, 2 & 3

The green box in figure 4.44 indicates the process to be done before the head pose estimation. The actions are sending the input image into two related rules (rules that are used to classify whether an input image is head pose category 1 or 2 and another rules are for head pose category 2 or 3) and further analyse on the result obtained. This rule is constructed in this way since that the differences between head pose category 1, 2, and 3 are not clear or not enough evidence for classification. This rule can be viewed as using the “divide and conquer” approach that divides the problem into smaller part and solve each part of the problem and finally combine the result. By dividing the problem (classifying input image into one of the three possible head poses), it helps in reducing the complexity of the problem and hence simplifies the classification process. Similarly, all the rules in voting session 2 are constructed.

4.9.4.4 Voting Mechanism

Before the discussion of voting process, the term "candidate" and "voter" have to be defined. In the voting process, a candidate is a person, a thing, or anything that is being voted, and one of the candidates would win the election and being selected to perform some assigned tasks. Voter is to make a decision on choosing the right candidate that best fits to be based on the information they have or what they believed.

In this process of voting, different sessions have different candidate and voters. For voting session 1, the candidate is the group of head pose (four of them) and the voter is the set of rules (16 of them in voting session 1); for voting session 2, the candidate is the specific head pose categories (different based on the input image and the voting result from voting session 1) and the voter is some of the rules out of 16 of them in voting session 2 (due to that not all the rules will be used depending on the situation). In the following discussion, the terms candidate and voters will refer to its meaning in voting session 1 and 2. The discussion starts with a simple flow diagram of the process of voting in the figure 4.45.

As shown in figure 4.45, the voting process starts when there is an input feature. Later, the voters will start to vote for the best candidate based on their own reasoning and belief. Finally the voting result will be sorted and the candidates with the highest and second highest vote counts will be selected. These candidates will be chosen for the next session of voting.

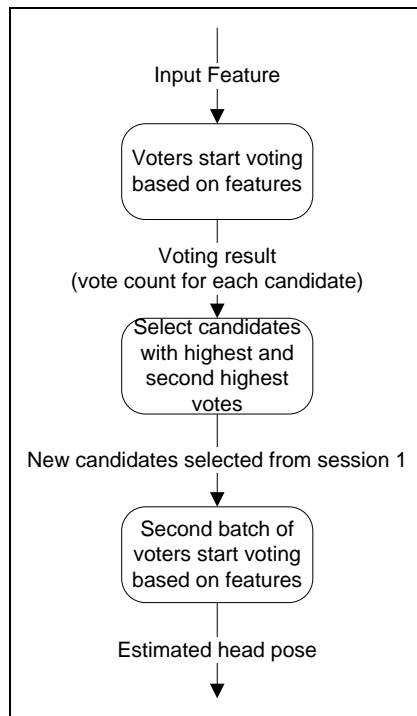


Figure 4.455 Voting process flow

The selection of new set of candidates is based on the result from voting session 1. Since the head poses are interrelated, the voting result from session 1 will have significant meaning for defining the possible head pose of that input image. For example, the highest vote goes to group A and second goes to group B. In this case, the input image can be considered as consisting of a set of characteristics that are similar to group A and some of that of group B. Therefore, the possible head pose for that input image could be that between group A and B (which is head pose category 1 and 2). Here, the head pose category 3 is not included since the highest vote goes to group A, it is more similar to group A compared to group B (base of group A is head pose category 1 and base of group B is head pose category 3). This is how the candidates are selected for the voting session 2. The table 4.5 shows all the candidates to be selected based on the given output from voting session 1.

Table 4.5 Candidate selection for voting session 2

Candidate with highest vote (group)	Candidate with second highest vote (group)	Selected candidates for voting session 2 (head pose category)
A	B	1 & 2
	C	1
	D	1 & 8
	B&C	1 & 2
	B&D	1, 2 & 8
	C&D	1 & 8
	B, C&D	1, 2 & 8
B	A	2 & 3
	C	3 & 4
	D	3
	A&C	2, 3 & 4
	A&D	2 & 3
	C&D	3 & 4
	A, C&D	2, 3 & 4
C	A	5
	B	4 & 5
	D	5 & 6
	A&B	4 & 5
	A&D	5 & 6
	B&D	4, 5 & 6
	A, B&D	4, 5 & 6
D	A	7 & 8
	B	7
	C	6 & 7
	A&B	7 & 8
	A&C	6, 7 & 8
	B&C	6 & 7
	A, B&C	6, 7 & 8

...to be continued

...continued

Table 4.5 Candidate selection for voting session 2

Candidate with highest vote (group)	Candidate with second highest vote (group)	Selected candidates for voting session 2 (head pose category)
A&B	NA	1, 2 & 3
B&C	NA	3, 4 & 5
C&D	NA	5, 6 & 7
A&D	NA	1, 7 & 8

After the candidate for voting session 2 is selected, only specific voters will be allowed to vote for their best candidate, again based on their reasoning and belief. Unlike session 1, voting session 2 will produce only one estimated head pose for the input image. This is how the whole voting process proceeds using the rules created.

4.10 Head Pose Monitoring (P9)

P9 monitors the scene based on the gaze direction (refer to section 4.10.4 for gaze direction calculation) and the movement direction (from P6) of the tracked person. P9 only aims at detecting the constant focus and looks around (left and right) from the direction he or she is moving.

Whenever a detected person remains in the scene for at least 1 second, the tracking information as well as the sequence of estimated head pose will be passed on to P9 for monitoring purpose. Initially, the input sequence of estimated head pose (from P8) will be smoothed in order to obtain a smooth

curve of estimated head pose over time. The smoothing process could reduce the influence of outlier towards the result of P9 (refer to section 4.10.2 for smoothing process).

For ease of calculation and implementation, the information gathered is grouped into blocks of information with duration of 1 second. For instance, if a person enters the scene and remains in the scene for 5 seconds, which would provide a total of 5 blocks of information (each block of information includes the sequence of estimated head pose, movement direction, and gaze direction within that particular time interval).

For each block of information, some adjustment will be made on the sequence of head pose for better analysis work (refer to section 4.10.3 for adjustment process). After all the preprocessing works, the gaze direction will be calculated based on the input information.

If the gaze and movement directions within a block of information are equal, it can be concluded that the person walks toward his/her sighted region. Conversely, it is said that the person walks toward a certain direction with his/her sight focused on another direction that is different from the former if the gaze and movement directions are not equal. For those blocks of information which gaze and movement directions are different, we further divide it into 2 cases:

- Gaze direction is defined, i.e. gaze direction is towards a particular direction other than movement direction.
- Gaze direction is undefined.

4.10.1 Detecting Head Motion

If gaze direction is defined

If the gaze direction is defined, which means that over half of the estimated head pose within a block is of the same category. In this case, the algorithm would only be focusing on checking which direction the person is looking at. For instance, a person walks into the scene and focuses on his left while moving forward, then the action performed by him or her would be constant, i.e. focus on direction 3 (if his movement direction is 5 and gaze direction is 3, from figure 4.15). All the actions performed will be saved for future analysis.

If gaze direction is undefined

If the gaze direction is not found, the system cannot make any decision such as a tracked person having constant gaze on a particular direction. Here, there are only 2 possible outcomes, looks around the surrounding scene within 1 second (1 block of information) or looks at a direction for less than half a second and turn away to face other directions.

To handle this situation, the system makes use of the estimated head poses within the block of information instead of the gaze direction to compare with the movement direction. The system will check whether the estimated head poses consist of facing towards the left of the movement direction as well as facing towards the right of the movement direction. If there is, the system will conclude that the person looks around the surrounding within that 1

second time interval. The possible outcome of head motion analysis for both defined and undefined gaze directions are summarized in table 4.6.

Table 4.6 Possible outcome from detecting head motion

Case no.	Gaze direction (category)	Movement direction (category)	Conditional statement	Output head motion
1	1-8	1-8	If the gaze direction and movement direction are equal.	Normal walking.
2	1-8	1-8	If gaze direction and movement direction are not equal.	Constant gaze on direction x , where x is the gaze direction (category).
3	1-8	Undefined	Automatic.	Constant gaze on direction x , where x is the gaze direction (category).
4	Undefined	1-8	If any two out of the sequence of head pose face the left and right of the movement direction.	Looks around the surrounding.
5	Undefined	Undefined	Automatic.	No head motion.

No matter which outcome was produced, the system will perform the head motion detection again but with a global point of view using all the detected head motions. This global information could be able to pinpoint those head motions (such as looks around the surrounding) that propagate through multiple blocks of information.

P9 will check on each block whether the subject is turning his or her head to the left or right of the movement direction. For instance, the subject is

moving towards the direction 7 and there exists a head pose 4 (any head pose that is ≤ 2 compared to the movement direction), P9 will conclude the block action as the subject turns his or her head to the left. Conversely, if the subject is moving towards the direction 2 and there exists a head pose 4 (any head pose that is ≥ 2 compared to the movement direction), P9 will conclude the block action as the subject turns his or her head to the right. As P9 seeks for information on head motion, the action “turn left” and “turn right” will be combined and a decision will be made that is the subject is looking towards the surrounding area.

Other than checking the turning of head towards left or right of the movement direction, P9 also checks the constant focus region across multiple blocks for detecting possible action of looks around of surrounding. Refer to formula 4.19 and 4.20 in section 4.6.2, if any two out of the sequence of detected head motions could produce $S \geq 3$, then the system will conclude that the person looked around the surrounding within that interval of time (depending on how many blocks of information are propagated).

$$S = abs(a - b), \text{ where } S = [0,4] \text{ and } n = 1,2,3 \quad (4.19)$$

$$S = (8 - S) \bmod 5, \quad \text{if } S > 4 \quad (4.20)$$

Since there is only a simple comparison of the information (gaze and movement directions) for analysis process, therefore P9 has low computational complexity which makes the algorithm suitable to be implemented in real-time intelligent system. The process flow of P9 is illustrated in figure 4.46.

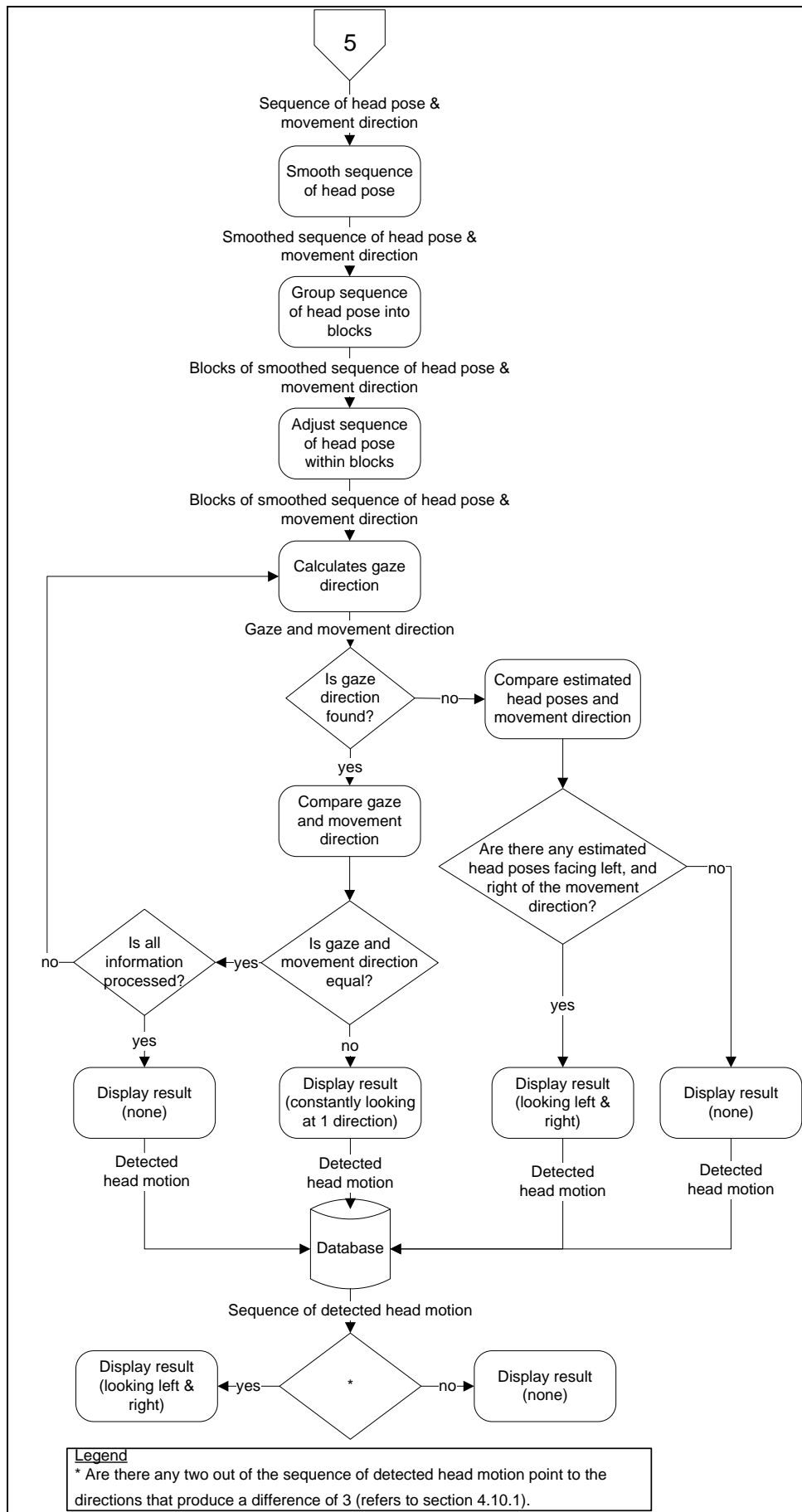


Figure 4.466 P9 process flow

4.10.2 Smoothing of Sequence of Head Poses

As time goes by, the estimated head poses are stored while the person remains in the scene. Those estimated head poses can be represented as a curve of estimated head pose over time. In fact, the algorithm for estimation of head pose may not produce 100% accuracy all the time. Therefore, the curve constructed may not be smooth. An assumption is made here where a person cannot turn their head fast enough ($\geq 90^\circ$ clockwise or counter clockwise in 0.1 second (the system running at 10 frames per second, 0.1 second for each frame)), so that the curve produced should be smooth all the time. Example of smooth curve and non-smooth curve are illustrated in figure 4.47 and 4.48 respectively.

The smoothing mechanism will try to transform the example as illustrated in figure 4.48 into a smooth curve as illustrated in figure 4.49. The transformation only involves the comparison of neighbouring nodes (estimated head pose of the curve), find the differences between neighbouring nodes and current node and change the node value if needed. This smoothing process can help in reducing the influence of outliers (illustrated in figure 4.48, time frame number 4) towards the scene analysis process such as calculate the gaze direction, determine the head movement of a person whether he or she is looking to the surrounding. In addition, this might also be able to reduce the incorrect estimation of head pose. The steps involved in smoothing process are illustrated in figure 4.50.

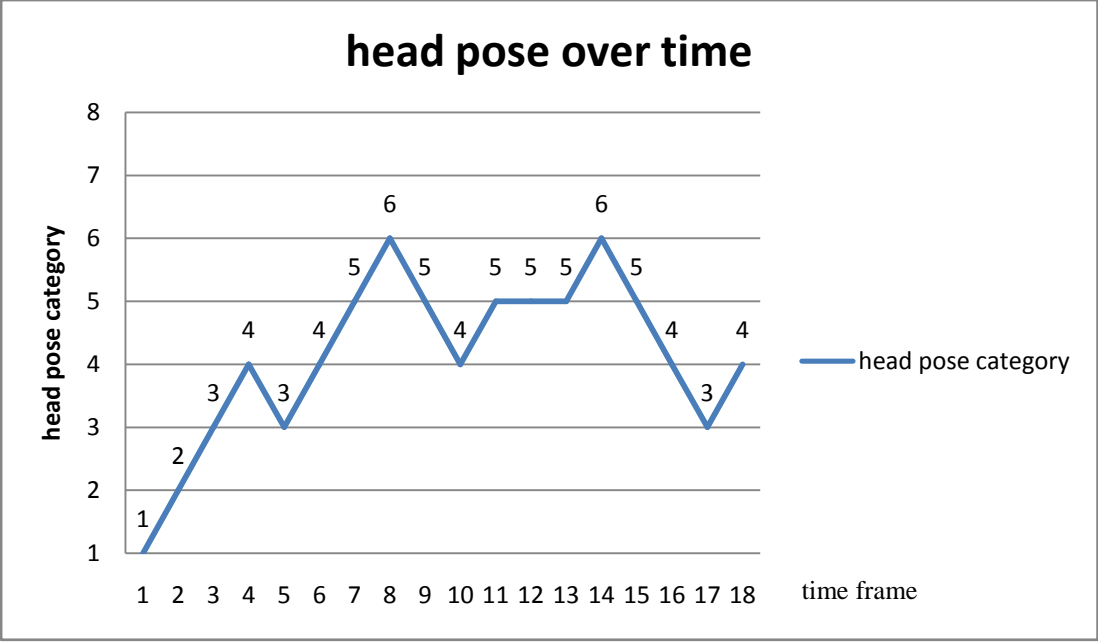


Figure 4.477 Example of smooth curve

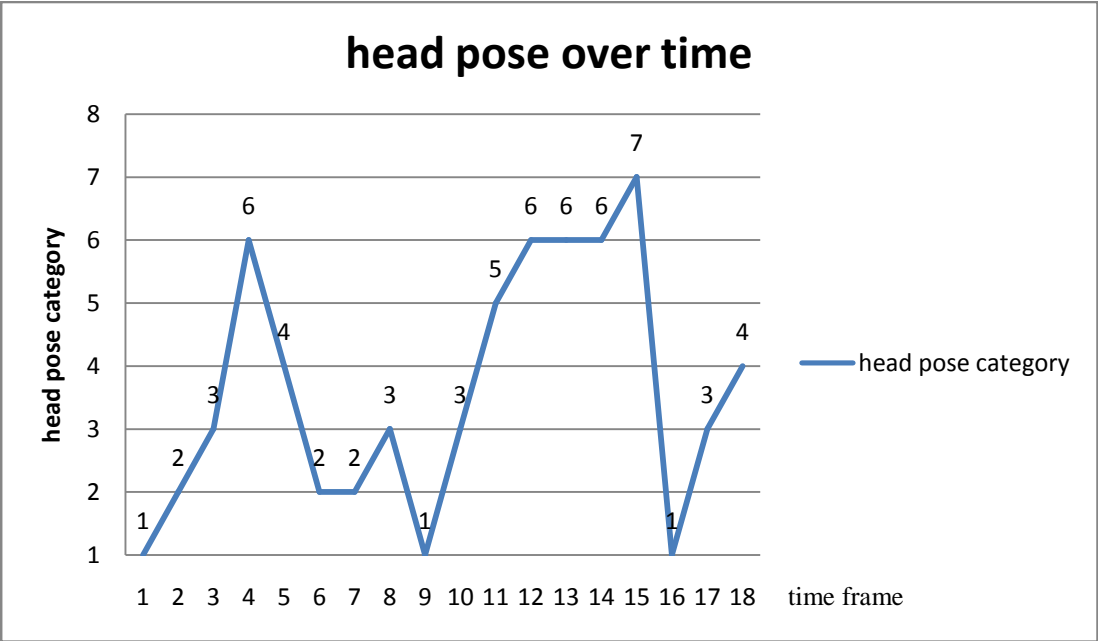


Figure 4.488 Example of non-smooth curve

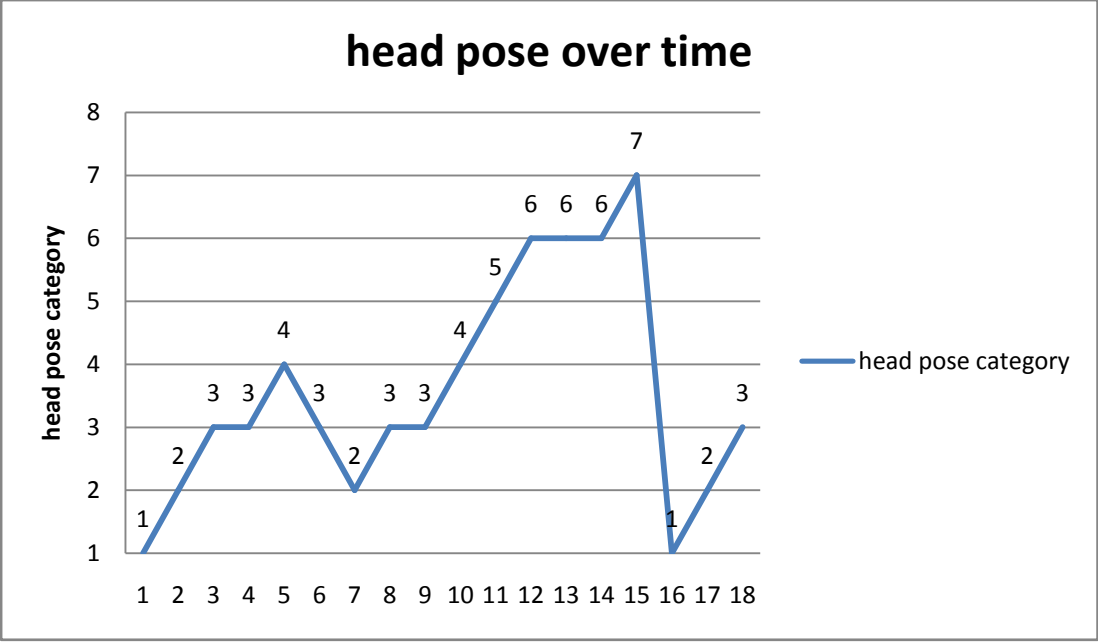


Figure 4.4949 Smoothed curve

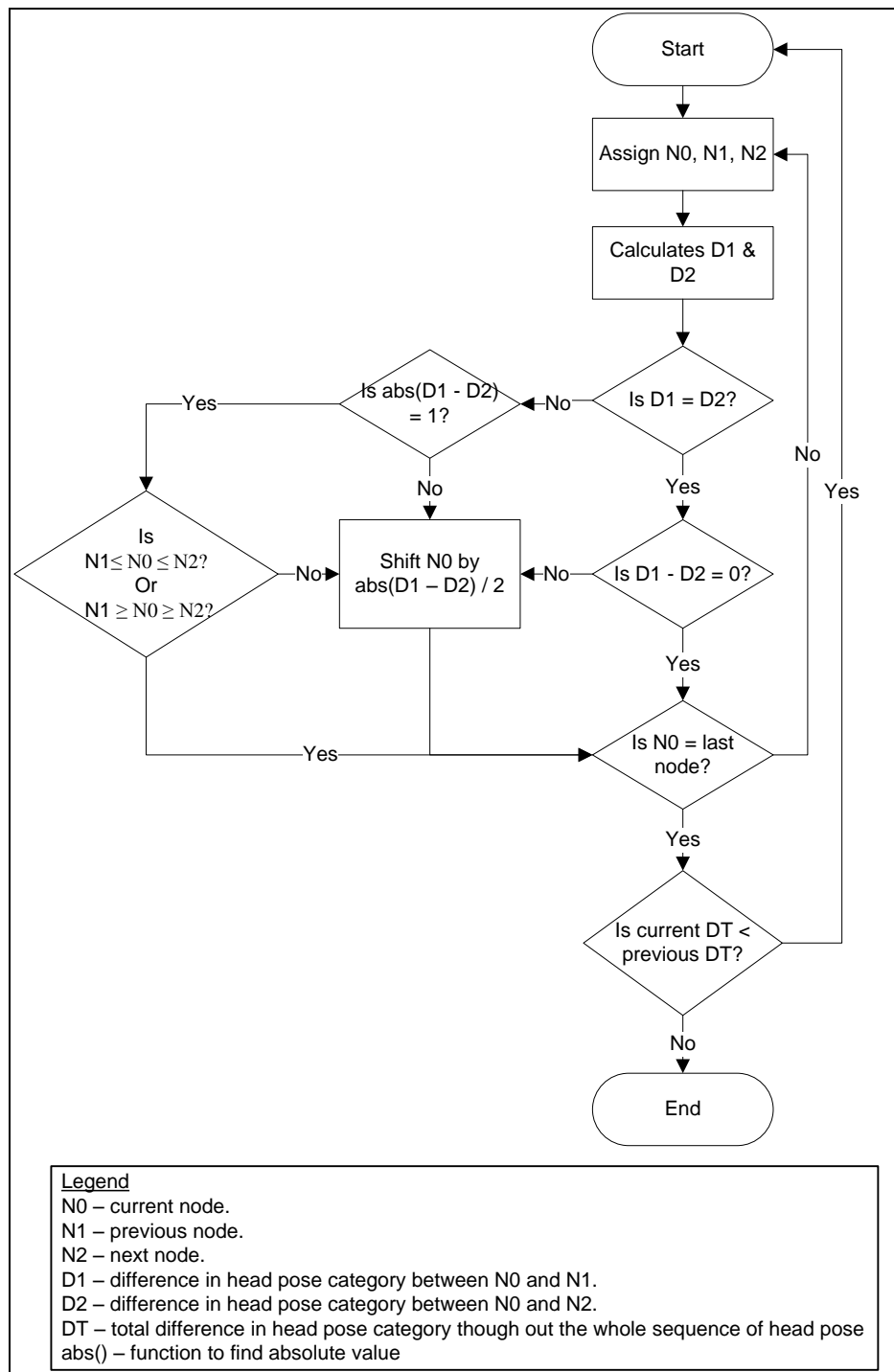


Figure 4.50 Smoothing process flow

4.10.3 Adjustment of Sequence of Head Poses

While the sequence of head pose within a block of information generated meets the similar cases as illustrated in figure 4.51 or figure 4.52, some adjustment needs to be done for ease of implementation of scene analysis process.

Notice that there is a “jump” of curve in both figure 4.51 and 4.52 when the head pose of the person changes from category 8 to 1 or vice versa. Refer to the figure 4.23 in section 4.9, there are 8 categories for estimated head pose. Each head pose is interrelated to its neighbouring head pose (e.g. head pose category 5 is related to head pose category 4 and 6) by rotating the subject’s head 45° to the left or right. Therefore, when arranging the sequence of head pose into graph for analysis, the “jump” of curve may exist when the person changes his or her head pose from category 1 to 8 or 8 to 1. It is also hard to interpret the curve information with the presence of “jump” of curve.

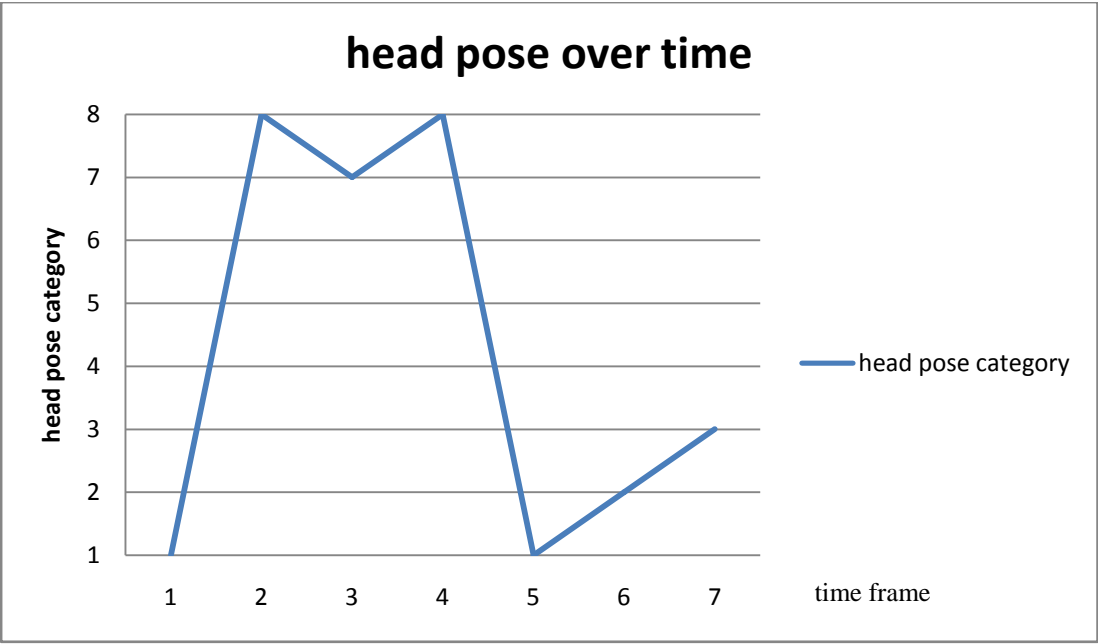


Figure 4.501 Head pose over time graph

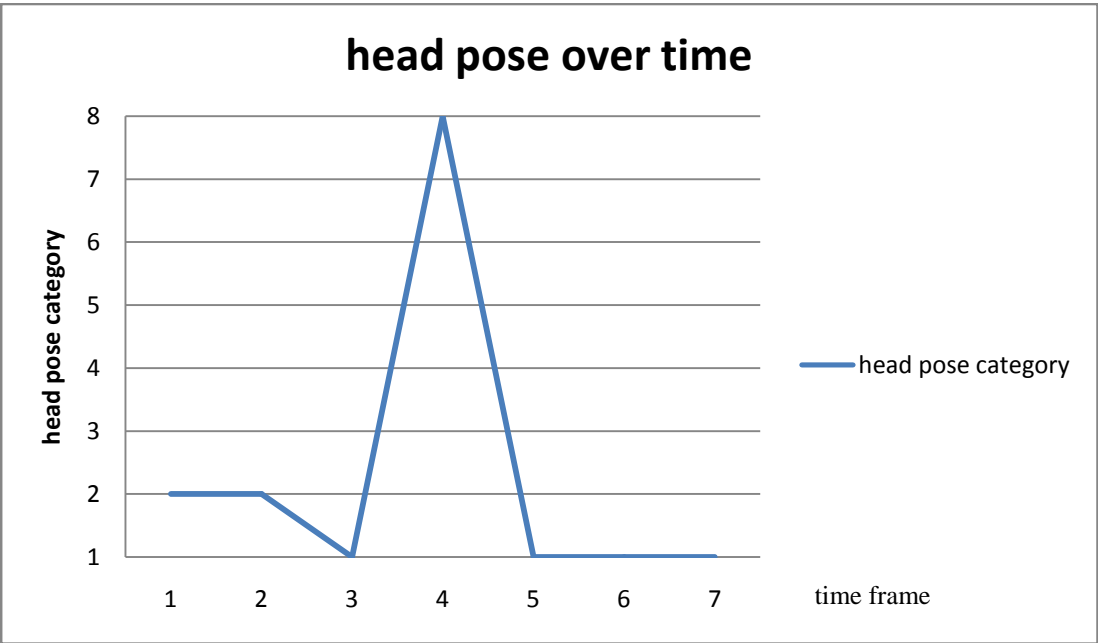


Figure 4.512 Head pose over time graph

To better represent the sequence of head pose for analysis process, adjustment will be made such as shifting part of the curve upwards so that the curve appears to be “linked”. Figure 4.53 and 4.54 illustrate the head pose

over time graph after applying the adjustment for the case in figure 4.51 and 4.52 respectively.

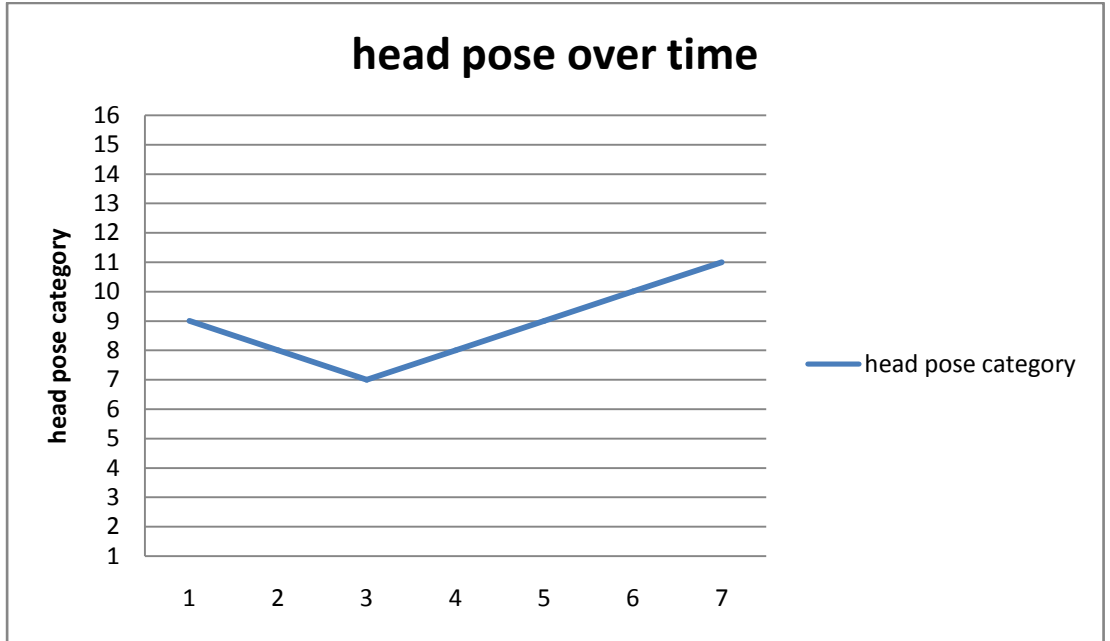


Figure 4.523 Adjusted head pose over time graph

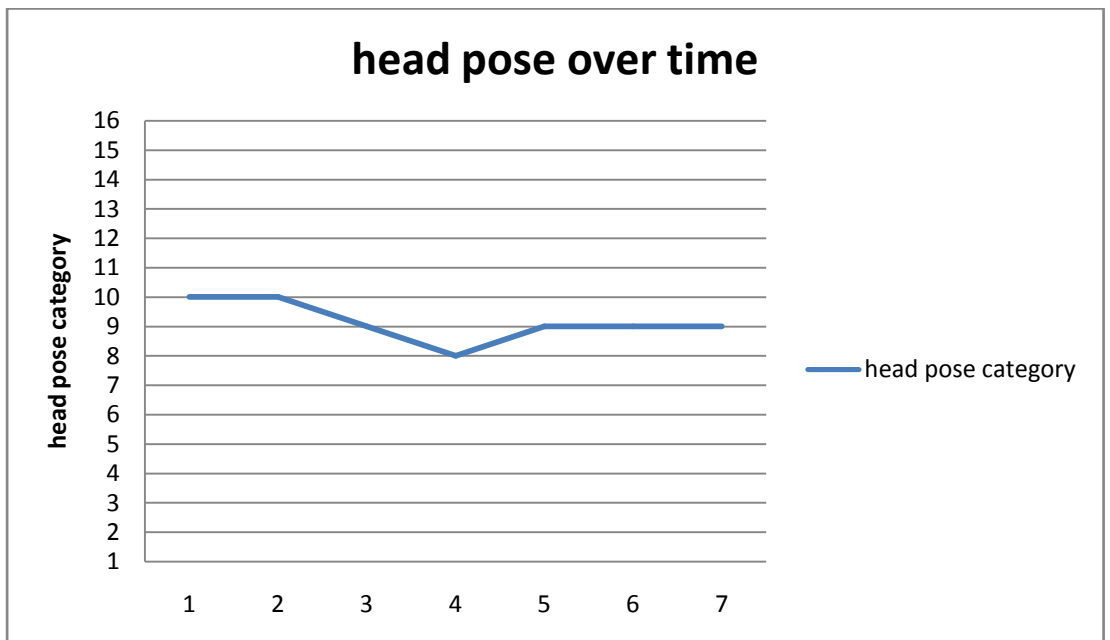


Figure 4.534 Adjusted head pose over time graph

In figure 4.53 and 4.54, the head pose category is doubled (number category of head pose originally is 8), this is due to the presence of “jump” of curve. To remove the “jump” of curve, part of the curve is moved upwards by 8 (which is the number of category of head pose).

Let the movement direction be 9 (originally is 1 but after the adjustment, the movement direction is also plus another 8 because the head pose category is doubled), the sequence of head pose in figure 4.53 shows that the person looks around the surrounding by turning his or her head to right then left; whereas for the sequence of head pose in figure 4.54 shows that the person looks slightly to the right then turn his or her head back to face the front. This analysis information can be easier to obtain after the adjustment. This adjustment also simplifies the implementation of scene monitoring process.

4.10.4 Gaze Direction Calculation

Similar to the motion direction calculation in P6, for every t new locations of the tracked human calculated using the tracking algorithm, the gaze direction of the human will be calculated based on the sequence of estimated head pose using the formula 4.21. The value t varies based on the frames per second of the input video signal, it is fixed to one second for ease of calculation.

$$Gazedirection = \begin{cases} x & , \text{ If } \geq 50\% \text{ of head pose } x \text{ found within } t \\ -1 & , \text{ otherwise} \end{cases} \quad (4.21)$$

where x is the category of head pose.

4.11 Conclusion

This concludes the discussion about the implementation of system. As discussed, system threshold values are the key for the system to work properly. It has to be tuned according to the environment before putting the system to work. In the next chapter, a few sets of threshold values are used for testing purposes and the result of system threshold values tuning is listed for each testing environment.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter will discuss the testing process of the proposed system including testing environment, testing data, testing result, and a discussion on the result obtained.

A total of 5 test cases were carefully selected from the whole dataset to test the proposed system. These 5 test cases were defined as 1 best case (BC), 3 normal cases (NC), and 1 worst case (WC). Those cases were listed below:

1. (BC) One person walks into the scene that looks to the left and right of his or her movement direction and leaves the scene.
2. (NC) One person walks into the scene that looks to the left and right of his or her movement direction (walks in a straight line) and leaves the scene.
3. (NC) One person walks into the scene and remains static at a random location. During that particular period, he or she looks towards a direction for 1 to 2 seconds before leaving the scene.
4. (NC) Two persons walk into the scene one after another and leave the scene with same gaze and movement directions.

5. (WC) Three persons talk into the scene one after another and leave the scene. All three persons having the same gaze as his or her movement directions.

5.2 Testing Environment

Proposed system is aimed for the implementation in real-time environment. Thus, the testing of the proposed system was took place in the school compound using equipments i.e. camera, laptop, etc. and tools i.e. visual studio, OpenCV, etc. described in the following subsection.

5.2.1 Testing Equipment

Several tools were used during the testing process such as a computer that is installed with the proposed system and a camera. The camera was placed at a higher location with roughly 30° from the ceiling to act as a mounted CCTV camera. The configuration of the equipment was illustrated in figure 5.1.



Figure 5.1 Configuration of testing equipment

5.2.2 Testing Tools

Tools that were used for the testing process includes the main tool: *visual studio professional edition 2008*; and two libraries: *OpenCV 2.0 library* (OpenCV2.0, 2009) for processing images and *SVMLight library* (Joachims, 1999) for building SVM classifier.

5.2.3 Testing Data

From the literature reviewed in chapter 2, the datasets used by scholars to train or test their proposed algorithm includes AMI dataset, CHIL dataset, Pointing'04 dataset, CMU-PIE dataset, FERET dataset, CAS-PEAL face

dataset, self-captured dataset, etc. Among the testing dataset mentioned, only CHIL dataset consists of subjects moving in the scene with varying pose, other dataset such as AMI, Pointing'04, CMU-PIE, FERET, and CAS-PEAL face dataset only consist of subjects who faced the camera with varying pose.

The proposed system was designed to perform in real-time basis in which the testing data should be in the form of video or sequence of images. Furthermore, there must be humans (with visible head and shoulder) moving or standing while in the scene. To test the monitoring algorithm (P9), the humans that appear in the scene should consist of 3 different actions such as looks normally (looks towards the direction of movement), looks around the surroundings and constant focus on a particular direction.

Scholars often use the most suitable dataset which best fits into the criteria of their proposed algorithm. Hence, in this research project, self-captured testing data was used for testing purpose since none of the datasets used by scholars were suitable that meets the criteria of the system.

The self-captured data consist of people with certain actions performed that was designed based on the module of the proposed system, i.e. video of people who walk into the scene with constant head pose or look around the surrounding area. The following subsections show the preparation of testing data and the specification of those data.

5.2.3.1 Testing Data Preparation

With the equipment configured as shown in figure 5.1, the testing data were captured in different environment setting, such as the lighting condition and the entry and exit points of the scene. Some of the data was captured in the afternoon in which the lightning condition for those data was under normal sunlight; some of the data was captured in the morning, those data were under the low light condition. Figure 5.2 shows the example of the captured scene with indication for entry and exit points.






Figure 5.2 Example scenes with entry and exit points (in red arrow)

There were a total of 12 actors who act during the data collection process. Those 12 actors enter and exit the scene according to the predefined

scheme of act for data collection. The 17 profiles of those 12 actors are shown in table 5.1 and the predefined scheme of act is shown in table 5.2.




Table 5.1 Profiles of actors

Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
1	White coloured top with long black trunk.	
2	Dark blue coloured top with long black trunk.	
3	White coloured top with long brown trunk. An additional backpack as accessory.	

To be continued...

...continued




Table 5.1 Profile of actors

Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
4	Light blue coloured top with long black trunk. An additional backpack as accessory.	
5	White coloured top with short green pant.	
6	Dark coloured top with short gray pant. An additional cap as accessory.	

To be continued...

...continued




Table 5.1 Profile of actors

Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
7	Red coloured top with short gray pant.	
8	Dark stripe coloured top with long brown trunk.	
9	Blue coloured top with long light blue trunk.	

To be continued...

...continued




Table 5.1 Profile of actors

Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
10	Dark stripe coloured top with long black trunk.	
11	Pink coloured top with long white trunk.	
12	Green coloured top with long gripe trunk.	

To be continued...

...continued

Table 5.1 Profile of actors

Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
13	Light green coloured top with long blue trunk.	
14	Red stripe coloured top with long gray trunk.	
15	Black coloured top with long black trunk.	

To be continued...

...continued

Table 5.1 Profile of actors



Actor no.	Cloth colour	Screen shot of actor (in green ellipse)
16	Red coloured top with long blue trunk.	
17	Purple coloured top with long blue trunk.	

Table 5.2 Predefined scheme of act

Scheme no.	No. of subjects	Acts
1	1*	Subject enters the scene and leaves the scene. Subject looks towards the direction of movement.
2	1*	Subject enters the scene and leaves the scene. Subject looks towards the direction other then the movement.
3	1*	Subject enters the scene and leaves the scene. Subject looks around the surrounding area.
4	1*	Subject enters the scene and remains static. Subject looks towards a direction while remains static in the scene.
5	1*	Subject enters the scene and remains static. Subject looks around the surrounding while remains static in the scene.

Remarks:

* if there are more than 1 subject, subjects are following any combination of the scheme of act

5.2.3.2 Testing Data Specification

The 5 testing data discussed are stored as videos of type “.avi”. All the videos are of same resolution, i.e. 320×240 pixels and 30 frames per second. However, each video is variants in length.

5.3 System Evaluation Based On Test Cases

The experiments were conducted using 5 test cases as stated in the introduction. The experiments will test the proposed system thoroughly i.e. test for each processes such as P4, P5, P6, P7, P8, and P9. The most suitable evaluation mechanism will be used to assess each process. Processes P1, P2, and P3 were not included for evaluation due to the reason that those processes are some preliminary steps or pre-processing of the proposed module 1 and 2, i.e. *human detection and tracking* and *scene monitoring based on sequence of estimated head pose*.

The characteristics and some information extracted from the testing videos of test cases were summarized in table 5.3.

Table 5.3 Characteristics of testing videos

Attribute	Test Case 1 (BC)	Test Case 2 (NC)	Test Case 3 (NC)	Test Case 4 (NC)	Test Case 5 (WC)
Video Format	<i>Avi</i>	<i>Avi</i>	<i>Avi</i>	<i>Avi</i>	<i>Avi</i>
Dimension	320×240	320×240	320×240	320×240	320×240
Frames Count	450	274	535	240	430
Video Length	00:00:15	00:00:09	00:00:18	00:00:08	00:00:14

To be continued...

...continued

Table 5.3 Characteristics of Testing Videos

Attribute	Test Case 1 (BC)	Test Case 2 (NC)	Test Case 3 (NC)	Test Case 4 (NC)	Test Case 5 (WC)
No. of Subject	1	1	1	2	3
Subject Enter/Exit on Frame Number	134/416	60/274	108/524	(S1) 48/150; (S2) 140/230	(S1) 30/150; (S2) 188/280; (S3) 356/430
Frames Count/Processed When Subject in Scene	283/141	215/107	417/208	(S1) 103/52; (S2) 91/45	(S1) 121/60; (S2) 93/46; (S3) 75/37
P4					
Total Number of Human Detection Process (<i>detection count</i>)	253,025	188,092	326,793	175,953	287,926
Total Number of <i>False Positive</i> Classification (<i>False positive count</i>)	0	0	0	0	0
Total Number of <i>False Negative</i> Classification (<i>False negative count</i>)	756	360	216	300	1692
P5					
Total Number of <i>Positive</i> Classification (<i>Positive count</i>)	142	107	208	(S1) 52; (S2) 45	(S1) 0; (S2) 8; (S3) 0
Total Number of <i>False Positive</i> Classification (<i>False positive count</i>)	2	8	16	(S1) 2; (S2) 7	(S1) 0; (S2) 6; (S3) 0
P7					
Total Number of Head Localize Process (<i>detection count</i>)	141	107	208	97	NA
Total Number of <i>False Positive</i> Classification (<i>False positive count</i>)	1	10	21	3	NA
Total Number of <i>False Negative</i> Classification (<i>False negative count</i>)	58	21	12	14	NA

Remark:

(S#) is the information for subject #. i.e. (S1) refers to subject 1.

Refer to table 5.3, the *total number of human detection process* is the number of occurrence of the human detection process throughout the whole testing video, i.e. detect human when there is motion (chapter 4, section 4.4

Motion Blob Segmentation), detect human when the reliability of the tracked position is low (chapter 4, section 4.6 *Human Tracking*). For all the detection of human, those motions that are wrongly classified as human would be counted as *total number of false positive*. When the motions are human, but the classifier classifies those as background or non-human object, it would contribute to the *total number of false negative*.

The *Total Number of Positive Classification* under P5 in table 5.3 represents the total number of frames that the tracker tracks the subject in scene; and the *Total Number of False Positive Classification* is the total number of frames that the head and shoulder part of the subject was outside of the region of interest of the tracker.

As for the attributes under P7 in table 5.3, *Total Number of Head Localize Process* represents how many times the P7 was initiated to locate the head position of the tracked subject. *Total Number of False Positive Classification* represents the number of frames that the output of P7 is not the position of the head of the tracked subject; whereas for the *Total Number of False Negative Classification* represents the number of frames that P7 is unable to identify the head position of the tracked subject. With the information gathered (as shown in table 5.3), some evaluation mechanisms can be used to evaluate the proposed system.

Evaluation mechanism for P4

To evaluate P4, three mechanisms were used such as false positive per window (*FPPW*), false negative per window (*FNPW*) with scale factor of 1.05, and the detection rate. A few terms listed in the following list will be used in the following discussion in section 5.3-*Evaluation mechanism for P4*.

- *Positive* – indicating the decision made by human classifier that is the input for P4 is classified as “human” (regardless of true or false decision).
- *Negative* – indicating the decision made by human classifier that is the input for P4 is classified as “non-human” (regardless of true or false decision).
- *False Positive* – Indicating that the decision made by human classifier (*Positive*) is incorrect.
- *False negative* – Indicating that the decision made by human classifier (*Negative*) is incorrect.

Whenever a human is detected, it would then be handled by the dedicated tracker. The region around the human (additional half of width of the detected human towards the top, bottom, left and right of the position of the tracked human) would not be included for human detection process anymore. Thus, this could affect the number of true positive generated. Therefore, evaluation using the false positive rate (*FPR*), false negative rate (*FNR*), true positive rate (*TPR*), and true negative rate (*TNR*) from Health Decision Strategies (2011) are not suitable for the evaluation of the proposed human detection algorithm.

To evaluate human detection algorithm, $FPPW$ and $FNPW$ will be calculated. $FPPW$ is the total number of false positives divides by the total number of detection windows on a particular testing video. The output figure will represent the average amount of false positives per detection window. Similarly, $FNPW$ is the total number of false negatives divides by the total number of detection windows on a particular testing video. The output figure will represent the average amount of false negatives per detection window.

The formulas 5.1 and 5.2 are used to calculate the $FPPW$ and $FNPW$ respectively.

$$FPPW = \frac{\text{false positive count}}{\text{total number of detection window}} \quad (5.1)$$

$$FNPW = \frac{\text{false negative count}}{\text{total number of detection window}} \quad (5.2)$$

Analysis can be conducted by referring the calculated value of $FPPW$ and $FNPW$. The lower the values of $FPPW$ and $FNPW$, the better the classification result of the algorithm. However, assessing $FPPW$ and $FNPW$ could not fully identify the performance of the human detector. This is due to the large amount of detection process in the testing video where most of the detections fall on the background or uninterested part of the subject (such as leg, body, etc.). Only a small portion of the detection process are accessing the pixels where is the object of interest (in this research work, the object of interest is the head and shoulder part of the human). In this case, producing low $FNPW$ is common. Thus, another evaluation mechanism was added to further illustrates the performance of the human detection.

The third evaluation mechanism was used to calculate the detection rate of the algorithm (percentage of detected human when the subjects appear in the scene). The formula 5.3 illustrates the formula to obtain the detection rate. When the detection rate is low, it indicates that the algorithm is unable to detect the presence of humans in most of the time even though the *FNPW* is low. The higher the detection rate, the better the performance of the algorithm.

$$\begin{aligned} & \textit{detection rate} = \\ & \frac{\textit{total number of frames that all the humans in the scene were detected}}{\textit{total number of frames that contain humans}} \end{aligned} \quad (5.3)$$

Other than accuracy test, the speed of computation of P4 was also examined.

Evaluation mechanism for P5

A novel evaluation mechanism was presented to evaluate P5. The evaluation mechanism used was named as *Euclidean distance error of tracked window per frame* (ETWPF). ETWPF calculates the normalized distance between the position of the tracked window and the actual position of the tracked object. The farther the distance, the larger the error. When ETWPF is 1, the tracked position is totally out from the object. The ETWPF can be calculated using the formula 5.4.

$$ETWPF = \frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}}{\textit{human size}} \quad (5.4)$$

$$aETWPF = \frac{\sum_{i=p}^{i=q} ETWPF}{q-p+1} \quad (5.5)$$

where $x_n, n = 1,2$, is the center of x -coordinate of tracked window and center of x -coordinate of actual position of tracked object; p represents the frame number when the object of interest is tracked; and q is the frame number when the object of interest leaves the scene or the video ends; *human size* is the diagonal distance of the bounding rectangle of tracked object.

After the calculation of ETWPF on P5, the average of ETWPF (formula 5.5) will be calculated to illustrate the average of deviation of tracked window towards the object of interest. As the name suggests, the lower the value of aETWPF, the more precise the algorithm of P5.

Solely rely on the novel evaluation mechanism stated previously to evaluate the P5 would not be enough as it only indicates how much the tracked position deviates from the actual position of human. For this proposed system, as long as the head and shoulder part of human is being tracked (bound in the rectangle by tracker), the process P7 can estimates the head position of the tracked person and P8 can estimates the head pose. Thus, even though the tracked position is slightly deviates from the actual position, it is acceptable as long as the head and shoulder part of the human is still obtainable within the bounding rectangle of the tracker. Therefore, the average of true positive rate, *aTPR* and average of false positive rate, *aFPR* are calculated to indicate how frequently the tracked location contains head and shoulder part of human. A few terms listed in the following list will be used in the following discussion in section 5.3-*Evaluation mechanism for P5*.

- *True Positive* – Indicating that the decision made by tracking mechanism that is the head and shoulder part of human is within the bounded position of tracker.
- *False Positive* – Indicating that the decision made by tracking mechanism that is either the head or shoulder part of human is out of the bounded position of the tracker (must be 100% in the bound position).

In some test cases, there was more than one person that would appear in the scene. Therefore, the average of rating (*aFPR* and *aTPR*) were used for the evaluation of P5 instead of *FPR* and *TPR*. *FPR* and *TPR* were meant for evaluation on a single person and the average of rating were meant for the evaluation of the whole testing video. The *FPR*, *TPR*, *aFPR* and *aTPR* were calculated using formulas 5.6, 5.7, 5.8 and 5.9 respectively.

$$FPR_i = \frac{\text{false positive count}_i}{\text{true positive count}_i + \text{false negative count}_i} \quad (5.6)$$

$$TPR_i = \frac{\text{true positive count}_i}{\text{true positive count}_i + \text{false negative count}_i} \quad (5.7)$$

$$aFPR = \frac{\sum_{n=1}^{n=\text{total instances}} FPR_n}{\text{total instances}} \quad (5.8)$$

$$aTPR = \frac{\sum_{n=1}^{n=\text{total instances}} TPR_n}{\text{total instances}} \quad (5.9)$$

where *i* is the indicators for tracked humans.

The *FPR* and *TPR* were both calculated by dividing the *false positive count + true positive count* is due to that whenever a tracker is initialized for a subject, it would always produce either a

true positive or *false positive*. Therefore, the formulas 5.6 and 5.7 were defined as they were.

Evaluation mechanism for P6

Previously in chapter 4, P6 calculates the direction of motion for the person being tracked as one of the nine categories of direction (as shown in chapter 4, section 4.7, figure 4.15) every second. To measure the performance of P6, modified version of the formulas 4.6 and 4.8 (in chapter 4, section 4.6.2), formulas 5.10 and 5.11 were used.

$$E = \text{abs}(a - b), \text{ where } E = [0,4] \quad (5.10)$$

$$E = (8 - E) \text{ mod } 5, \text{ if } E > 4 \quad (5.11)$$

where E is the error value, a and b are the direction calculated from P6 and actual direction respectively. Note that if there is no direction i.e. the object of interest remains static, that is directionless, the value of direction (a or b) will be -1. The maximum and minimum values of E are 4 and 0 respectively.

The E calculated with respect to blocks of time (each block is equal to 1 second) will indicate the deviation of the direction calculated from P6 from the actual direction of motion. The smaller the value of E , the better the algorithm of P6.

Evaluation mechanism for P7

Evaluations of P7 were conducted using *FPR* and *FNR*. However, the 2 mechanisms mentioned from Health Decision Strategies (2011) are not suitable for the evaluation. P7 was highly dependent on the result from P5 that

is the tracked position. P7 will try to locate the head position within the tracked position of the human. Hence, if the tracked position was deviated from the target, the P7 might produce more false positive count. Thus, the modified version of *FPR* and *FNR* as shown in formulas 5.12 and 5.13 were used for the evaluation of P7. A few terms listed in the following list will be used in the following discussion in section 5.3-*Evaluation mechanism for P7*.

- *Positive* – indicating the decision made by human head detector that is the head position is found on the input for P7 (regardless of true or false decision).
- *Negative* – indicating the decision made by human head detector that is the head position is not found on the input for P7 (regardless of true or false decision).
- *True Positive* – Indicating that the decision made by human head detector (*Positive*) is correct.
- *True Negative* – Indicating that the decision made by human head detector (*Negative*) is correct.
- *False Positive* – Indicating that the decision made by human head detector (*Positive*) is incorrect.
- *False negative* – Indicating that the decision made by human head detector (*Negative*) is incorrect.

Using these formulas (5.12 and 5.13) for evaluation of P7, it would be clear that how much percentage of the localization process fails in locating the head position; how frequently the algorithm of P7 fails, etc. The higher the value of *FPR* and *FNR*, the more error produced from the algorithm of P7.

$$FPR = \frac{\text{false positive count}}{\text{total decision made}} \quad (5.12)$$

$$FNR = \frac{\text{false negative count}}{\text{total decision made}} \quad (5.13)$$

Evaluation mechanism for P8

For P8, it categorizes the input head image from P7 into one of the eight categories of head pose. Since the head poses defined (in chapter 4, section 4.9, figure 4.23) are interrelated, it is more suitable to evaluate the classification of P8 using the mechanism similar to the one for P6.

The formulas 5.14 and 5.15 showed the calculation of error in classification of head pose in P8.

$$E = \text{abs}(a - b), \text{ where } E = [0,4] \quad (5.14)$$

$$E = (8 - E) \text{ mod } 5, \text{ if } E > 4 \quad (5.15)$$

where E is the error value, a and b are the head pose estimated from P8 and actual head pose respectively. Note that if there is no head pose estimated i.e. the input head image does not contain head information (misclassification of head position in P7), the estimation of P8 might fail and return a head pose of category -1 that is no head pose. Thus, the value of head pose (a or b) will be -1. The maximum and minimum values of E are 4 and 0 respectively.

A graph would be drawn to illustrate the error, E over time (number of frame) and the average of error, \bar{aE} will be calculated to indicate the overall deviation of estimated head pose from the actual head pose throughout the whole testing data.

Evaluation mechanism for P9

Evaluation of P9 is very subjective. There is no standard evaluation mechanism yet for P9 to measure the accuracy or precision of the algorithm. Hence, the evaluation of P9 has to be done literally, that is analyses on a series of frames (testing video) and describes the monitoring process of P9 whether it makes a correct decision on the action done by the actors. Evaluation on P9 also reflects the performance of the whole proposed system as P9 is the last process of the proposed system. Any deficiency suffered from previous processes i.e. P1 to P8, will directly affect the result of P9. If the evaluation result of P9 is good, that means all the processes perform well; conversely, if the evaluation result of P9 is poor, that could means one or more of the processes are suffered in deficiency. Any poor decision made on the previous processes of P9 would result in poor performance of P9.

5.3.1 Evaluation Using Test Case 1

Some frames from the testing video for test case 1 were illustrated in figure 5.3. This test case consists of a person (wearing dark colour cloth with long pant) that walks into the scene at frame numbered 134 with same gaze and movement direction. Later on, his movement stops at a random location starting at frame numbered 200 and turn his gaze towards a direction of his back. After 1 to 2 seconds, that person continued his path to collect an object on the floor and leave the scene. Some screen shots of the execution of proposed system on test case 1 were shown in figure 5.4.



Figure 5.3 Frames from testing video of test case 1



Figure 5.4 Screen shots of program execution on testing video of test case 1

Using the evaluation mechanisms defined in section 5.3, the evaluation was conducted for processes of the proposed system based on the information in table 5.3.

5.3.1.1 Evaluation of P4

The three evaluation mechanisms for P4 were calculated as following:

$$FPPW = \frac{0}{253,025} = 0$$

$$FNPW = \frac{756}{253,025} = 0.003 = 3 \times 10^{-3}$$

$$detection\ rate = \frac{78}{141} = 5.5319 \times 10^{-1}$$

Out of 141 frames that contain human, P4 was only able to locate the subject in 78 frames that was more than 55%. The low detection rate was due

to the body shape of the subject (when the subject remains static, the side view of the subject hardens the human detection process as the training samples are more focused on the front and back view of a human) and the bending action (the subject bend down to collect an object on the floor). Nevertheless, with zero *FPPW*, there was no non-human object being classified as human in this test case. In addition, the *FNPW* was nearly zero as well. With a successive of over 250k times of human classification process, only 756 of them were falsely classified, that is classifies a human object as non-human object.

The computational speed of P4 is illustrated in figure 5.5.

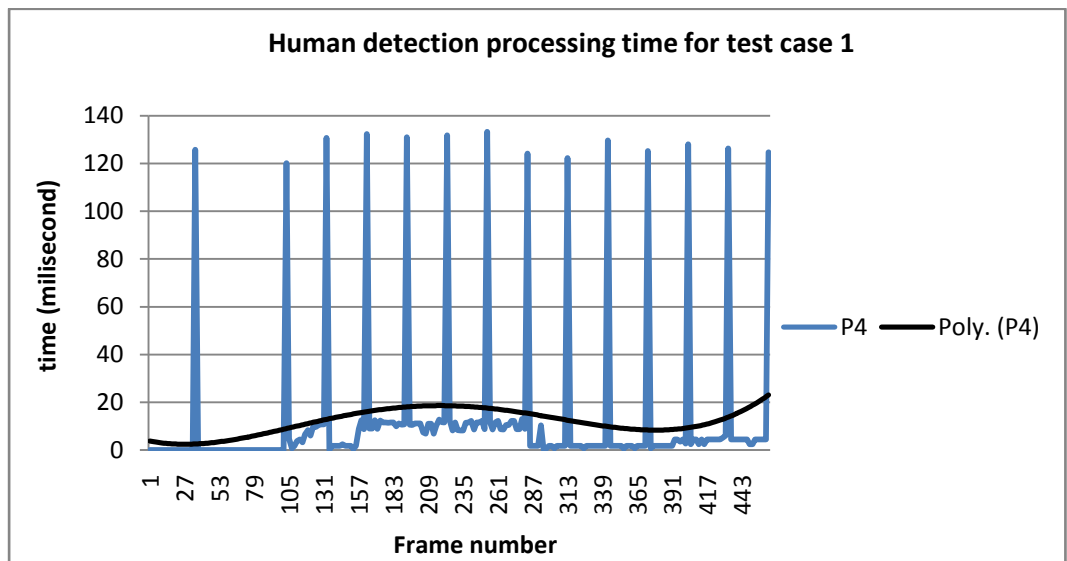


Figure 5.5 Human detection processing time for test case 1

The blue line in figure 5.5 represents the computational time for the proposed human detection process, P4 with a trend line that was in black colour.

Most of the computational speed of P4 was less than 20 milliseconds per frame. Notice that there are spikes in the graph that was more than 120 milliseconds of computational time per frame. This was due to the periodic human classification process that invoked periodically of one second time interval. The average of computational speed of P4 for test case 1 was 11.7796 milliseconds per frame.

5.3.1.2 Evaluation of P5

As for evaluation of human tracking algorithm, the *ETWPF* over time graph for test case 1 was shown in figure 5.6.

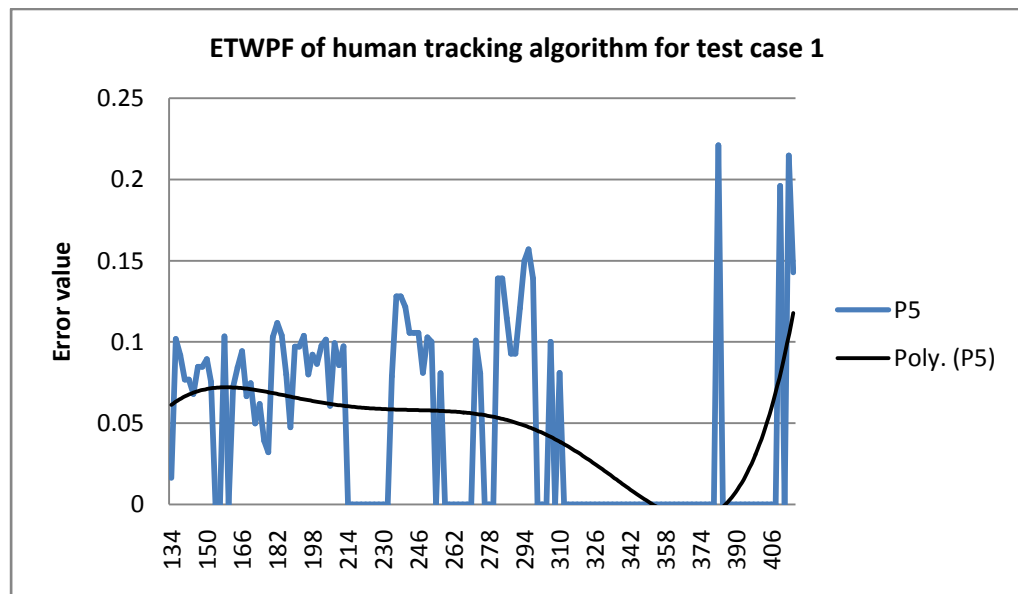


Figure 5.6 *ETWPF* evaluation over time graph for human tracking algorithm on test case 1

The blue line in figure 5.6 represents the error (Euclidean distance between object of interest and tracked position) in tracking by the proposed human tracking process, P5 with a trend line that was in black colour.

The average of $ETWPF$ value of P5 for test case 1 was calculated as following:

$$aETWPF_{P5} = \frac{6.4154}{142} = 0.0452 = 4.52 \times 10^{-2}$$

Refer to the $aETWPF$ values calculated, the average of errors for P5 was less than 0.05 which indicates that the tracker were almost perfectly tracks the object of interest with a margin of error of about 5% deviation from the actual target.

Secondly, to test whether the tracker always have the information of head and shoulder, the $aFPR$ and $aTPR$ were calculated for P5.

$$aFPR_{P5} = FPR_{P5} = \frac{2}{142} = 0.014085 = 1.4085 \times 10^{-2}$$

$$aTPR_{P5} = TPR_{P5} = 1 - 1.4085 \times 10^{-2} = 9.8593 \times 10^{-1}$$

Note that only 2 out of 142 frames were false positive that was either the head or shoulder part of human was out of the bounded position of the tracker. This figure shows that the tracker was able to contain the head and shoulder of the object of interest within the tracked position 98.6% of the time. Summary of evaluation of P5 was shown in table 5.4.

Table 5.4 Summary of evaluation of human tracking algorithm for test case 1

Evaluation Mechanism	Values of evaluation of P5
$\alpha ETWPF$	4.52×10^{-2}
αFPR	1.4085×10^{-2}
αTPR	9.8593×10^{-1}

5.3.1.3 Evaluation of P6

The graph of error, E for P6 was shown in figure 5.7 which indicates the error in motion direction calculation for each blocks of information.

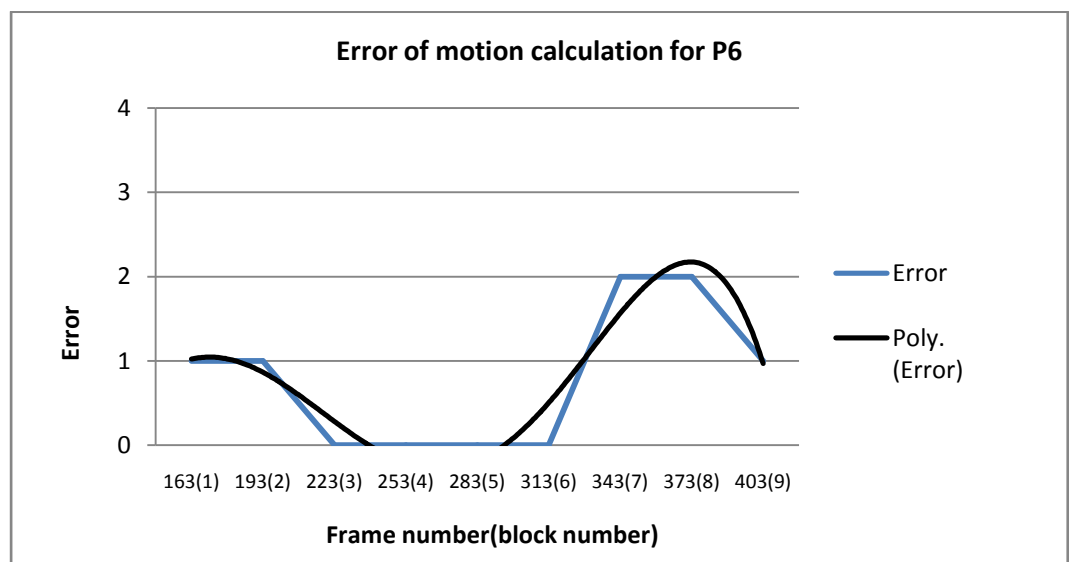


Figure 5.7 Error of motion calculation for P6

From the figure 5.7, the blue line indicates the error of motion direction calculation from P6 over time. The trend of the error over time was drawn as the black line in the same figure. The subject enters the scene on frame numbered 134, hence the first block of information was formed on the 163th frame ($134 + 30 = 164$, where the 30 was the frames per second of the

testing video). There were only 9 blocks collected for test case 1. The screen shots of the system execution while each block was formed are illustrated in figure 5.8.

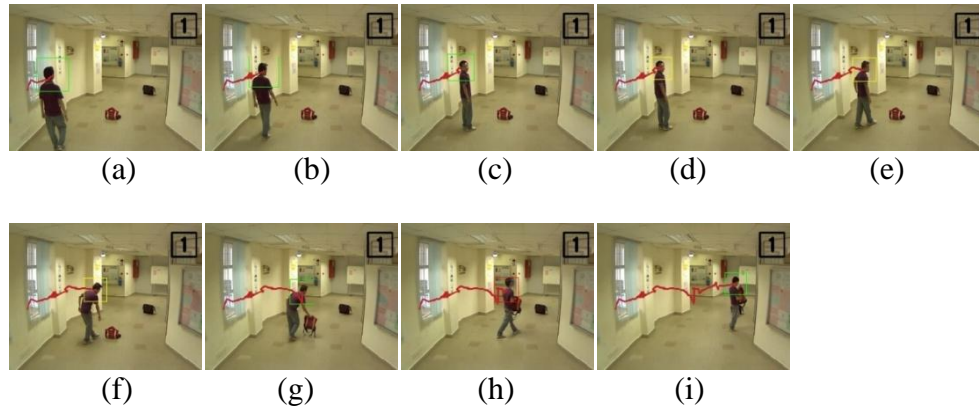


Figure 5.8 Screen shots of system execution which each block was formed

Notice that on blocks numbered 7 and 8 in figure 5.7, the error in motion direction calculation was 2. This was due to the action done by the subject i.e. picking up the object on the floor as shown in figure 5.8 (g) and (i). Deduced from 5.8 (f) and (g), the block numbered 7 consists of a path (in red line in 5.8) information that was shaped “V”. In this case, the P6 could not produce accurate result since the most occurrence of local direction will be chosen as the motion calculated of that particular block of information.

The tracking of human’s head and shoulder produced the irregular path for the subject in the scene due to the similar reason that was the subject picks up the object on the floor. Even though the subject remains static on that particular moment to pick up the object, the tracker follows the head and shoulder of the subject which was bent down and back to normal after picked

up the object. Thus, the system would mistakenly regards the bending action as the subject moved nearer or further from the camera. This would lead the process P6 in making the incorrect decision. The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{1 + 1 + 0 + 0 + 0 + 0 + 2 + 2 + 1}{9} = 7.7778 \times 10^{-1}$$

The P6 calculates the motion direction of the subject in test case 1 accurately with an average of 7.7778×10^{-1} calculation error (8 directions in total and maximum of error = 4).

5.3.1.4 Evaluation of P7

The two evaluation mechanisms for P7 were calculated as following:

$$FPR = \frac{1}{141} = 0.0071 = 7.0922 \times 10^{-3}$$

$$FNR = \frac{58}{141} = 0.4113 = 4.1135 \times 10^{-1}$$

Refer to figure 5.8, when the subject bent down to collect the object on the floor, the head position was unable to be located since the shape of the body was different from normal i.e. standing, normal walking.

From the calculation above, P7 produced a lot of false negative result that was the P7 could not locate the position of the head for more than 40% of the time. This result also means that almost half of the time when the subject was in the scene, the head position was undefined. This would affect the subsequent processes i.e. P8 and P9. Without the head position, P8 could not

proceed with the head pose estimation process and this would further influence the monitoring process. Nevertheless, the *FPR* was low in which only 1 head localized was located on the incorrect position. This result indicates that the P7 was able to accurately locating the head position of the tracked human, however the hit rate was low.

5.3.1.5 Evaluation of P8

The head pose estimation for each input from P7 was illustrated in figure 5.9. The blue dots in the figure 5.9 represent the estimated head pose and the black line represents the trend of the estimated head pose. Notice that some of the head pose are blank that was no estimated head pose for a certain frame. This could due to the previous process i.e. P7 that did not successfully locating the head position of the tracked human and hence no input to P8 for estimation process. The figure 5.10 shows the ground truth of the head poses for comparison.

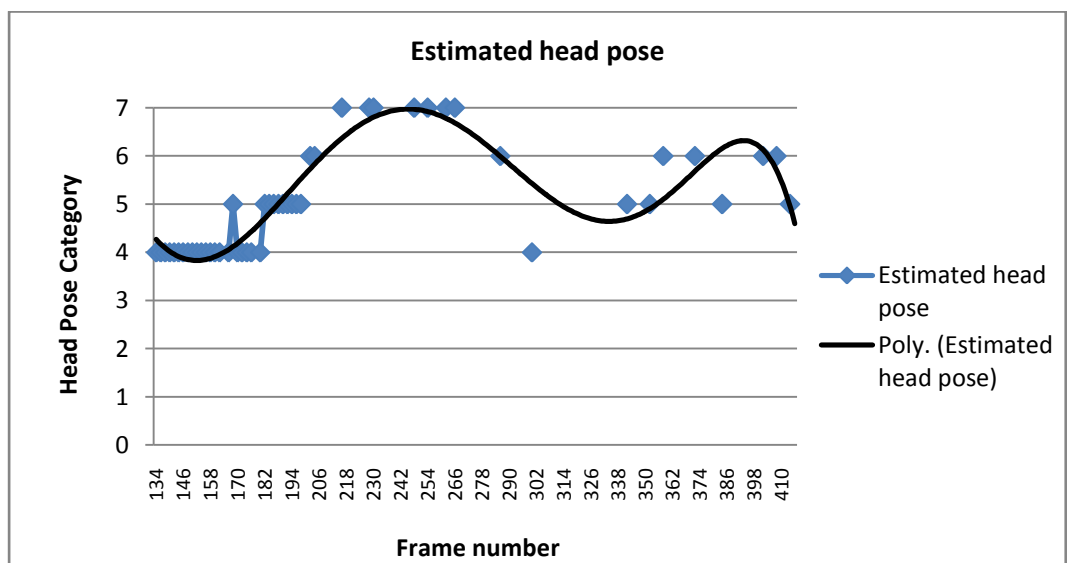


Figure 5.9 Estimated head pose for test case 1

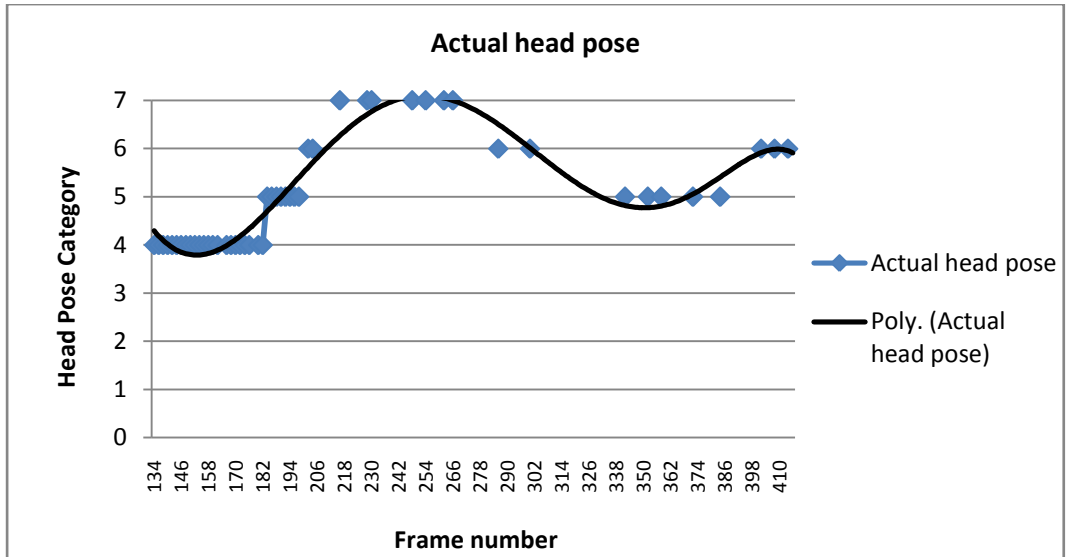


Figure 5.10 Actual head pose for test case 1

Notice that there was only a slight variation on frame 170 and some frames near the end of the graph. Nevertheless, the overall pattern of the curves in figure 5.9 and 5.10 were similar.

The graph of error, E for P8 was shown in figure 5.11 which indicates the error in head pose estimation for each input from P7.

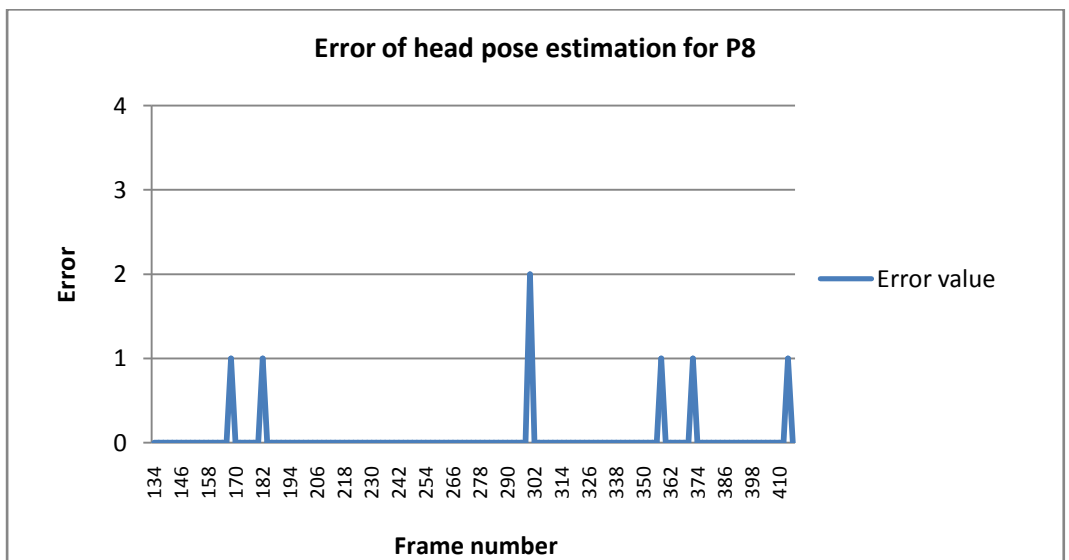


Figure 5.11 Error of head pose estimation for P8

The error values shown in figure 5.11 were calculated using formula 5.14 and 5.15. Here, it was clear that the head pose estimation result was good as most of the error values were zero that was correct estimation. The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{1 + 1 + 2 + 1 + 1 + 1}{50} = 0.14 = 1.4 \times 10^{-1}$$

Among 142 frames (the time period that the subject appears in the scene), P7 was only able to locate the head position of the subject on 50 frames. Hence, the aE was calculated with a divider of 50. For test case 1, only 6 out of the 50 estimation process by P8 was deviated from the ground truth result. This shows the effectiveness of the algorithm of P8 in estimating the head pose of the subject who appears in the scene. The P8 estimates the head pose of the subject in test case 1 accurately with an average of 1.4×10^{-1} estimation error (8 directions in total and maximum of error = 4).

5.3.1.6 Evaluation of P9

As described in section 5.3.2.5, the head poses were not always been successfully estimated and stored as inputs P9. This could due to the reason that P7 fails to locate the head position of the subject, or the estimation from P8 deviates from the actual head pose. This would affect the process of P9. Hence, the estimated head poses from P8 was being smoothed for ease of processing. The collection of head poses and the smoothed version of collection of head poses were illustrated in figure 5.9 and 5.12.

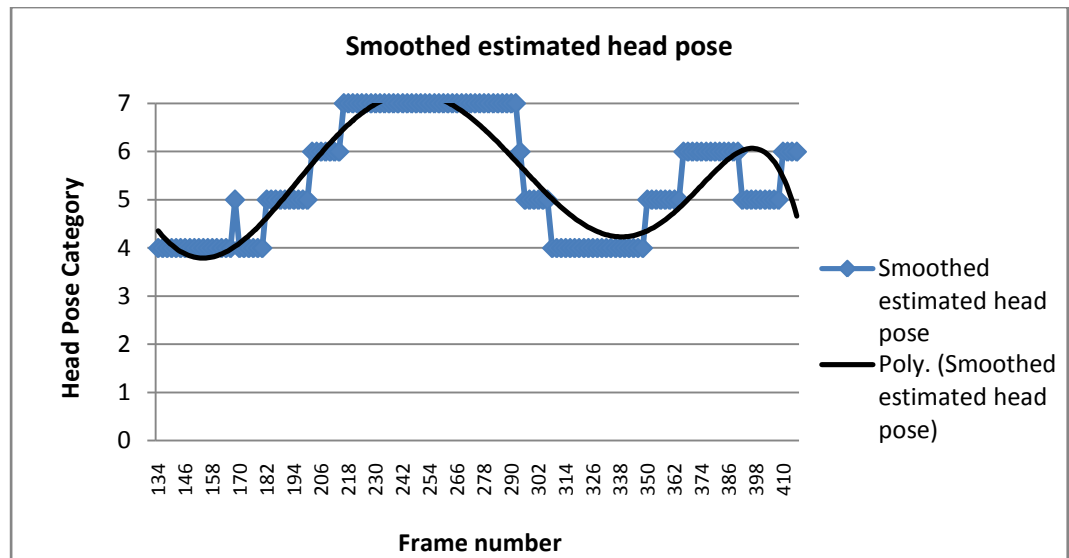


Figure 5.12 Smoothed estimated head poses

For every blocks of information, P9 detects the actions i.e. constant gaze on a particular location, looks around the surrounding area, or looks towards the direction of motion, of the tracked person. Therefore, to better analyse and discuss on the test case, the classification result from P9 throughout the whole testing video were arranged and summarized in table 5.5. The starting and ending frame for each block of information will be specified (that was 30 frames for each block) with the classification result. A short discussion on the result obtained will be included in the table 5.5 as well. The column *Gaze Direction* was calculated from the smoothed sequence of head pose and the column *Movement Direction* are obtained from P6. There were only 3 options to be filled into the column *Result from P9* and *Ground Truth* which were LS (looks straight towards a direction other than the direction of movement), LA (looks around the surrounding), and LD (looks towards the direction of movement).

Table 5.5 Summary of result from P9 for test case 1

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
1	134/163	<p>134 138 142 146 150 154 158 162</p>	4	4	LD	LD
2	164/193	<p>164 168 172 176 180 184 188 192</p>	4	4	LD	LD
3	194/223	<p>194 198 202 206 210 214 218 222</p>	6	5	LD	LD
4	224/253	<p>224 228 232 236 240 244 248 252</p>	7	-	LS	LS
5	254/283	<p>254 258 262 266 270 274 278 282</p>	7	-	LS	LS
6	284/313	<p>284 288 292 296 300 304 308 312</p>	-	6	LA	LA

To be continued...

...continued

Table 5.5 Summary of Result from P9

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
7	314/343	<p>314 318 322 326 330 334 338 342</p>	4	7	LA	LS
8	344/373	<p>344 348 352 356 360 364 368 372</p>	5	4	LD	LD
9	374/403	<p>374 378 382 386 390 394 398 402</p>	6	5	LD	LD

From table 5.5, most of the decision made by P9 was correct other than the decision on block 7. From the beginning, block numbered 1, 2, and 3 were quite straight forward that was the subject moves in a direction with his gaze focused at his front. Later on, the subject stops at a particular location and looks towards the direction 7 while remains static.

Notice that the block numbered 6 was classified as LA which was correct due to the pre-action done on block numbered 3. On block number 3, the subject turn his head to the right followed by turning his head back to the left on block numbered 6 as shown in the sequence of head pose in table 5.5.

This triggered the monitoring system to classify the action by subject that was looking at the surrounding.

For block-wise decision on block numbered 7, the subject was LS towards the direction 4 while moving towards the direction 7. However, the gaze towards direction 4 on block numbered 7 was regarded as the continuous of the LA action on block numbered 3 to 6. Therefore, the classifier misclassifies the LS action as LA. Other than the misclassification on block numbered 7, everything was classified correctly.

5.3.2 Evaluation Using Test Case 2

Some frames from the testing video for test case 2 were illustrated in figure 5.13. This test case consists of a person (wearing white colour cloth with long pant) that walks into the scene at frame numbered 60 and looks to the left and right on the way to the other exit of the scene. Later on, he exits the scene on frame numbered 274. Some screen shots of the execution of proposed system on test case 2 were shown in figure 5.14.



Figure 5.13 Frames from testing video of test case 2



Figure 5.14 Screen shots of program execution on testing video of test case

2

Using the evaluation mechanisms defined in section 5.3, the evaluation was conducted for processes of the proposed system based on the information in table 5.3.

5.3.2.1 Evaluation of P4

The three evaluation mechanisms for P4 were calculated as following:

$$FPPW = \frac{0}{188,092} = 0$$

$$FNPW = \frac{360}{188,092} = 0.0019 = 1.914 \times 10^{-3}$$

$$detection\ rate = \frac{77}{107} = 0.7196 = 7.196 \times 10^{-1}$$

Out of 107 frames that contain human, P4 was able to locate all the subjects in 77 frames that were more than 70%. With zero *FPPW*, there was no non-human object being classified as human in this test case. In addition, the *FNPW* was nearly zero as well. With a successive of over 180k times of human classification process, only 360 of them were falsely classified, that was classifies a human object as non-human object. Notice that the subject in the scene was wearing white cloth which makes the background subtraction

algorithm in P1 suffers from deficiency. The motion detected was poor due to the white colour cloth. However, P4 still able to maintain low *FNP*.

The computational speed of P4 was illustrated in figure 5.15.

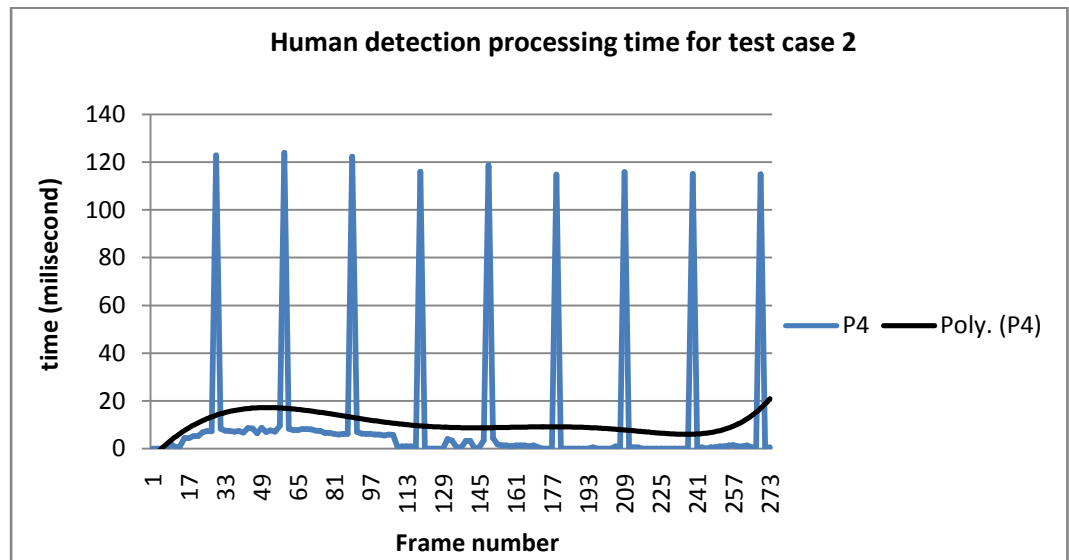


Figure 5.15 Human detection processing time for test case 2

The blue line in figure 5.15 represents the computational time for the proposed human detection process, P4 with a trend line that was in black colour.

Most of the computational speed of P4 was less than 20 milliseconds per frame. Notice that there were spikes in the graph that was more than 120 milliseconds of computational time per frame. This was due to the periodic human classification process that invoked periodically of one second time interval. The average of computational speed of P4 for test case 2 was 10.3799 milliseconds per frame.

5.3.2.2 Evaluation of P5

As for evaluation of human tracking algorithm, the *ETWPF* over time graph for test case 2 was shown in figure 5.16.

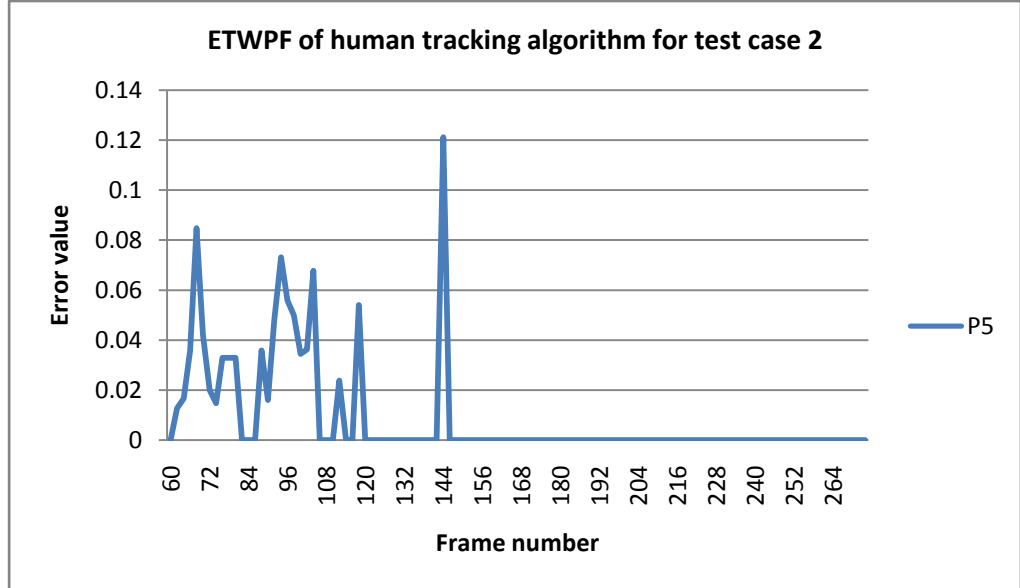


Figure 5.16 *ETWPF* evaluation over time graph for human tracking algorithm on test case 2

The blue line in figure 5.16 represents the error (Euclidean distance between object of interest and tracked position) in tracking by the proposed human tracking process, P5 with a trend line that was in black colour.

The average of *ETWPF* value of P5 for test case 2 were calculated as following:

$$aETWPF_{P5} = \frac{0.9428}{107} = 0.0088 = 8.8 \times 10^{-3}$$

Refer to the *aETWPF* values calculated, the average of errors for P5 was less than 0.01 which indicates that the tracker were almost perfectly tracks

the object of interest with a margin of error of less than 1% deviation from the actual target.

Secondly, to test whether the tracker always have the information of head and shoulder, the $aFPR$ and $aTPR$ was calculated for P5.

$$aFPR_{P5} = FPR_{P5} = \frac{8}{107} = 0.0748 = 7.48 \times 10^{-2}$$

$$aTPR_{P5} = TPR_{P5} = 1 - 7.48 \times 10^{-2} = 9.252 \times 10^{-1}$$

Note that only 8 out of 107 frames were false positive that was either the head or shoulder part of human was out of the bounded position of the tracker. This figure shows that the tracker was able to contain the head and shoulder of the object of interest within the tracked position 92.52% of the time. Summary of evaluation of P5 was shown in table 5.6.

Table 5.6 Summary of evaluation of human tracking algorithm for test case 2

Evaluation Mechanism	Values of evaluation of P5
$aETWPF$	8.8×10^{-3}
$aFPR$	7.48×10^{-2}
$aTPR$	9.252×10^{-1}

5.3.2.3 Evaluation of P6

The graph of error, E for P6 was shown in figure 5.17 which indicates the error in motion direction calculation for each blocks of information.

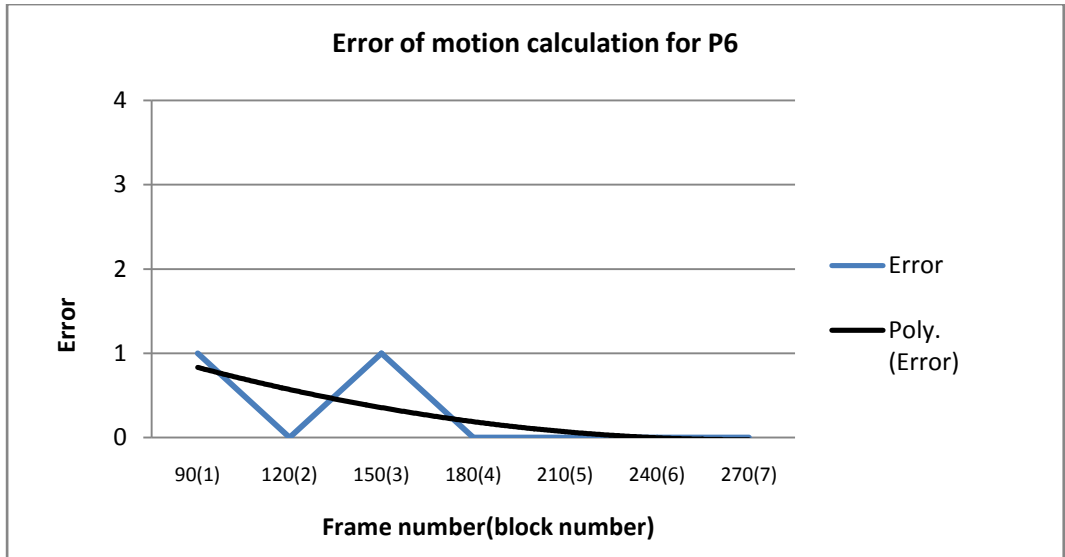


Figure 5.17 Error of motion calculation for P6

From the figure 5.17, the blue line indicates the error of motion direction calculation from P6 over time. The trend of the error over time was drawn as the black line in the same figure. The subject enters the scene on frame numbered 60, hence the first block of information was formed on the 90th frame ($60 + 30 = 90$, where the 30 was the frames per second of the testing video). There were only 7 blocks collected for test case 2. The screen shots of the system execution while each block was formed were illustrated in figure 5.18.

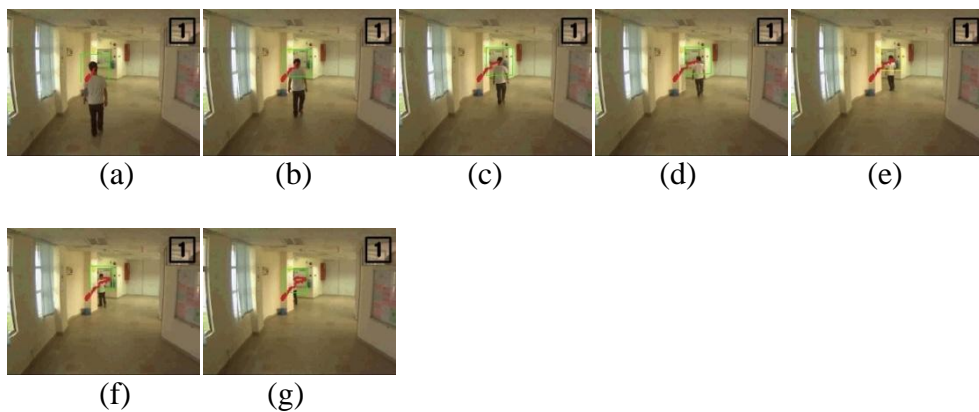


Figure 5.18 Screen shots of system execution which each block was formed

The result shown in figure 5.17 was good, nearly all of the motion directions calculated were correct except for block numbered 1 and 3. Note that the subject was moving upwards (direction of 4) but the system interprets the information as the subject moving to the direction of 5. This was inevitable as only a single wide angle camera was used. The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{1 + 0 + 1 + 0 + 0 + 0 + 0}{7} = 0.2857 = 2.857 \times 10^{-1}$$

The P6 calculates the motion direction of the subject in test case 2 accurately with an average of 2.857×10^{-1} calculation error (8 directions in total and maximum of error = 4).

5.3.2.4 Evaluation of P7

The two evaluation mechanisms for P7 were calculated as following:

$$FPR = \frac{10}{99} = 0.101 = 1.01 \times 10^{-1}$$

$$FNR = \frac{21}{99} = 0.2121 = 2.121 \times 10^{-1}$$

Refer to figure 5.18, when the subject moves in the scene, the subject became smaller in size. This could cause P7 to make the incorrect decision when locating the head position. Furthermore, the subject was wearing white colour cloth which hardens the process of P7 to locate the head position as the subject has been misclassified as background. Therefore, P7 has high FNR

that was 21% of the time, the head location was undefined; and locating the incorrect head position 10% of the time in this test case.

5.3.2.5 Evaluation of P8

The head pose estimation for each input from P7 was illustrated in figure 5.19. The blue dots in the figure 5.19 represent the estimated head pose and the black line represents the trend of the estimated head pose. Notice that some of the head pose were blank that was no estimated head pose for a certain frame. This could due to the previous process i.e. P7 that did not successfully locating the head position of the tracked human and hence no input to P8 for estimation process. The figure 5.20 shows the ground truth of the head poses for comparison.

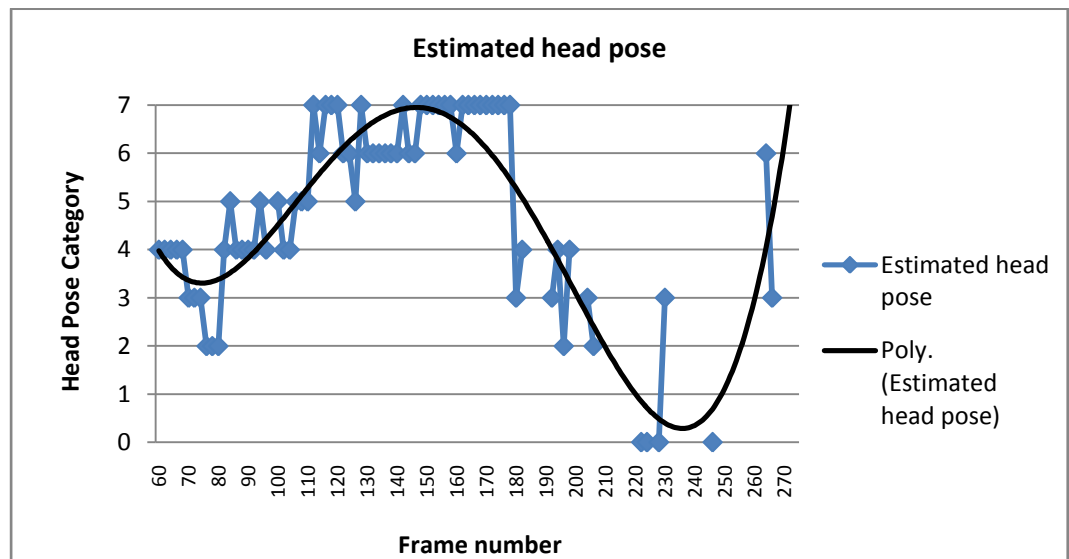


Figure 5.19 Estimated head pose for test case 2

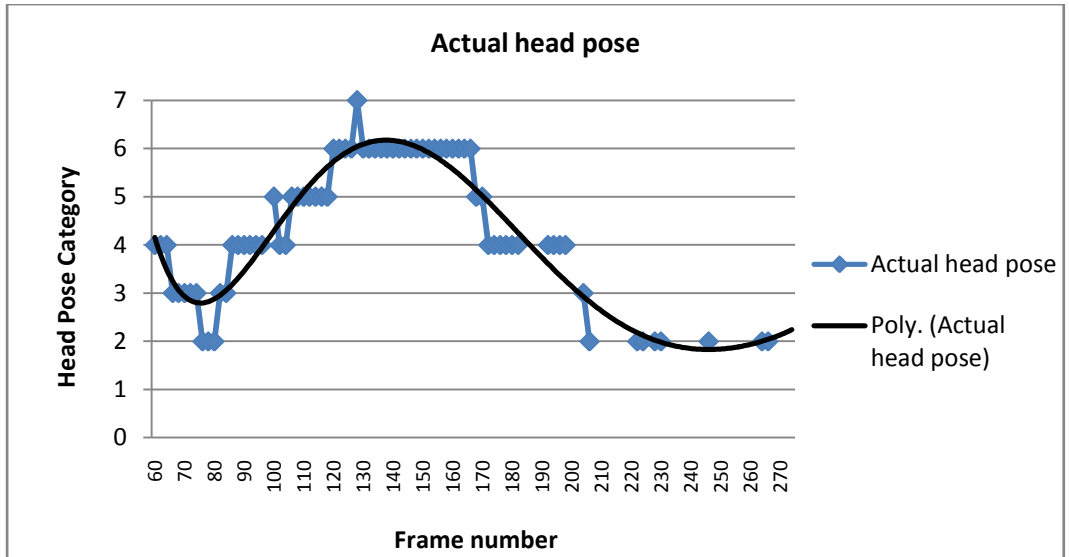


Figure 5.20 Actual head pose for test case 2

Notice that the estimated head poses were correct at the beginning stage and some deviation at the centre and later stage. Nevertheless, the overall pattern of the curves in figure 5.19 and 5.20 were considered similar.

The graph of error, E for P8 was shown in figure 5.21 which indicates the error in head pose estimation for each input from P7.

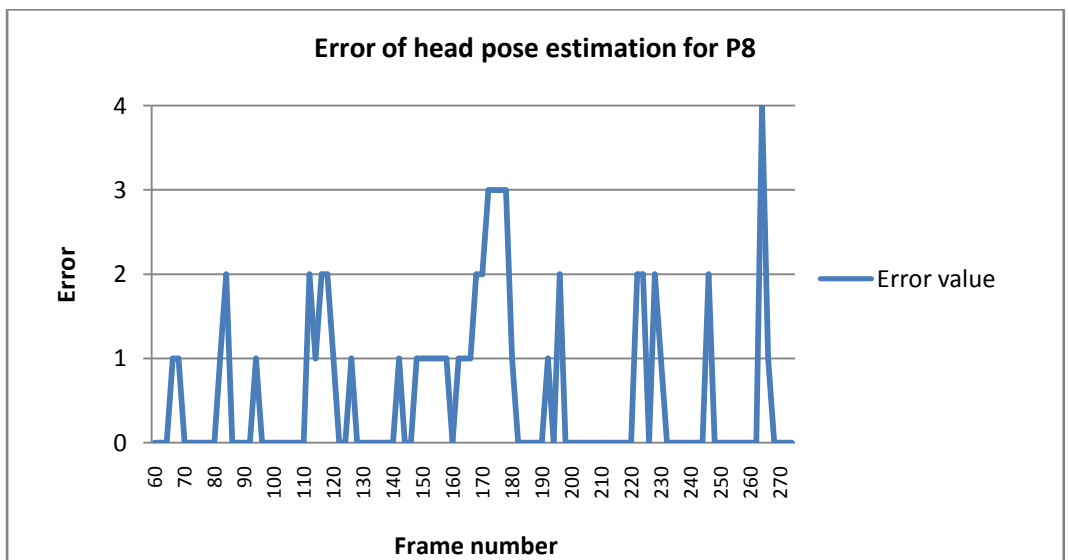


Figure 5.21 Error of head pose estimation for P8

The error values shown in figure 5.21 were calculated using formula 5.14 and 5.14. Here, it was clear that the head pose estimation result was not good as there were a lot of errors in the estimation result. This could due to the incorrect localization of head position by P7. From previous section (section 5.3.1.4 *Evaluation of P7*), there were 10% of incorrect localization of head position which ended up in incorrect estimation of head pose.

Notice that from frame numbered 100 to 180, the estimation of head pose tend to estimate one category higher than the actual head pose (i.e. estimated head pose was category of 6 but the actual head pose was category of 5; and the frame numbered 200 to 250, the estimation of head pose tend to estimate to lower category than the actual head pose. This could due to the scene colour that was about similar to the skin colour. The P8 mistakenly regards the background pixels as the pixels on the subject and hence the estimation results were bias towards neighbouring category of head pose which consist of more skin colour. This could cause the P8 to interpret the colour information wrongly. The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{59}{74} = 0.7973 = 7.973 \times 10^{-1}$$

Among 108 frames (the time period that the subject appears in the scene), P7 was only able to locate the head position of the subject on 74 frames. Hence, the aE was calculated with a divider of 74. For test case 2, only 37 out of the 74 (50%) estimation processes by P8 was correct, others were deviates from the ground truth result. This shows that the P8 algorithm

was performing poor in this scenario that was the subject wearing white colour which became similar to skin colour when shined by the natural sunlight. The P8 estimates the head pose of the subject in test case 2 accurately with an average of 7.973×10^{-1} estimation error (8 directions in total and maximum of error = 4).

5.3.2.6 Evaluation of P9

As described in section 5.3.1.5, the head poses were not always been successfully estimated and stored as inputs P9. This could due to the reason that P7 fails to locate the head position of the subject, or the estimation from P8 deviates from the actual head pose. This would affect the process of P9. Hence, the estimated head poses from P8 was being smoothed for ease of processing. The collection of head poses and the smoothed version of collection of head poses were illustrated in figure 5.19 and 5.22.

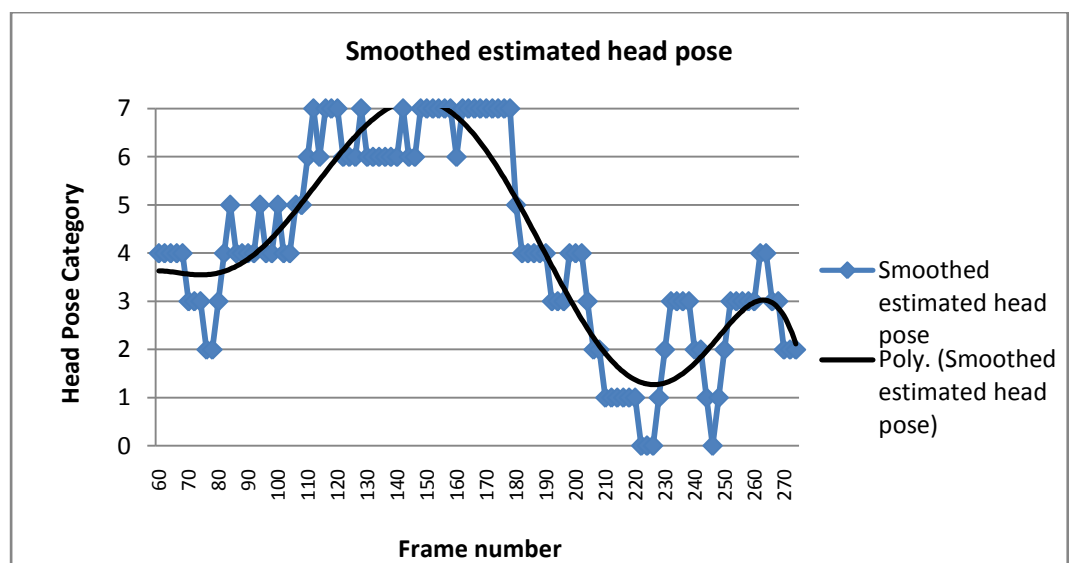
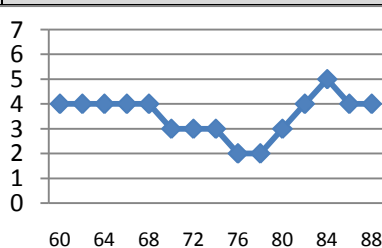
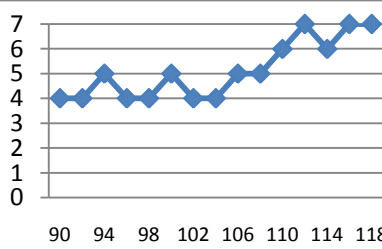


Figure 5.22 Smoothed estimated head poses

For every blocks of information, P9 detects the actions i.e. constant gaze on a particular location, looks around the surrounding area, or looks towards the direction of motion, of the tracked person. Therefore, to better analyse and discuss on the test case, the classification result from P9 throughout the whole testing video were arranged and summarized in table 5.7. The starting and ending frame for each block of information will be specified (that is 30 frames for each block) with the classification result. A short discussion on the result obtained will be included in the table 5.7 as well. The column *Gaze Direction* is calculated from the smoothed sequence of head pose and the column *Movement Direction* were obtained from P6. There were only 3 options to be filled into the column *Result from P9* and *Ground Truth* which were LS (looks straight towards a direction other than the direction of movement), LA (looks around the surrounding), and LD (looks towards the direction of movement).

Table 5.7 Summary of result from P9 for test case 2

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
1	60/89		4	5	LD	LD
2	90/119		-	5	LA	LA

To be continued...

...continued

Table 5.7 Summary of Result from P9

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
3	120/149	<p>120 124 128 132 136 140 144 148</p>	6	6	LD	LS
4	150/179	<p>150 154 158 162 166 170 174 178</p>	7	-	LS	LS
5	180/209	<p>180 184 188 192 196 200 204 208</p>	4	-	LA	LD
6	210/239	<p>210 214 218 222 226 230 234 238</p>	1	-	LA	LD
7	240/269	<p>240 244 248 252 256 260 264 268</p>	3	-	LA	LD

From table 5.7, only 3 out of 7 the decisions made by P9 were correct.

From the beginning, notice that the subject moves towards the direction of 5 and turns his head to the left on block numbered 1 and then turns his head to

the right on block numbered 2. P9 was able to detect that turning of head and identifies the action that looks around the surrounding successfully.

On the third block, the action should be looks straight towards a direction, however the P9 identifies that as looks towards the movement direction. This error was caused by the incorrect classification of movement direction by P7. The subject was moving towards the direction of 4, but the P7 classifies that as 6. Thus, the gaze and movement direction ended up the same that was 6 which leads the P9 to make the decision of looks towards the movement direction.

The block numbered 4 was a special case. The P7 misclassified the movement direction as no direction, but the subject was moving towards the direction of 4 (due to the camera angle, the subject seems remain static). Therefore, the actual result should be looks straight towards the direction 7 that was not the movement direction of 4. By coincidence, the decision made by P9 on block numbered 4 correctly from the incorrect information from previous processes.

As for block numbered 5, 6, and 7, the results from P9 were incorrect due to the incorrect estimation of head poses by P8. The head pose should be in category of 2 after frame numbered 200 (as illustrated in figure 5.20). The incorrect estimation of head poses caused P9 to misclassify the action as looks around the surrounding area. Furthermore, the subject was moving towards the direction of 4 on block numbered 5 and 2 on block numbered 6 and 7. The

incorrect result from P7 was also part of the reason that caused the incorrect result from P9.

5.3.3 Evaluation Using Test Case 3

Some frames from the testing video for test case 3 were illustrated in figure 5.23. This test case consists of a person (wearing dark colour cloth with long pant) that walks into the scene at frame numbered 108 and looks to the left and right on the way to the other exit of the scene. Later on, he exits the scene on frame numbered 492. Some screen shots of the execution of proposed system on test case 3 were shown in figure 5.24.



Figure 5.23 Frames from testing video of test case 3



Figure 5.24 Screen shots of program execution on testing video of test case

3

Using the evaluation mechanisms defined in section 5.3, the evaluation was conducted for processes of the proposed system based on the information in table 5.3.

5.3.3.1 Evaluation of P4

The three evaluation mechanisms for P4 were calculated as following:

$$FPPW = \frac{0}{326,793} = 0$$

$$FNPW = \frac{216}{326,793} = 0.0007 = 7 \times 10^{-4}$$

$$detection\ rate = \frac{190}{208} = 0.9134 = 9.134 \times 10^{-1}$$

P4 scores almost perfectly in this test case. With zero *FPPW*, there was no non-human object being classified as human in this test case in this test case. In addition, the *FNPW* was nearly zero as well. With a successive of over 300k times of human classification process, only 216 of them were falsely classified, that was classifies a human object as non-human object. Furthermore, the detection rate was over 90%.

The computational speed of P4 was illustrated in figure 5.25.

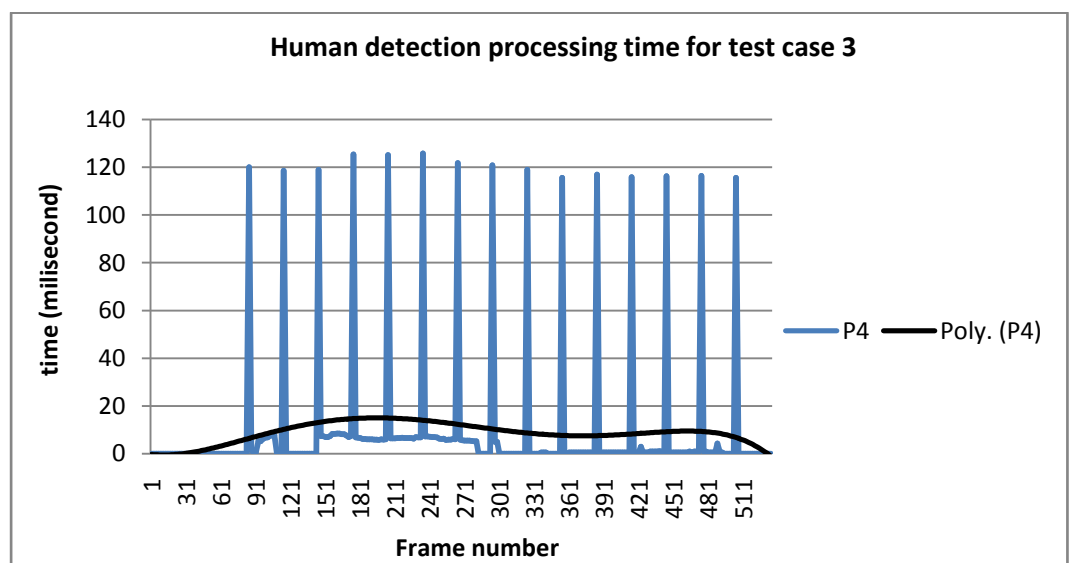


Figure 5.25 Human detection processing time for test case 3

The blue line in figure 5.25 represents the computational time for the proposed human detection process, P4 with a trend line that was in black colour.

At the beginning of the video, notice that from frame numbered 1 to 83, the human detection processing time for P4 was zero. This is due to the insufficient motion pixels (no motions in the scene) to trigger the process of P4. Most of the computational speed of P4 was less than 20 milliseconds per frame. Notice that there were spikes in the graph that was more than 120 milliseconds of computational time per frame. This was due to the periodic human classification process that invoked periodically of one second time interval. The average of computational speed of P4 for test case 2 was 8.651 milliseconds per frame.

5.3.3.2 Evaluation of P5

As for evaluation of human tracking algorithm, the *ETWPF* over time graph for test case 3 was shown in figure 5.26.

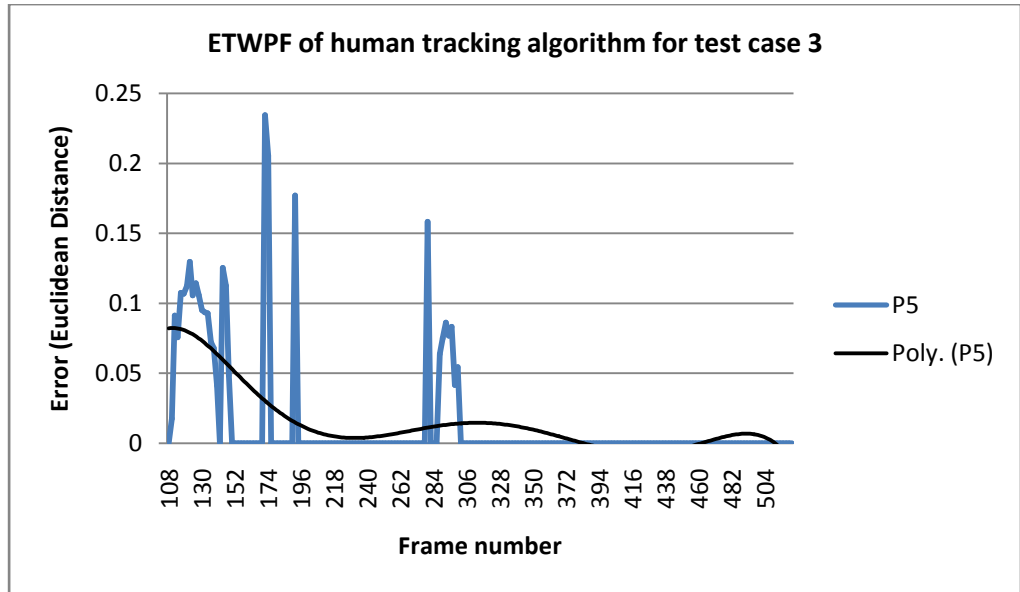


Figure 5.26 *ETWPF* evaluation over time graph for human tracking algorithm on test case 3

The blue line in figure 5.26 represents the error (Euclidean distance between object of interest and tracked position) in tracking by the proposed human tracking process, P5 with a trend line that was in black colour.

The average of *ETWPF* value of P5 for test case 3 was calculated as following:

$$aETWPF_{P5} = \frac{2.9731}{208} = 0.0143 = 1.43 \times 10^{-2}$$

Refer to the *aETWPF* values calculated, the average of errors for P5 was around 0.015 which indicates that the tracker was able to track the object of interest with a margin of error of about 1.5% deviation from the actual target.

Secondly, to test whether the tracker always have the information of head and shoulder, the $aFPR$ and $aTPR$ were calculated for P5.

$$aFPR_{P5} = FPR_{P5} = \frac{16}{208} = 0.0769 = 7.69 \times 10^{-2}$$

$$aTPR_{P5} = TPR_{P5} = 1 - 7.69 \times 10^{-2} = 9.231 \times 10^{-1}$$

From the calculation above, P5 was able to contain the head and shoulder of the target for over 92% of the time. Summary of evaluation of P5 was shown in table 5.8.

Table 5.8 Summary of evaluation of human tracking algorithm for test case 3

Evaluation Mechanism	Values of evaluation of P5
$aETWPF$	1.43×10^{-2}
$aFPR$	7.69×10^{-2}
$aTPR$	9.231×10^{-1}

5.3.3.3 Evaluation of P6

The graph of error, E for P6 was shown in figure 5.27 which indicates the error in motion direction calculation for each blocks of information.

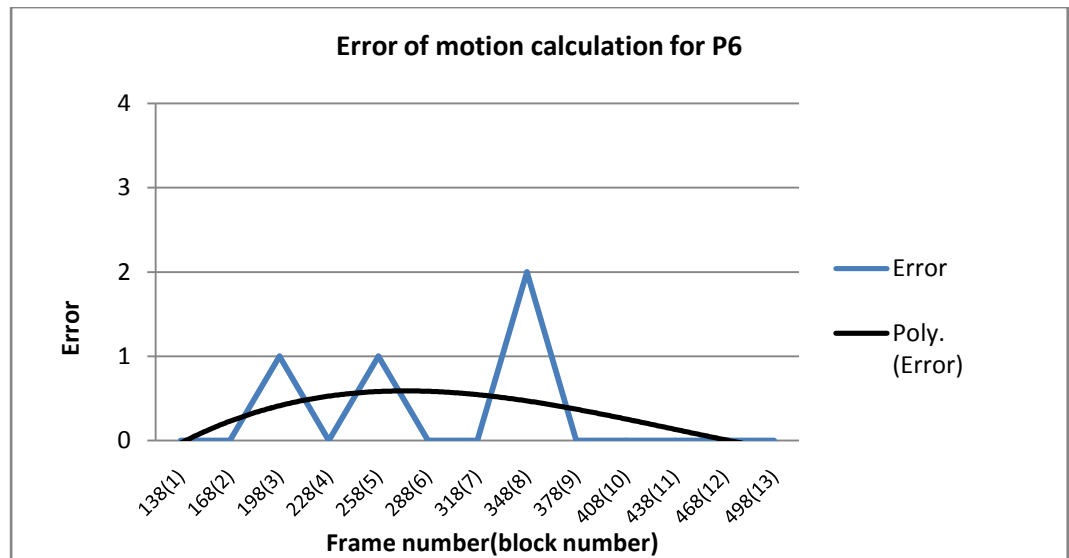


Figure 5.27 Error of motion calculation for P6

From the figure 5.27, the blue line indicates the error of motion direction calculation from P6 over time. The trend of the error over time was drawn as the black line in the same figure. The subject enters the scene on frame numbered 108, hence the first block of information was formed on the 138th frame ($108 + 30 = 138$, where the 30 was the frames per second of the testing video). There were 13 blocks collected for test case 3. The screen shots of the system execution while each block was formed were illustrated in figure 5.28.

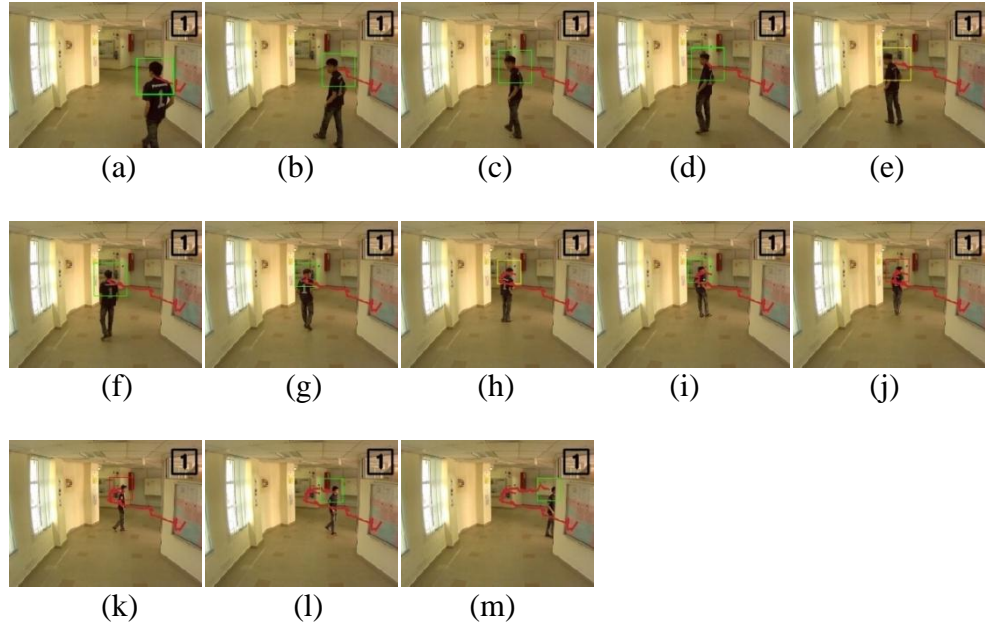


Figure 5.28 Screen shots of system execution which each block was formed

The result shown in figure 5.27 was good, nearly all of the motion directions calculated were correct except for block numbered 8. Note that from figure 5.28 (g) and (h), the subject was moving upwards (direction of 4), however movement of the subject was not moving upwards but moving upwards with his body leaning to the left and right. Hence, the system misclassified the movement direction of the subject on block numbered 8. The average of error, aE throughout the whole testing video was calculated as following:

$$\begin{aligned}
 aE &= \frac{0 + 0 + 1 + 0 + 1 + 0 + 0 + 2 + 0 + 0 + 0 + 0 + 0}{13} = 0.3077 \\
 &= 3.077 \times 10^{-1}
 \end{aligned}$$

The P6 calculates the motion direction of the subject in test case 3 accurately with an average of 3.077×10^{-1} calculation error (8 directions in total and maximum of error = 4).

5.3.3.4 Evaluation of P7

The two evaluation mechanisms for P7 were calculated as following:

$$FPR = \frac{21}{208} = 0.101 = 1.01 \times 10^{-1}$$

$$FNR = \frac{12}{208} = 0.0577 = 5.77 \times 10^{-2}$$

This test case shows that the P7 was performing well in localizing the head position of the subject in the scene. From the calculation of the evaluation mechanisms, P7 only misclassifies the head position 10% of the time and could not find the head position 6% of the time. The good evaluation result from this test case might due to the reason that the subject in the scene was wearing dark colour cloth which was highly differentiable from the background pixels. Hence, the previous process P5 can track the subject accurately and ease the process of P7.

5.3.3.5 Evaluation of P8

The head pose estimation for each input from P7 was illustrated in figure 5.29. The blue dots in the figure 5.29 represent the estimated head pose and the black line represents the trend of the estimated head pose. Notice that some of the head pose were blank that was no estimated head pose for a certain frame. This could due to the previous process i.e. P7 that did not successfully locating the head position of the tracked human and hence no

input to P8 for estimation process. The figure 5.30 shows the ground truth of the head poses for comparison.

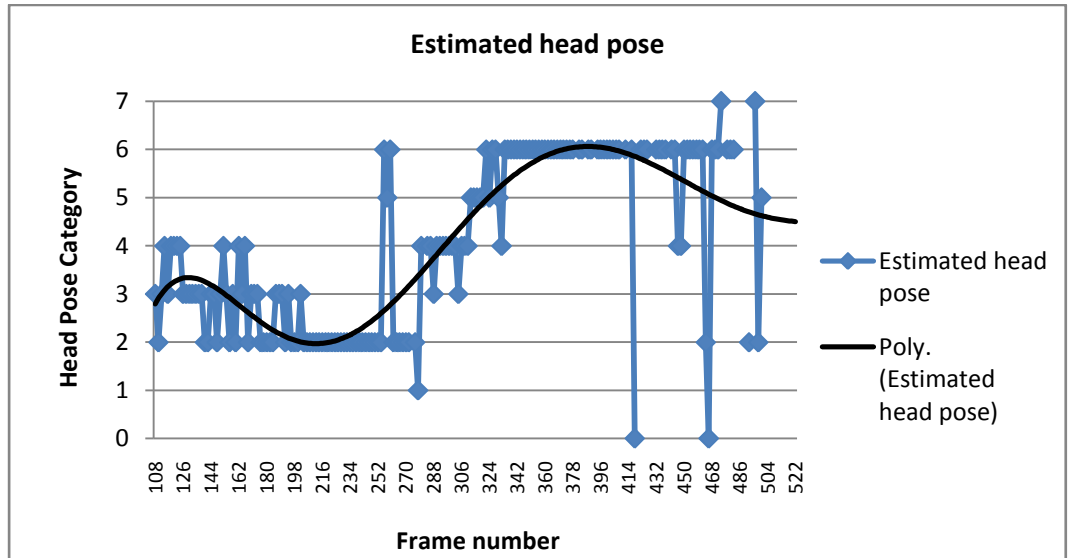


Figure 5.29 Estimated head pose for test case 3

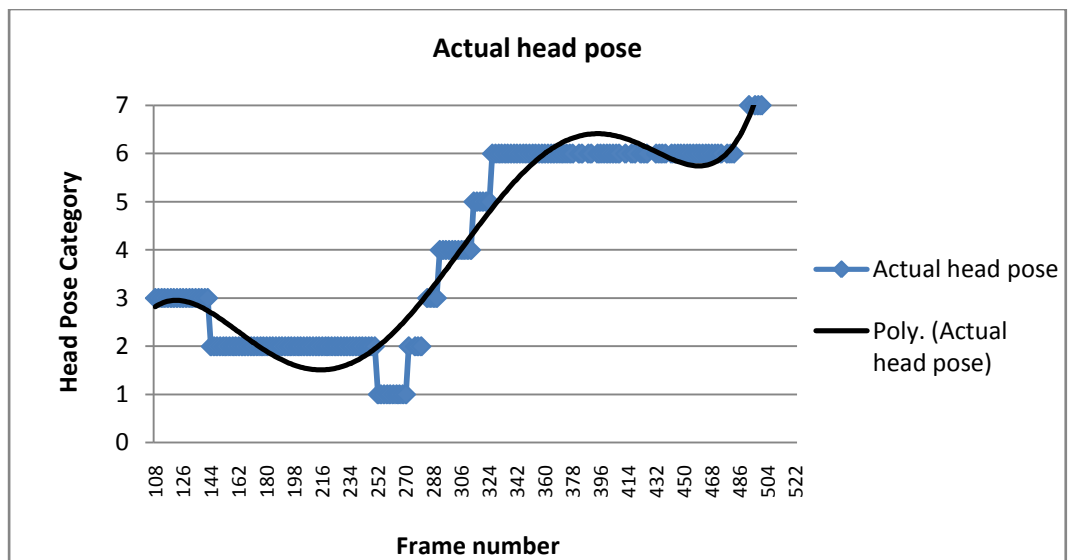


Figure 5.30 Actual head pose for test case 3

From the overview of both figure 5.29 and 5.30, the head pose curve was about similar at the middle part of the curve. The starting part and the part

near the end of the estimated head pose curve were deviated from the actual head pose curve. Notice that the some estimated head poses were correct that was the same as the actual head poses and some of them were only deviated one category of head pose away from the actual head poses i.e. estimated head pose was category 2 and actual head pose was category 3. As stated in chapter 4, section 4.9.4.2 *Grouping of Head Pose*, the neighbouring head poses was actually interrelated. Hence, the evaluation result of P8 was still acceptable since most of the deviated estimated head poses from the actual head poses were only one category away from the actual category (as shown in figure 5.31).

The graph of error, E for P8 was shown in figure 5.31 which indicates the error in head pose estimation for each input from P7.

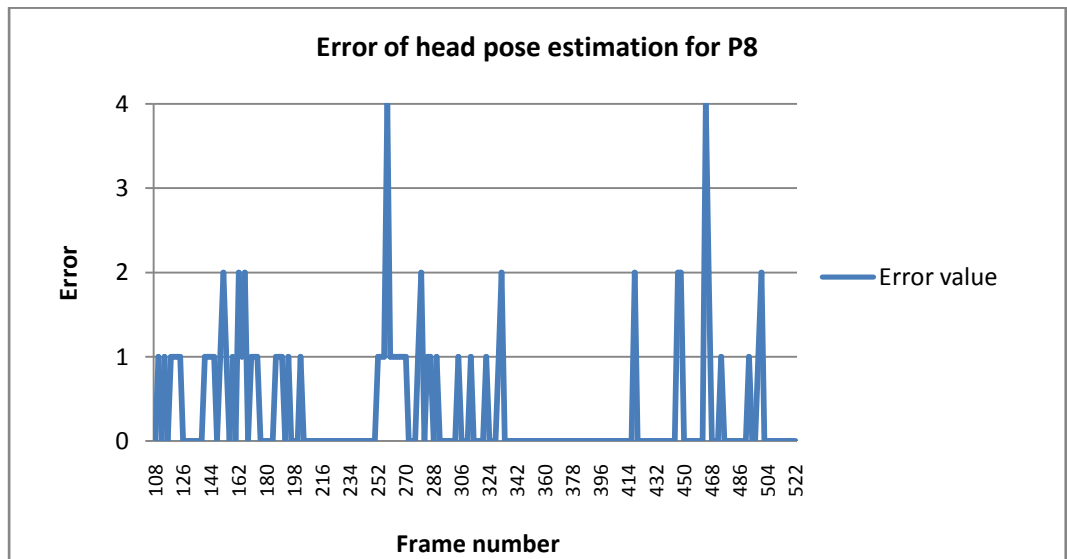


Figure 5.31 Error of head pose estimation for P8

The error values shown in figure 5.31 were calculated using formula 5.14 and 5.15. From the figure, there were two estimations with maximum

error value that were on frame numbered 258 and 464. Three possible reasons that responsible for the errors could be the previous process i.e. P7, the subject's size was too small when moves far from the camera and the background colour was similar to the skin colour and it might confused P8 in estimation.

The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{70}{182} = 0.3846 = 3.846 \times 10^{-1}$$

Among 208 frames (the time period that the subject appears in the scene), P7 was only able to locate the head position of the subject on 182 frames. Hence, the aE was calculated with a divider of 182. For test case 3, 126 out of the 182 (69%) estimation processes by P8 was correct, others were deviates from the ground truth result. This shows that the P8 algorithm was performing average in this scenario. The P8 estimates the head pose of the subject in test case 3 accurately with an average of 3.846×10^{-1} estimation error (8 directions in total and maximum of error = 4).

5.3.3.6 Evaluation of P9

As described in section 5.3.3.5, the head poses were not always been successfully estimated and stored as inputs P9. This could due to the reason that P7 fails to locate the head position of the subject, or the estimation from P8 deviates from the actual head pose. This would affect the process of P9.

Hence, the estimated head poses from P8 was being smoothed for ease of processing. The collection of head poses and the smoothed version of collection of head poses were illustrated in figure 5.29 and 5.32

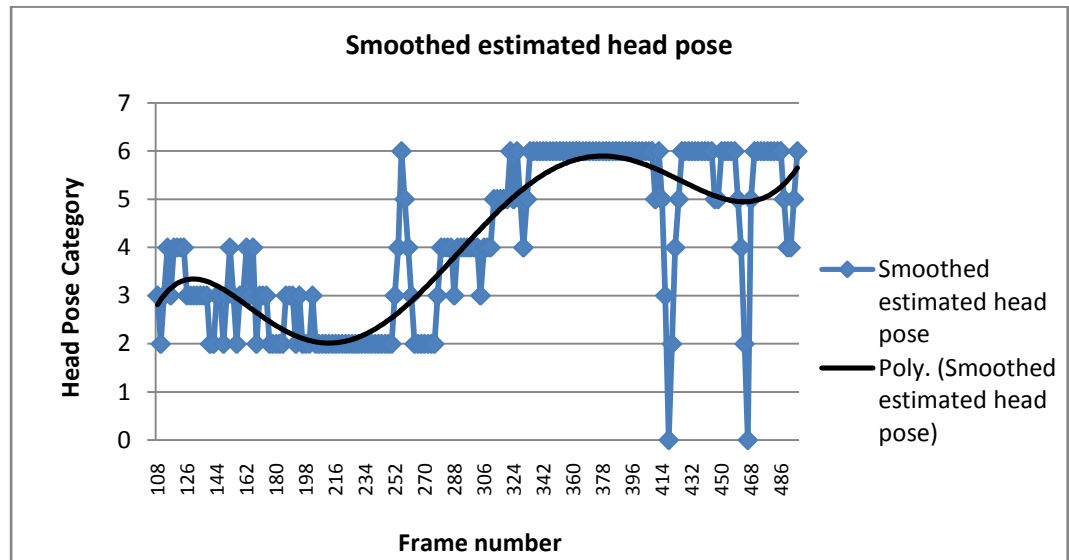


Figure 5.32 Smoothed estimated head poses

For every blocks of information, P9 detects the actions i.e. constant gaze on a particular location, looks around the surrounding area, or looks towards the direction of motion, of the tracked person. Therefore, to better analyse and discuss on the test case, the classification result from P9 throughout the whole testing video were arranged and summarized in table 5.9. The starting and ending frame for each block of information will be specified (that was 30 frames for each block) with the classification result. A short discussion on the result obtained will be included in the table 5.9 as well. The column *Gaze Direction* was calculated from the smoothed sequence of head pose and the column *Movement Direction* were obtained from P6. There were only 3 options to be filled into the column *Result from P9* and *Ground*

Truth which were LS (looks straight towards a direction other than the direction of movement), LA (looks around the surrounding), and LD (looks towards the direction of movement).

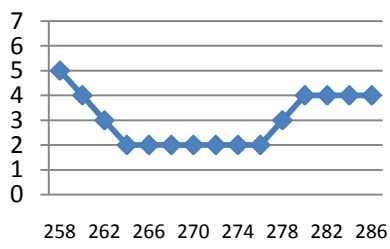
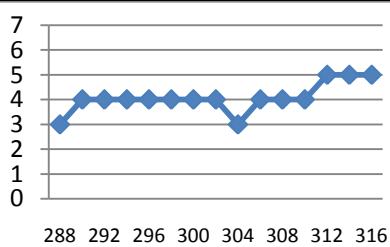
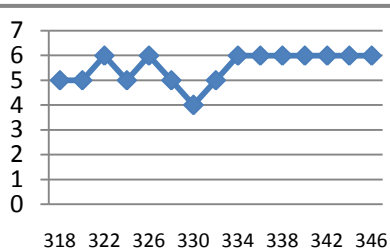
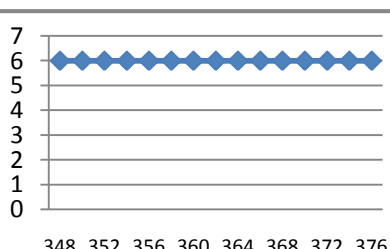
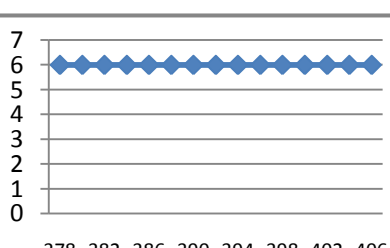
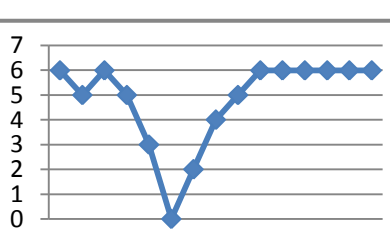
Table 5.9 Summary of result from P9 for test case 3

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
1	108/137	<p>108 112 116 120 124 128 132 136</p>	3	3	LD	LD
2	138/167	<p>138 142 146 150 154 158 162 166</p>	3	3	LD	LD
3	168/197	<p>168 172 176 180 184 188 192 196</p>	2	2	LD	LD
4	198/227	<p>198 202 206 210 214 218 222 226</p>	2	-	LS	LS
5	228/257	<p>228 232 236 240 244 248 252 256</p>	2	2	LD	LS

To be continued...

...continued

Table 5.9 Summary of Result from P9

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
6	258/287		2	-	LS	LS
7	288/317		4	-	LS	LD
8	318/347		6	6	LD	LA
9	348/377		6	3	LA	LA
10	378/407		6	-	LA	LS
11	408/437		6	-	LA	LD

To be continued...

...continued

Table 5.9 Summary of Result from P9

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth																		
12	438/467	<table border="1"> <caption>Data for Block 12 Head Pose</caption> <thead> <tr> <th>Frame No.</th> <th>Head Pose</th> </tr> </thead> <tbody> <tr><td>438</td><td>6</td></tr> <tr><td>442</td><td>6</td></tr> <tr><td>446</td><td>5</td></tr> <tr><td>450</td><td>5</td></tr> <tr><td>454</td><td>6</td></tr> <tr><td>458</td><td>6</td></tr> <tr><td>462</td><td>4</td></tr> <tr><td>466</td><td>0</td></tr> </tbody> </table>	Frame No.	Head Pose	438	6	442	6	446	5	450	5	454	6	458	6	462	4	466	0	6	6	LD	LD
Frame No.	Head Pose																							
438	6																							
442	6																							
446	5																							
450	5																							
454	6																							
458	6																							
462	4																							
466	0																							
13	468/497	<table border="1"> <caption>Data for Block 13 Head Pose</caption> <thead> <tr> <th>Frame No.</th> <th>Head Pose</th> </tr> </thead> <tbody> <tr><td>468</td><td>5</td></tr> <tr><td>472</td><td>6</td></tr> <tr><td>476</td><td>6</td></tr> <tr><td>480</td><td>6</td></tr> <tr><td>484</td><td>6</td></tr> <tr><td>488</td><td>5</td></tr> <tr><td>492</td><td>4</td></tr> <tr><td>496</td><td>7</td></tr> </tbody> </table>	Frame No.	Head Pose	468	5	472	6	476	6	480	6	484	6	488	5	492	4	496	7	6	5	LD	LD
Frame No.	Head Pose																							
468	5																							
472	6																							
476	6																							
480	6																							
484	6																							
488	5																							
492	4																							
496	7																							

From table 5.9, only 8 out of 13 decisions made by P9 were correct. The first and second blocks were straight forward, the subject enters the scene and moves towards the direction of 3 with the gaze direction of 3. Later on the third block, the subject continues to move towards the direction of 2 and remain static on fourth block while stare towards the direction of 2.

On the fifth block, P7 wrongly estimates the movement direction. The subject was still remaining static but P7 classifies it as moving towards direction of 2. This could due to the body movement of the subject. The subject's body was leant towards the direction of 2 while remains static. This cause the P7 to classify the leaning action as moving towards the direction of 2 and influence P9 in making the wrong decision.

On block numbered 6, the decision made by P9 was correct coincidentally. From the ground truth, the subject was moving towards the direction of 4 and stares on the direction of 2, hence the decision should be looks towards a direction other than his movement direction. However, the P7 was unable to accurately detect the movement of the subject when the subject was moving in direction of 4. Thus, the movement on block numbered 6 was classified as no movement. Since the gaze direction was 2, therefore the result from P9 was the same as the ground truth.

On block numbered 7, the result from P9 was different from the ground truth was due to the result from P7. The movement direction on block numbered 7 was 4 but P7 classifies it as no movement due to the similar reason as described on block numbered 5. It was similar for block numbered 8, result from P9 was incorrect. Notice that from block numbered 1 to 4, the subject turns his head to the left and turn to the right from block numbered 5 to 8. Hence, the ground truth result on frame numbered 8 was looks around the surrounding area. Nevertheless, P9 was able to uncover the looks around action done by the subject on block numbered 9. From block numbered 4 that was looks straight at direction 2, block numbered 7 that was looks straight at direction 4, and block numbered 9 that was looks towards the direction of 6. Focusing on 3 different directions showed enough evidence to identify the subject as looking around the surrounding area.

On blocks numbered 10 and 11, P9 takes that information as the continuous of the previous action i.e. looks around the surrounding since the

head poses were connected. However, they were not. On blocks numbered 10 and 11, the subject was moving towards the direction of 6 with the gaze direction of 6. Hence, the result should be looks towards the direction of movement. Even though there were some errors in monitoring process, P9 was able to correctly classify 8 out of 13 blocks of information as discussed previously on test case 3.

5.3.4 Evaluation Using Test Case 4

Some frames from the testing video for test case 4 were illustrated in figure 5.33. This test case consists of two persons (wearing purple colour cloth with short pant and wearing white colour cloth with short pant) that walk into the scene one after another at frame numbered 48 and 150; and exit at frame numbered 140 and 230. Both of the subjects look toward their movement directions and leave the scene. Some screen shots of the execution of proposed system on test case 4 were shown in figure 5.34.

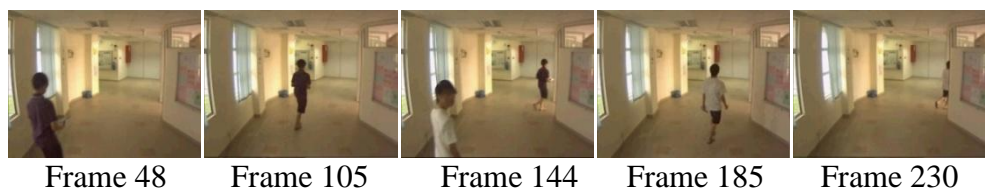


Figure 5.33 Frames from testing video of test case 4

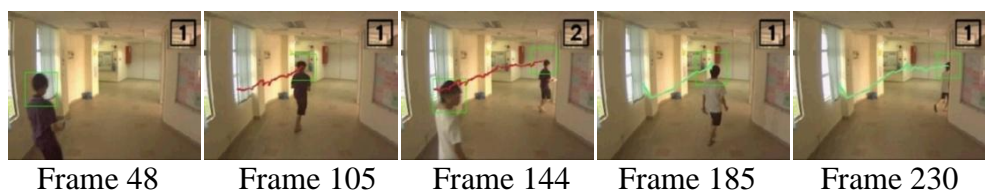


Figure 5.34 Screen shots of program execution on testing video of test case 4

Using the evaluation mechanisms defined in section 5.3, the evaluation was conducted for processes of the proposed system based on the information in table 5.3.

5.3.4.1 Evaluation of P4

The three evaluation mechanisms for P4 were calculated as following:

$$FPPW = \frac{0}{175,953} = 0$$

$$FNPW = \frac{300}{175,953} = 0.0017 = 1.7 \times 10^{-3}$$

$$detection\ rate = \frac{67}{92} = 0.7283 = 7.283 \times 10^{-1}$$

P4 scores well in this test case. With zero *FPPW*, there was no non-human object being classified as human in this test case. In addition, the *FNPW* was nearly zero as well. With a successive of over 170k times of human classification process, only 300 of them were falsely classified, that was classifies a human object as non-human object. In addition, the detection rate of P4 was over 70%.

The computational speed of P4 was illustrated in figure 5.35.

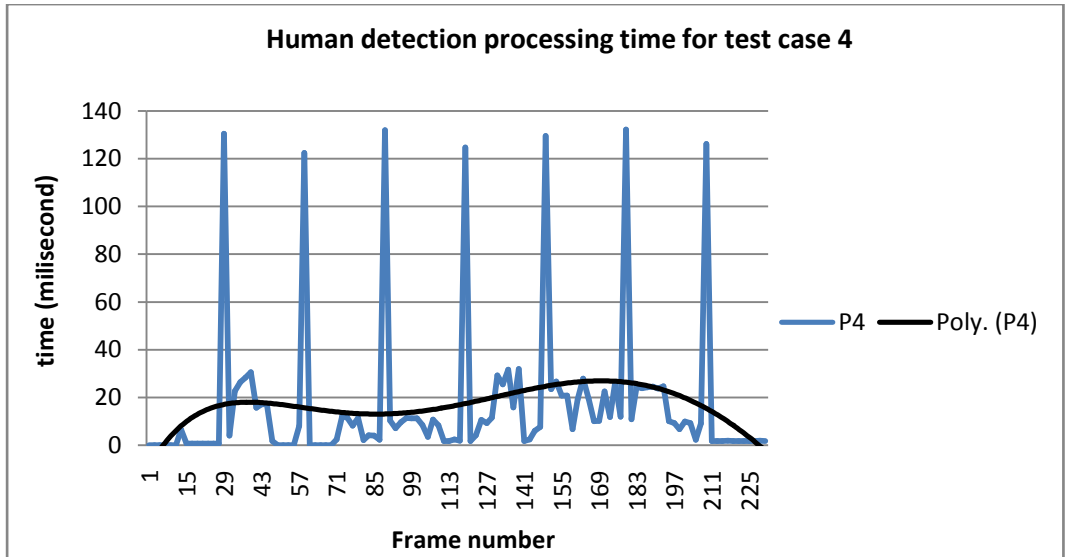


Figure 5.35 Human detection processing time for test case 4

The blue line in figure 5.35 represents the computational time for the proposed human detection process, P4 with a trend line that was in black colour.

Most of the computational speed of P4 was around 20 milliseconds per frame. Notice that there were spikes in the graph that was more than 120 milliseconds of computational time per frame. This was due to the periodic human classification process that invoked periodically of one second time interval. The average of computational speed of P4 for test case 2 was 16.51093 milliseconds per frame.

5.3.4.2 Evaluation of P5

As for evaluation of human tracking algorithm, the *ETWPF* over time graph for subject A and B of test case 4 were shown in figure 5.36 and 5.37 respectively.

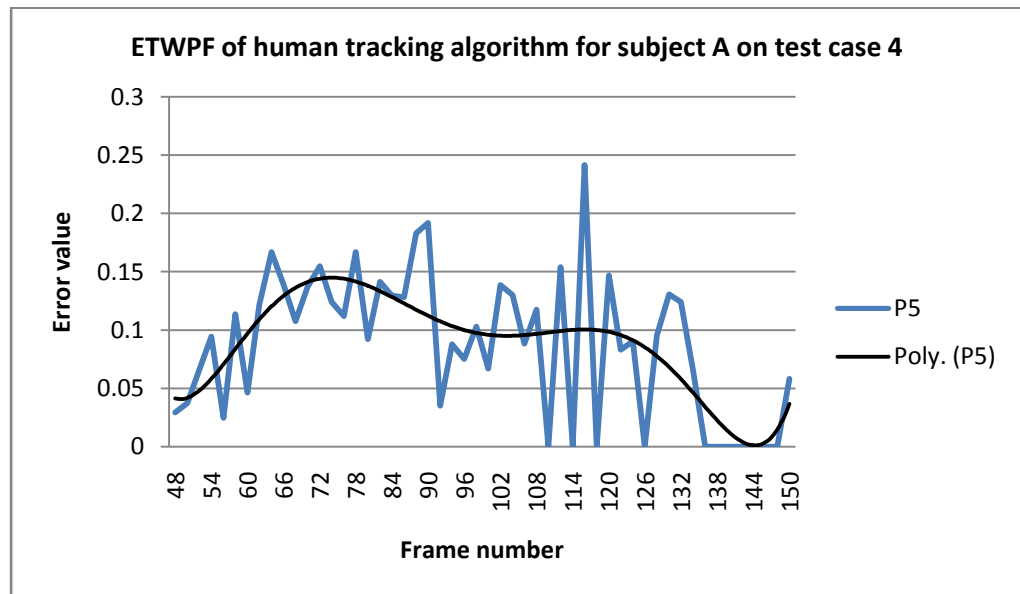


Figure 5.36 *ETWPF* evaluation over time graph for human tracking algorithm for subject A on test case 4

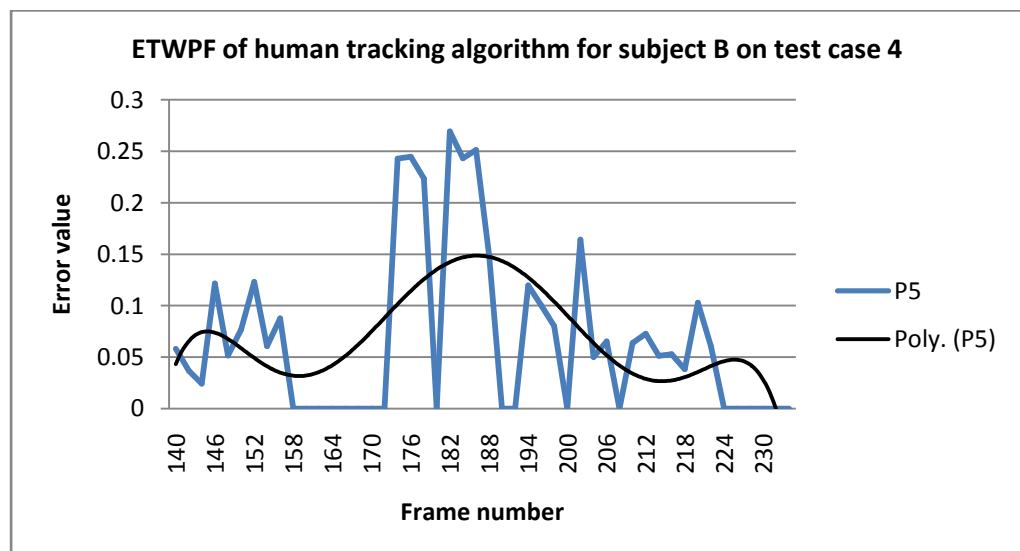


Figure 5.37 *ETWPF* evaluation over time graph for human tracking algorithm for subject B on test case 4

The blue line in figure 5.36 and 5.37 represents the error (Euclidean distance between object of interest and tracked position) in tracking of subject A and B by the proposed human tracking process, P5 with a trend line that was in black colour.

The average of $ETWPF$ value of P5 for test case 4 was calculated as following:

$$aETWPF_{Subject A,P5} = \frac{4.5438}{52} = 0.0874 = 8.74 \times 10^{-2}$$

$$aETWPF_{Subject B,P5} = \frac{3.2862}{45} = 0.073 = 7.3 \times 10^{-2}$$

$$aETWPF_{P5} = \frac{0.0874 + 0.073}{2} = 0.0802 = 8.02 \times 10^{-2}$$

Refer to the $aETWPF$ values calculated, the average of errors for P5 was around 8.02×10^{-2} which indicates that the tracker was able to track the object of interest with a margin of error of about 8% deviation from the actual target.

Secondly, to test whether the tracker always have the information of head and shoulder, the $aFPR$ and $aTPR$ were calculated for P5.

$$FPR_{SubjectA ,P5} = \frac{2}{52} = 0.0385 = 3.85 \times 10^{-2}$$

$$FPR_{SubjectB ,P5} = \frac{7}{45} = 0.1556 = 1.556 \times 10^{-1}$$

$$TPR_{SubjectA ,P5} = 1 - 3.85 \times 10^{-2} = 0.9615 = 9.615 \times 10^{-1}$$

$$TPR_{SubjectB ,P5} = 1 - 1.556 \times 10^{-1} = 0.8444 = 8.444 \times 10^{-1}$$

$$aFPR_{P5} = \frac{3.85 \times 10^{-2} + 1.556 \times 10^{-1}}{2} = 0.0971 = 9.71 \times 10^{-2}$$

$$aTPR_{P5} = \frac{9.615 \times 10^{-1} + 8.444 \times 10^{-1}}{2} = 0.903 = 9.03 \times 10^{-1}$$

P5 was able to track the object of interest and contain the head and shoulder part of the tracked human over 90% of the time. Summary of evaluation of P5 was shown in table 5.10.

Table 5. 10 Summary of evaluation of human tracking algorithm for test case 4

Evaluation Mechanism	Algorithm with Enhancement (P5)
<i>aETWPF</i>	8.02×10^{-2}
<i>aFPR</i>	9.71×10^{-2}
<i>aTPR</i>	9.03×10^{-1}

5.3.4.3 Evaluation of P6

The graph of error, *E* for P6 was shown in figure 5.38 which indicates the error in motion direction calculation for each blocks of information.

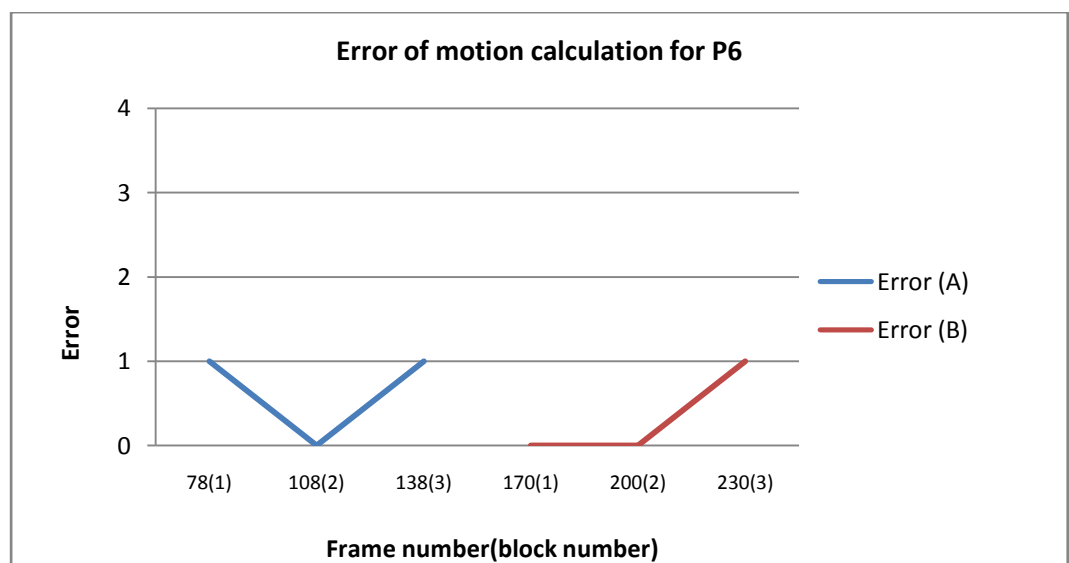


Figure 5.38 Error of motion calculation for P6

From the figure 5.38, the blue and red lines indicate the error of motion direction calculation for subject A and B from P6 over time. The subject A enters the scene on frame numbered 48, hence the first block of information was formed on the 78th frame ($48 + 30 = 78$, where the 30 was the frames per second of the testing video); the subject B enters the scene on frame numbered 140, and the first block of information was formed on the 170th frame ($140 + 30 = 170$). There were 6 blocks (3 blocks for each subject) collected for test case 4. The screen shots of the system execution while each block was formed were illustrated in figure 5.39.

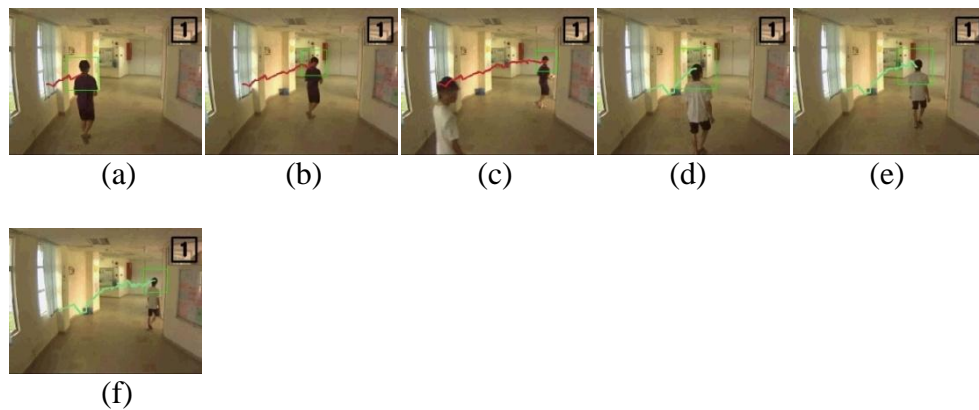


Figure 5.39 Screen shots of system execution which each block was formed

The result shown in figure 5.38 was good, nearly all of the motion directions calculated were correct or with low deviation from the actual result. The average of error, aE throughout the whole testing video was calculated as following:

$$aE = \frac{1 + 0 + 1 + 0 + 0 + 1}{6} = 0.5 = 5 \times 10^{-1}$$

The P6 calculates the motion direction of the subject in test case 4 accurately with an average of 5×10^{-1} calculation error (8 directions in total and maximum of error = 4).

5.3.4.4 Evaluation of P7

The two evaluation mechanisms for P7 were calculated as following:

$$FPR = \frac{3}{97} = 0.0309 = 3.09 \times 10^{-2}$$

$$FNR = \frac{14}{97} = 0.1443 = 1.443 \times 10^{-1}$$

Even though there was a subject who wears white colour cloth which was similar to the background colour when the sunlight shines on it, only 3% of the process of P7 was incorrect in this test case including both subjects. This shows the effectiveness of P7 on localizing the head position of the subject in the scene. Despite the low *FPR*, P7 was unable to find the head position of the subject 14% of the time.

5.3.4.5 Evaluation of P8

The head pose estimation for each input from P7 was illustrated in figure 5.40 (for subject A) and 5.41 (for subject B). The blue dots in the figure 5.40 and 5.41 represent the estimated head pose and the black line represents the trend of the estimated head pose. Notice that some of the head pose were blank that was no estimated head pose for a certain frame. This could due to

the previous process i.e. P7 that did not successfully locating the head position of the tracked human and hence no input to P8 for estimation process. The figure 5.42 and 5.43 show the ground truth of the head poses for subject A and B for comparison.

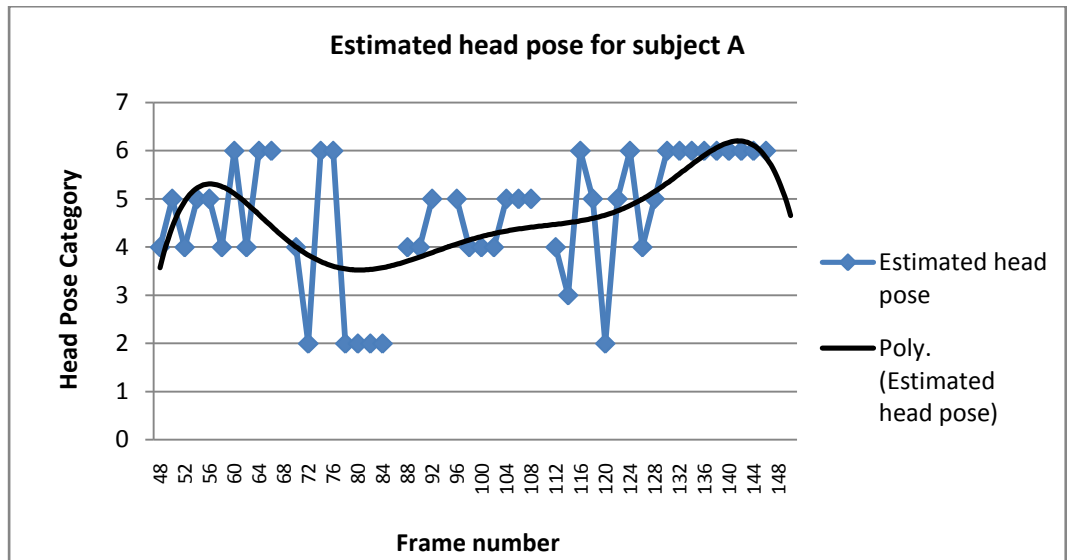


Figure 5.40 Estimated head pose for subject A for test case 4

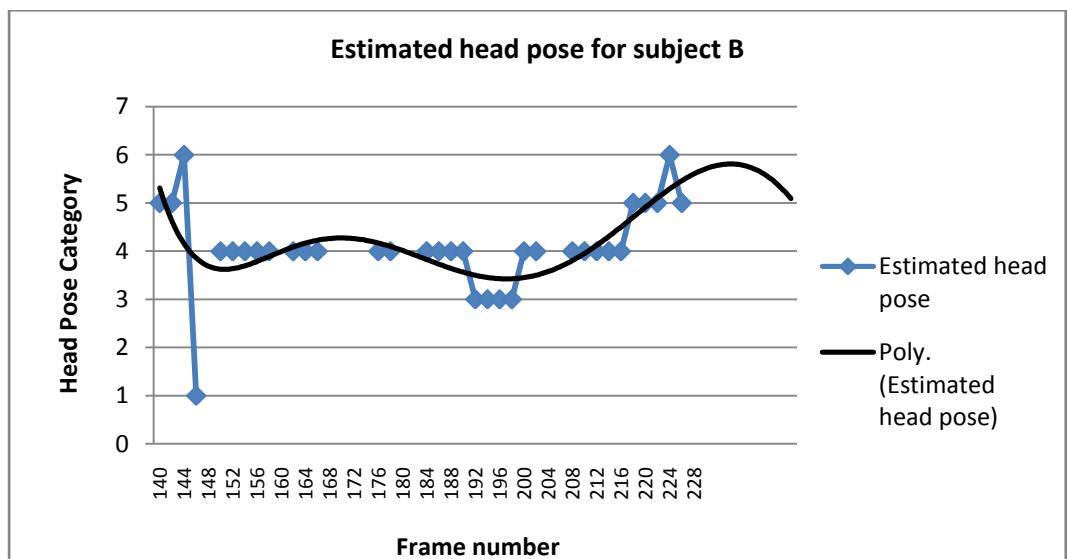


Figure 5.41 Estimated head pose for subject B for test case 4

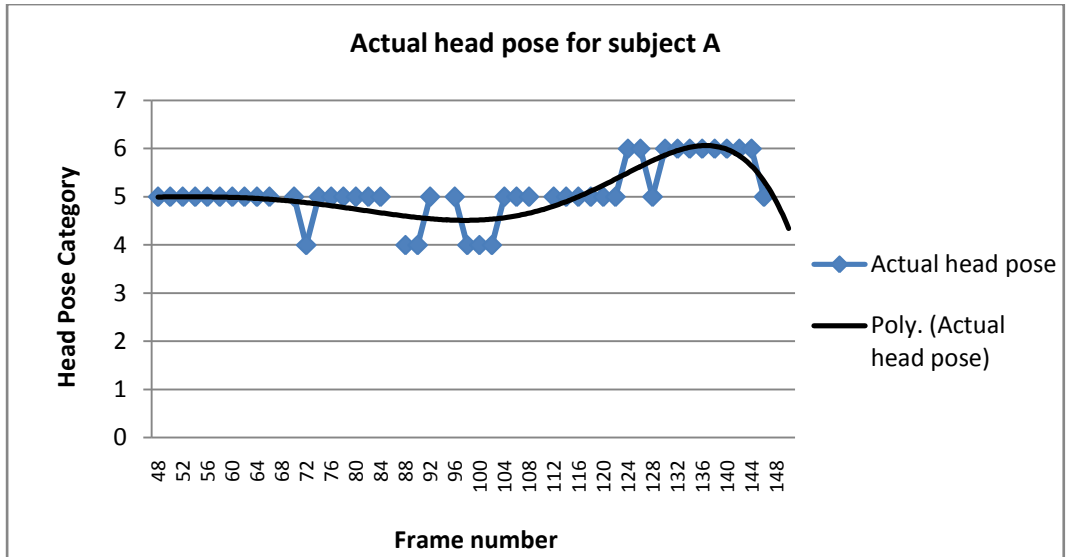


Figure 5.42 Actual head pose for subject A for test case 4

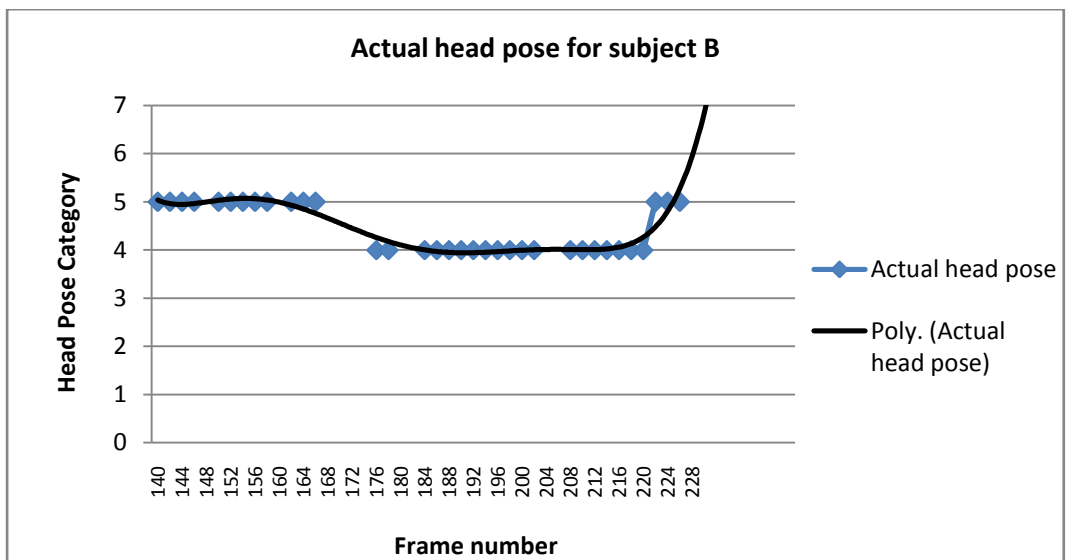


Figure 5.43 Actual head pose for subject B for test case 4

Refer to the figure 5.40 and 5.42, notice that the estimated results were quite different from the actual results. Only the middle part and the part near the end of the estimated head pose curve were similar to the actual head pose curve. Similar to the test case 4, some of the estimated head poses at the beginning part of the estimated head poses were deviated one category away

from the actual head poses, it was still acceptable since the neighbouring category of head pose were interrelated, only slight differences between the neighbouring category of head pose.

Notice that the head pose estimation between frame numbered 66 to 84 and 108 to 126 deviate far from the actual head pose. This could due to the two reasons. Since the P7 accurately localized the head position of the subjects (refer to section 5.3.4.4 Evaluation of P7), the problem would not be from P7. Therefore, one of the reasons was the insufficient brightness of the scene. The skin colour detection algorithm was unable to extract the skin region effectively and results in inaccurate estimation of head pose. Secondly, when the skin colour detection algorithm was unable to effectively retrieve the features, P8 was unable to use those extracted feature for head pose estimation.

As for subject B, notice that it was having the similar trend compared to subject A that was at the beginning part of the curve, the estimated head poses tend to get closer to the category 4; and the estimated head poses near the end of the curve tend to get to higher category of head pose as shown in figure 5.41 and 5.43. For subject B, P8 was able to accurately estimate the head poses but some of the estimated head poses having slight deviation that was one category away from the actual head poses. Combine the results from subject A and B, it was clear that when the subjects first enter the scene (beginning part of the curve), the estimated head poses tend to get closer to the head pose category 4 (which was back of the head, skin percentage would be

zero); and when the subjects near the exit, the estimation result became better compared to the result at the beginning part of the curve. This was the evidence that shows the insufficient brightness of the scene at the place where the subjects enter the scene. All the colour of the pixels became darker which would affect the P8 in estimating the head pose. P8 would mistakenly regards the darker skin pixels as “dark colour” (hair colour) and produce the incorrect estimation result.

The graph of error, E for P8 was shown in figure 5.44 and 5.45 which indicate the errors in head pose estimation for subject A and B for each input from P7.

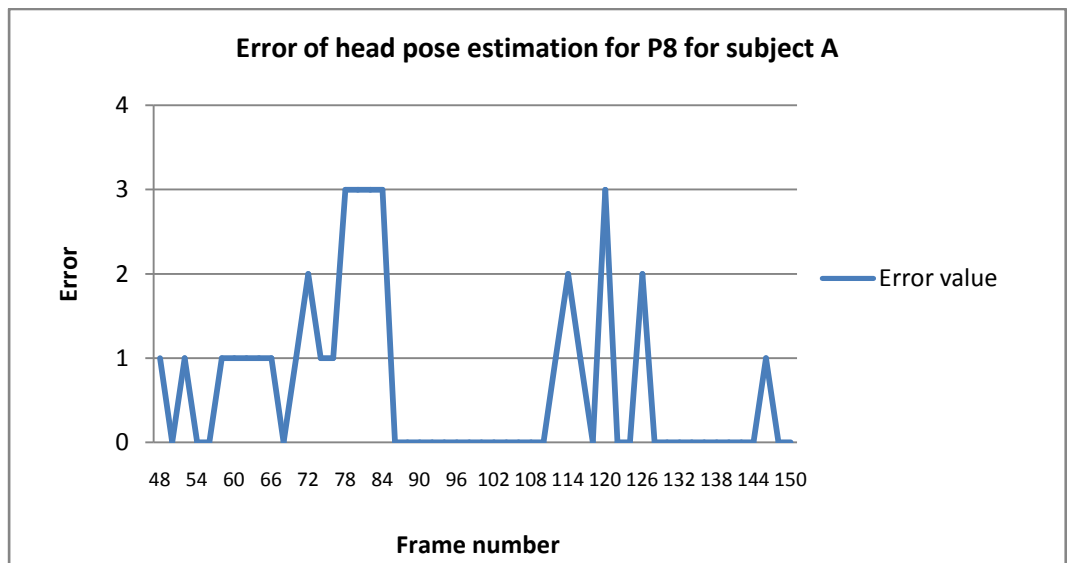


Figure 5.44 Error of head pose estimation for P8 for subject A

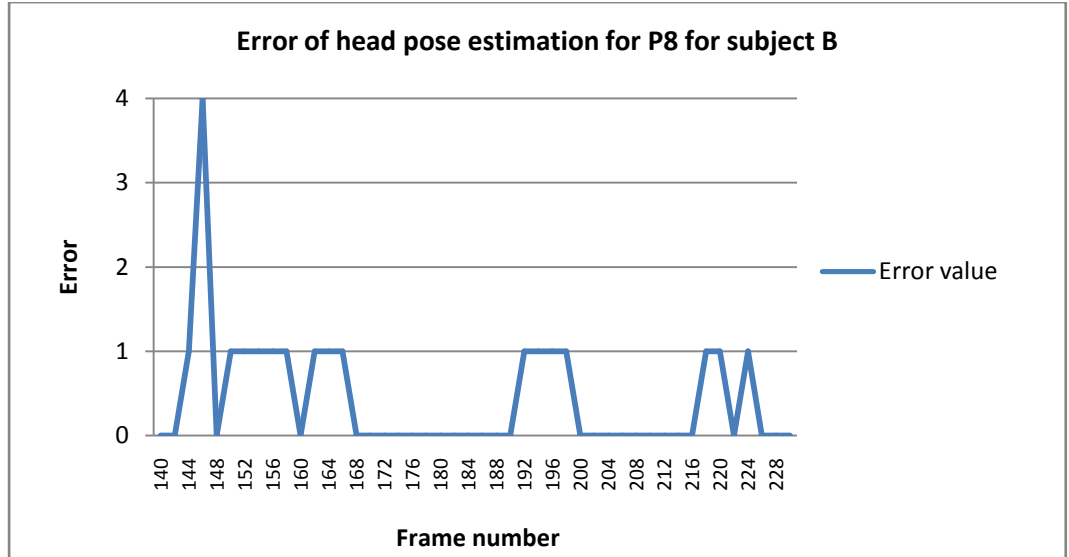


Figure 5.45 Error of head pose estimation for P8 for subject B

The error values shown in figure 5.44 and 5.45 were calculated using formula 5.14 and 5.15. From the figures, it was clear that the estimation result of subject B was better than that of subject A. The hair style of subject could also affect the estimation result in some way. The hair style of subject A was blocking part of the face, hence reducing the skin colour of the face. This affects the estimation process and produced more errors.

The average of error, aE throughout the whole testing video was calculated as following:

$$aE_{subject A} = \frac{34}{46} = 0.7391 = 7.391 \times 10^{-1}$$

$$aE_{subject A} = \frac{20}{34} = 0.5882 = 5.882 \times 10^{-1}$$

$$aE = \frac{7.391 \times 10^{-1} + 5.882 \times 10^{-1}}{2} = 0.6637 = 6.637 \times 10^{-1}$$

Among 52 and 45 frames (the time period that the subject A and B appear in the scene), P7 was only able to locate the head position of the subject on 46 and 34 frames. Hence, the aE for subject A and B were calculated with a divider of 46 and 34 respectively. For test case 4, 42 out of the 80 (52.5%) estimation processes by P8 was correct, others were deviates from the ground truth result. This shows that the P8 algorithm was performing average in this scenario. The P8 estimates the head pose of the subject in test case 4 accurately with an average of 6.637×10^{-1} estimation error (8 directions in total and maximum of error = 4).

5.3.4.6 Evaluation of P9

As described in section 5.3.4.5, the head poses were not always been successfully estimated and stored as inputs P9. This could due to the reason that P7 fails to locate the head position of the subject, or the estimation from P8 deviates from the actual head pose. This would affect the process of P9. Hence, the estimated head poses from P8 was being smoothed for ease of processing. The collection of head poses (figure 5.40 and 5.41) and the smoothed version of collection of head poses (figure 5.46 and 5.47) were illustrated in the following.

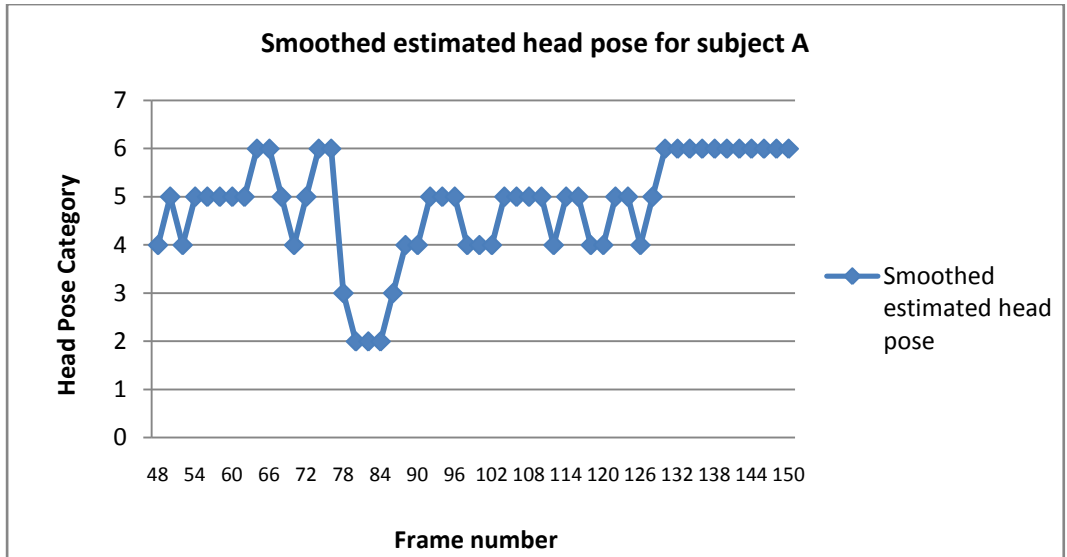


Figure 5.46 Smoothed estimated head poses for subject A

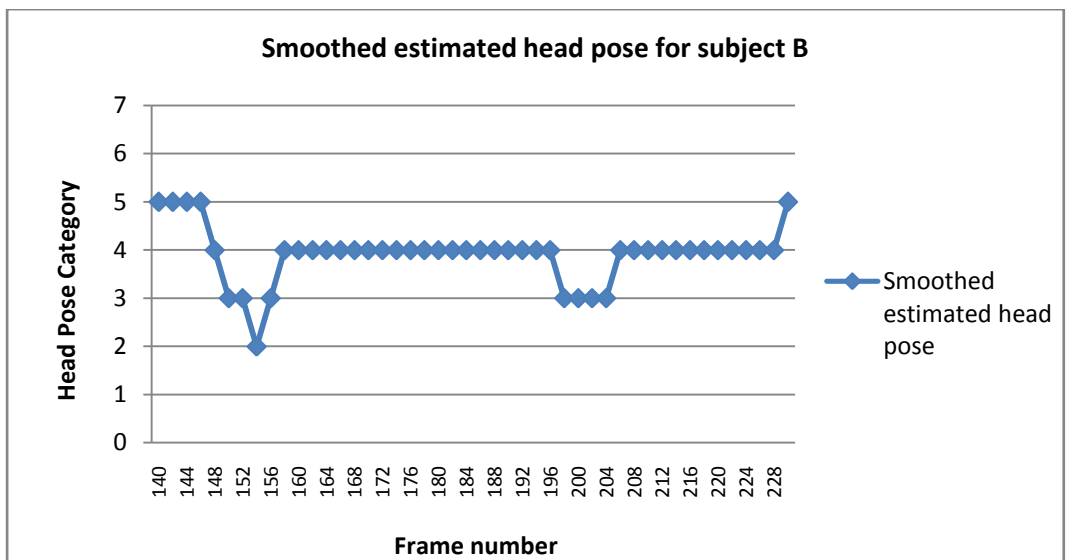


Figure 5.47 Smoothed estimated head poses for subject B

For every blocks of information, P9 detects the actions i.e. constant gaze on a particular location, looks around the surrounding area, or looks towards the direction of motion, of the tracked person. Therefore, to better analyse and discuss on the test case, the classification result from P9 throughout the whole testing video were arranged and summarized in table

5.11. The starting and ending frame for each block of information will be specified (that was 30 frames for each block) with the classification result. A short discussion on the result obtained will be included in the table 5.11 as well. The column *Gaze Direction* was calculated from the smoothed sequence of head pose and the column *Movement Direction* were obtained from P6. There were only 3 options to be filled into the column *Result from P9* and *Ground Truth* which were LS (looks straight towards a direction other than the direction of movement), LA (looks around the surrounding), and LD (looks towards the direction of movement). The *Block No.* for subject A and B were denoted as a number with alphabet A and B respectively.

Table 5.11 Summary of result from P9 for test case 4

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth																																
1A	48/77	<table border="1"> <caption>Data for Block 1A Head Pose</caption> <thead> <tr> <th>Frame No.</th> <th>Head Pose</th> </tr> </thead> <tbody> <tr><td>48</td><td>4.0</td></tr> <tr><td>50</td><td>5.0</td></tr> <tr><td>52</td><td>4.0</td></tr> <tr><td>54</td><td>5.0</td></tr> <tr><td>56</td><td>5.0</td></tr> <tr><td>58</td><td>5.0</td></tr> <tr><td>60</td><td>5.0</td></tr> <tr><td>62</td><td>5.0</td></tr> <tr><td>64</td><td>6.0</td></tr> <tr><td>66</td><td>6.0</td></tr> <tr><td>68</td><td>5.0</td></tr> <tr><td>70</td><td>4.0</td></tr> <tr><td>72</td><td>5.0</td></tr> <tr><td>74</td><td>6.0</td></tr> <tr><td>76</td><td>6.0</td></tr> </tbody> </table>	Frame No.	Head Pose	48	4.0	50	5.0	52	4.0	54	5.0	56	5.0	58	5.0	60	5.0	62	5.0	64	6.0	66	6.0	68	5.0	70	4.0	72	5.0	74	6.0	76	6.0	5	4	LD	LD
Frame No.	Head Pose																																					
48	4.0																																					
50	5.0																																					
52	4.0																																					
54	5.0																																					
56	5.0																																					
58	5.0																																					
60	5.0																																					
62	5.0																																					
64	6.0																																					
66	6.0																																					
68	5.0																																					
70	4.0																																					
72	5.0																																					
74	6.0																																					
76	6.0																																					
2A	78/107	<table border="1"> <caption>Data for Block 2A Head Pose</caption> <thead> <tr> <th>Frame No.</th> <th>Head Pose</th> </tr> </thead> <tbody> <tr><td>78</td><td>3.0</td></tr> <tr><td>80</td><td>2.0</td></tr> <tr><td>82</td><td>2.0</td></tr> <tr><td>84</td><td>2.0</td></tr> <tr><td>86</td><td>3.0</td></tr> <tr><td>88</td><td>4.0</td></tr> <tr><td>90</td><td>4.0</td></tr> <tr><td>92</td><td>5.0</td></tr> <tr><td>94</td><td>5.0</td></tr> <tr><td>96</td><td>5.0</td></tr> <tr><td>98</td><td>4.0</td></tr> <tr><td>100</td><td>4.0</td></tr> <tr><td>102</td><td>4.0</td></tr> <tr><td>104</td><td>5.0</td></tr> <tr><td>106</td><td>5.0</td></tr> </tbody> </table>	Frame No.	Head Pose	78	3.0	80	2.0	82	2.0	84	2.0	86	3.0	88	4.0	90	4.0	92	5.0	94	5.0	96	5.0	98	4.0	100	4.0	102	4.0	104	5.0	106	5.0	-	5	LA	LD
Frame No.	Head Pose																																					
78	3.0																																					
80	2.0																																					
82	2.0																																					
84	2.0																																					
86	3.0																																					
88	4.0																																					
90	4.0																																					
92	5.0																																					
94	5.0																																					
96	5.0																																					
98	4.0																																					
100	4.0																																					
102	4.0																																					
104	5.0																																					
106	5.0																																					
3A	108/137	<table border="1"> <caption>Data for Block 3A Head Pose</caption> <thead> <tr> <th>Frame No.</th> <th>Head Pose</th> </tr> </thead> <tbody> <tr><td>108</td><td>5.0</td></tr> <tr><td>110</td><td>5.0</td></tr> <tr><td>112</td><td>4.0</td></tr> <tr><td>114</td><td>5.0</td></tr> <tr><td>116</td><td>5.0</td></tr> <tr><td>118</td><td>4.0</td></tr> <tr><td>120</td><td>4.0</td></tr> <tr><td>122</td><td>5.0</td></tr> <tr><td>124</td><td>5.0</td></tr> <tr><td>126</td><td>4.0</td></tr> <tr><td>128</td><td>5.0</td></tr> <tr><td>130</td><td>6.0</td></tr> <tr><td>132</td><td>6.0</td></tr> <tr><td>134</td><td>6.0</td></tr> <tr><td>136</td><td>6.0</td></tr> </tbody> </table>	Frame No.	Head Pose	108	5.0	110	5.0	112	4.0	114	5.0	116	5.0	118	4.0	120	4.0	122	5.0	124	5.0	126	4.0	128	5.0	130	6.0	132	6.0	134	6.0	136	6.0	5	5	LD	LD
Frame No.	Head Pose																																					
108	5.0																																					
110	5.0																																					
112	4.0																																					
114	5.0																																					
116	5.0																																					
118	4.0																																					
120	4.0																																					
122	5.0																																					
124	5.0																																					
126	4.0																																					
128	5.0																																					
130	6.0																																					
132	6.0																																					
134	6.0																																					
136	6.0																																					

To be continued...

...continued

Table 5.11 Summary of result from P9

Block No.	Frame No. (Start/End)	Smoothed Sequence of Head Pose	Gaze Direction	Motion Direction	Result from P9	Ground Truth
1B	140/169		4	5	LD	LD
2B	170/199		4	5	LD	LD
3B	200/229		4	6	LS	LS

From table 5.11, most of the decision made by P9 was correct other than the decision on block numbered 2A. In this test case, both subject A and B enter the scene and looks towards the direction of movement before exit the scene.

Notice that the block numbered 2A was classified as LA which was the only incorrect decision was due to the incorrect information from P8. From section 5.3.4.5 *Evaluation of P8*, the most of the head poses for block numbered 2A should be under category of 5. However, due to the incorrect estimation result, the combined information from blocks numbered 1A and 2A

became looks to the right side on block numbered 1A and then looks to the left on block numbered 2A. Thus, P9 would classify this information as looks around the surrounding area which was incorrect. Nevertheless, P9 still able to identify 5 out of 6 blocks of information correctly. In this test case, the proposed system performs well in such a way that most of the actions done by the testing subjects in the testing video were able to identify correctly.

5.3.5 Evaluation Using Test Case 5

Some frames from the testing video for test case 5 were illustrated in figure 5.48. This test case consists of three persons (wearing white colour cloth with long pant, wearing blue colour cloth with long pant and wearing white colour cloth with short pant) that walk into the scene one after another at frame numbered 30, 188 and 356; and exit at frame numbered 150, 280 and 430. All of the subjects look toward their movement directions and leave the scene. Some screen shots of the execution of proposed system on test case 5 were shown in figure 5.49.



Figure 5.48 Frames from testing video of test case 5



Frame 38 Frame 136 Frame 234 Frame 356 Frame 420

Figure 5.49 Screen shots of program execution on testing video of test case

5

Using the evaluation mechanisms defined in section 5.3, the evaluation was conducted for processes of the proposed system based on the information in table 5.3.

5.3.5.1 Evaluation of P4

The three evaluation mechanisms for P4 were calculated as following:

$$FPPW = \frac{0}{287,926} = 0$$

$$FNPW = \frac{1692}{287,926} = 0.0059 = 5.9 \times 10^{-3}$$

$$detection\ rate = \frac{2}{143} = 0.014 = 1.4 \times 10^{-2}$$

Notice that there was no false positive in this test case and the *FNPW* was low (0.6%). However, P4 was only able to detect all the subjects appear in the scene in 2 out of 143 frames. This result was unbearable as most of the subjects appear in the scene were undetected by P4. For the first subject who enters the scene was wearing a backpack and a white coloured shirt. The white coloured shirt makes the motion detection algorithm (P1) fails to obtain the motion blob effectively and the flawed subtracted motion hardens the human

detection process. In addition, the shape of the subject was slightly different from normal humans that were without the backpack. This further hardens the process of human detection since only normal humans without backpack were used to train the human classifier. Similarly, second and third subjects' cloth colour was making the motion detection algorithm fails to obtain the motion blob effectively and resulting in poor human detection.

Another reason for the low detection rate was due to the small object size when the subject was far from the camera. When the subject was very small in size, the gradients obtained (for HOG) might differ from the gradients from those training images for human detector. P4 works poorly in this test case.

Since both the algorithms were unable to detect humans appear in the scene effectively, accessing the speed of computation was superfluous. Thus, accessing the speed of computation was skipped in for this test case.

5.3.5.2 Evaluation of P5

As for evaluation of human tracking algorithm, only the *ETWPF* over time graph for subject B of test case 5 was shown in figure 5.50 since P4 was unable to detect the subject A and C.

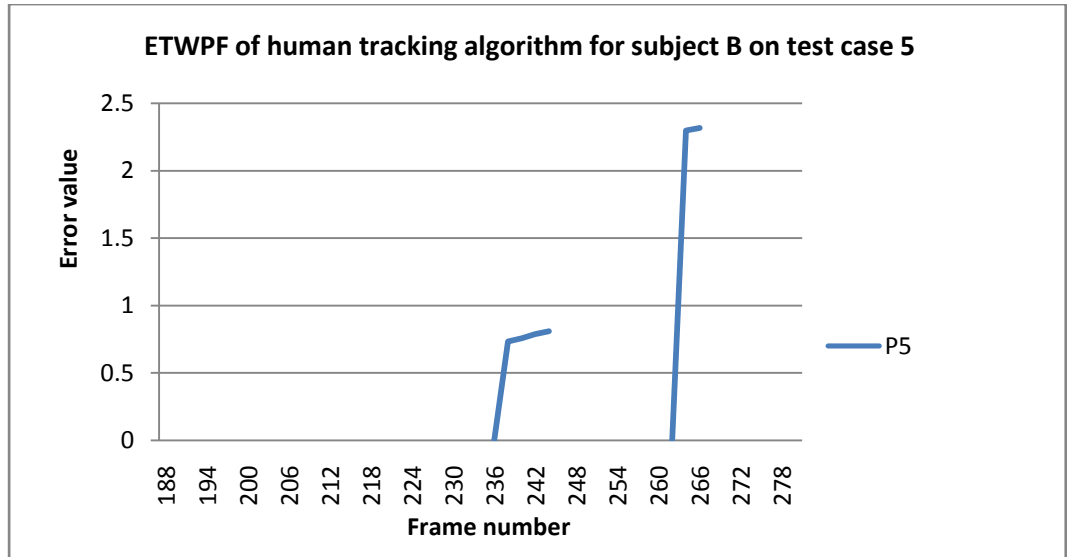


Figure 5.50 **ETWPF** evaluation over time graph for human tracking algorithm for subject B on test case 5

The blue line in figure 5.50 represents the error (Euclidean distance between object of interest and tracked position) in tracking of subject B by the proposed human tracking process.

From figure 5.50, most of the values of *ETWPF* for both P5 were deviated far from the subject B. Most of the errors were over the error value of 1 from the subject, which means that the tracker was totally out of the target object and considered lost of tracking (as shown in figure 5.50). The large error in tracking could be due to the previous processes i.e. P1 to P4 in which the subjects were undetected as a human that appear in the scene. Even though subject B was detected as human (in some frames), the tracking result was far from accurate.

When subject B was tracked, the motion blob of subject B was in a poorly defined state. Due to the lighting and the background colour, the motion blob of subject B was broken i.e. a few motion blobs that belong to the same subject. In this case, P5 was unable to effectively estimate the next position of the object of interest. Thus, resulting in poor tracking of object of interest.

The average of $ETWPF$ value of P5 for test case 5 was calculated as following:

$$aETWPF_{Subject B,P5} = \frac{7.7048}{8} = 0.9631 = 9.631 \times 10^{-1}$$

Refer to the $aETWPF$ values calculated, the average of errors (for subject B) for P5 was around 0.95, which is deviated far from the object of interest.

Secondly, to test whether the tracker always have the information of head and shoulder, the $aFPR$ and $aTPR$ were calculated for P5.

$$FPR_{SubjectA ,P5} = \frac{0}{0} = \text{undefined}$$

$$FPR_{SubjectB ,P5} = \frac{6}{8} = 0.75 = 7.5 \times 10^{-1}$$

$$FPR_{SubjectC ,P5} = \frac{0}{0} = \text{undefined}$$

$$TPR_{SubjectA ,P5} = \text{undefined}$$

$$TPR_{SubjectB ,P5} = 1 - 7.5 \times 10^{-1} = 2.5 \times 10^{-1}$$

$$TPR_{SubjectC ,P5} = \text{undefined}$$

$$aFPR_{P5} = \text{undefined}$$

$$aTPR_{P5} = \text{undefined}$$

Since subject A and C were undetected as human by P4, some of the evaluation mechanisms were undefined. Without a good prior information (from previous processes), the tracking algorithm could not perform well.

5.3.4.3 Evaluation of P6

Based on the tracking information in section 5.3.4.2 *Evaluation of P5*, there was no information for this process to be accessed.

5.3.5.4 Evaluation of P7

Similar to the section 5.3.4.3 *Evaluation of P6*, other than the moment when the tracker were first assigned, all the trackers did not consist any of the information required i.e. head and shoulder of the subject. Hence, P7 was unable to estimate the location of the head.

5.3.5.5 Evaluation of P8

Zero output from P7 means zero input for P8. There was nothing to assess in this test case.

5.3.5.6 Evaluation of P9

Since there was no block of information generated, therefore there was no input for P9. However, this does not mean that there was no error but the proposed system fails to retrieve the information from the testing video and process them. This test case was the worst case among all the testing videos.

5.3.6 Summary of Evaluation

As a summary, the evaluation on P4, P5, P6, P7, P8, and P9 will be concluded and arranged in table 5.12 for better comparison.

Table 5.122 Summary of evaluation of proposed system

Evaluation	Test Case 1 (BC)	Test Case 2 (NC)	Test Case 3 (NC)	Test Case 4 (NC)	Test Case 5 (WC)
P4					
<i>FPPW</i>	0	0	0	0	0
<i>FNPW</i>	2.988×10^{-3}	1.914×10^{-3}	7.0×10^{-4}	1.7×10^{-3}	5.9×10^{-3}
Detection rate	5.532×10^{-1}	7.196×10^{-1}	9.134×10^{-1}	7.283×10^{-1}	1.4×10^{-2}
P5					
<i>aETWPF</i>	4.52×10^{-2}	8.8×10^{-3}	1.43×10^{-2}	8.02×10^{-2}	<i>undefined</i>
<i>aFPR</i>	1.409×10^{-2}	7.48×10^{-2}	7.69×10^{-2}	9.71×10^{-2}	<i>undefined</i>
<i>aTPR</i>	9.859×10^{-1}	9.252×10^{-1}	9.231×10^{-1}	9.03×10^{-1}	<i>undefined</i>
P6					
<i>aE</i>	7.778×10^{-1}	2.857×10^{-1}	3.077×10^{-1}	5.0×10^{-1}	NA
P7					
<i>FPR</i>	7.092×10^{-3}	1.01×10^{-1}	1.01×10^{-1}	3.093×10^{-2}	NA
<i>FNR</i>	4.114×10^{-1}	2.121×10^{-1}	5.77×10^{-2}	1.443×10^{-1}	NA
P8					
<i>aE</i>	1.4×10^{-1}	7.973×10^{-1}	3.846×10^{-1}	6.637×10^{-1}	NA
P9					
Percentage Of Correct Decision	8.889×10^{-1}	4.286×10^{-1}	6.154×10^{-1}	8.333×10^{-1}	NA

Remarks:

* Not all of the subjects appear in the scene were included in the calculation

From table 5.12, test case 1 had the highest percentage of correct decision on P9. Note that the evaluation on P9 was actually the evaluation of the whole system since the input of P9 was the output from previous processes i.e. P1 to P8. As for test case 5, there was no single block of information generated due to insufficient information and hence there were no activities for process P6 to P9. There was nothing to assess.

Notice that the *FPPW* value for all test cases was 0 which means that the human classifier used in P4 was well trained that produced zero false positive result. However, the detection rate was just about average (except test case 3 that was over 90% detection rate). This might due to the high hit threshold of human detection. This was acceptable since it eliminates the false positive result from P4 which would affect the subsequent processes and wastes precious resources. P4 works better in brighter lighting condition of the scene. Notice that the result of P4 was the best in test case 3. This might due to the high brightness of the scene using florescence light bulb. Furthermore, the subject was wearing dark coloured shirt and dark coloured long pant which having high contrast between the subject's colour and the background colour. In this case, the motion detection algorithm in P1 could produce the high quality motion blob (motion blob that having the shapes of human that appears in the scene) for the subsequent processes i.e. P2 to P9 so that they could fully utilize the information of the scene.

The distance between the subject and the camera was another factor that affects the process of P4. It became harder for P4 to detect human when

the subject was small in size. This could be due to the insufficient information gathered from the small sized motion blob obtained. It was similar for P5. The result of P5 was influenced by three factors includes the cloth colour of the subject, the distance between the subject and the camera, and the lighting condition of the scene.

For test case 1 and 3, even though the subject was wearing dark coloured cloth and the background was bright, the $aETWPF$ value was still higher than that of test case 2 where the subject in test case 2 was wearing white colour cloth (that was similar to the background pixels' colour) and the brightness level of the scene was low. This was due to the action done by the subject in the scene. In test case 1, the subject bent down to collect the object on the floor. This action caused some part of the head and shoulder be occluded. The tracked points were unable to locate the tracked position and caused the deviation from the original position. As for test case 3, the subject was constantly turning his head to the left and right. In the meanwhile, the subject was leaning to the side (left and right) while moving. This hardens the prediction of possible next location of the subject and ends up in slight deviation from the target. Refer to the section 5.3.2 *Evaluation Using Test Case 2*, the subject was moving straight until the end of the scene (top left side of the scene). For the system, the subject seems like remains static in the scene. This makes the tracking easier since the object did not turn his body (change the shape of the body) and did not move in a curly pattern (only walks in a straight line away from the camera). Therefore, having a lower $aETWPF$ value.

Refer to the table 5.12, the evaluation of P6, the aE values for test case 1 and 4 were higher compared to test case 2 and 3. This was due to the camera angle and camera position. Only a camera was used in this research work that was a wide angle camera mounted high below the ceiling. Due to the wide angle property of the camera, the scene captured needs to be calibrated so that when the subject was either near or far from the camera, the size of the subject and the distance moved when subject was near or far from the camera can be normalized. It would be easier for the system to interpret the information of the scene.

Some of the errors in the evaluation of P6 were due to the insufficient class of direction (8 directions and 1 directionless). This happens when the subject was moving from the left to right or right to left. For instance, when the subject enters the scene from the left bottom part of the scene and exits the scene on the right middle part of the scene. If the subject was moving in a straight line, the system would perceive the direction of movement as somewhere between the direction of 4 and 5 or 5 and 6. In this case, it was hard for the system to decide which direction it was. Other than the reasons stated above, another possible cause of errors in evaluation of P6 might be due to the forming of block of information. In the proposed system, the block of information was formed when the system collects 15 frames (the testing video was 30 frames per second and only half of the frames were processed. It was fixed to one second duration for the block of information). The duration of each block of information might not be suitable as it was too short or too long.

It was fixed to one second for ease of computation. More investigation needed to define the best duration for each block of information.

For P7, notice that when the subject was moving far from the camera, P7 would produce more false positive results (test case 2 and 3). This was understandable since the pixels of the subject became less when the subject farther from the camera. With small number of pixels within the tracked position, it was hard to locate the head position. For instance, only 10 pixels to build up the head and shoulder part of the tracked person. It was nearly impossible to locate the head position using the algorithm of P7; As for test case 1 and 4, the subjects in test case 1 and 4 did not move far from the camera. However, test case 4 produced more false positive results compared to test case 1. This was due to the white colour shirt worn by one of the subject in test case 4. As the white colour shirt was similar to the background colour, the motion blob obtained for that subject might be broken into several pieces and caused the essential parts (head and shoulder of the subject) separates from being bounded by the tracker. Therefore, the system could not locate the position of the head due to the constraint (there must be the head and shoulder part in the tracked position so that there was enough information for P7 to locate the head position).

P7 was too dependent on the motion pixels as P7 locates the head position by calculating the number of motion pixels within the tracked position. It was not suitable to be implemented in crowded scene where all the motion pixels might be mixed up.

P8 was one of the cores of the system. For P8 to have a better result, a few factors were involved such as the brightness of the scene, the size of the head of the subject, sensitivity of the skin colour detector, and also the darkness of the subject's hair colour. From the table 5.12, some deduction can be made such as a brighter scene allows P8 to have a better estimation result; when subject moves far from the camera would reduce the accuracy of P8.

When the scene was low in brightness i.e. test case 2 and 4, the colour of the wall and the floor of the scene were somehow similar to the skin colour. Furthermore, the darkened scene also makes the subject's colour became darker. The skin colour of the head might be affected by shadow and become dark. In this case, the skin colour detector used might mistakenly regard the skin pixels as hair pixels since the brightness level was low. Even though the system was equipped with the colour balancing algorithm (refer to chapter 4, section 4.9.1 *Colour Normalization*) to balance the colour of the scene before the process of P8, when it comes to low brightness of the scene, the result from the colour balancing algorithm was still low in brightness but with balanced colour (since the whole scene was low in brightness, average of the scene was also low in brightness). Nevertheless, the colour balancing algorithm did enhance the performance of P8 when the scene colour was somehow yellowish (due to the evening sunlight). In this case, most of the background pixels (pixels that similar to the skin colour tone) were detected as skin pixels by the skin colour detector. The colour balancing algorithm could balance the colour of the scene to bring back the original colour of the scene. Hence, eliminates the background pixels from being grouped as skin pixels.

As for the second points stated in previous paragraph, when the size of the subject was small i.e. far from the camera, the size of the head of the subject was small as well. Imagine when a subject's head was made up by only 15 pixels, it was hard to divide the regions for feature extraction (refer to chapter 4, section 4.9.2 *Region Division*). Thus, results in poor head pose estimation result.

P9 monitors the subject's sequence of head pose for actions such as looks straight towards the direction of movement, constant focus on a direction other than his or her movement direction, and looks around the surrounding area. P9 was very vulnerable as any previous processes i.e. P1 to P8 perform poorly due to some conditions stated previously, the result of P9 would be affected. Other than the previous processes, the duration of block of information defined for monitoring (refer to chapter 4, section 4.10 *Head Pose Monitoring*) would be another factor that affect P9. As stated previously, the duration for the block of information was fixed as one second for ease of computation. It might not be suitable for real life monitoring process. Test case 5 in the table 5.12 illustrates well about P9. If any previous processes fails to function, P9 would produce incorrect result or unable to function.

5.4 Conclusion

This concludes the discussion about the testing of the proposed system. As discussed, a complete system was built up by a lot of tiny components. Any

components that unable to function well would affect the overall process of the system. Thus, the most essential part of the system was not at the end of the process chain, but at the beginning of the process chain i.e. P1. If P1 was unable to obtain the motion blobs effectively, all the subsequent processes would not be able to acquire the true information (the actual information of the scene), not to mention about the accuracy of the system.

The next chapter will conclude the research work by defining the future enhancements about this research work and contributions of this research work towards the research world.

CHAPTER 6

FUTURE WORKS AND CONCLUSION

6.1 Introduction

There are three main deliverables in this chapter includes the contributions, future works and conclusion of this research work. Future works refer to the improvement of the proposed system in terms of efficiency, robustness, speed of computation, as well as the accuracy of processes; whereas the conclusion provides a summary of the research work.

6.2 Contribution

This research work has introduced some contributions toward the society as well as the academics research industry. For the second module of this research work, “scene monitoring based on sequence of estimated head pose”, a new feature set that is suitable, simple, and small feature size for head pose estimation was proposed. Furthermore, a novel head pose estimation approach was introduced that fully utilize the defined feature set for better classification process. Such approach for head pose estimation is very sensitive to colour, it is suitable for estimation where the image is blurred or small. Finally, a simple yet useful approach for scene monitoring based on sequence of estimated head pose was introduced. Such monitoring approach is

light, which enables the saving of computer resources; having both global and local detection (detection based on latest information retrieved and the combination of all information retrieved) of head movements; and could be modified or installs additional rules (sequence of head pose that defines an action) for more head movements to be identified.

Last but not least, some components in the second module such as the following help in the preparation of data for better representation of information for scene monitoring:

1. Automatic head localization using hill climbing algorithm.
2. Dynamically smooth the estimated head pose over time curve for better representation of head motion information for analysis.
3. Incorporate colour balancing technique in feature extraction for head pose estimation process.

6.3 Future Work

As discussed in previous chapter, there are some limitations in the proposed system which prevents the deployment of such system as real life applications. To make the system works better and more feasible in tackling real life environment, it is necessary to further enhance the system to minimize those limitations.

First of all, a shadow remover process for the motion extraction process in P1 can be added which guarantees a better quality of extracted motion regions. By removing the shadow of objects in the scene, the detection space and search space for P4 and P5 will be reduced automatically. The detection rate of P4 and tracking efficiency of P5 can be increase and hence increase the overall accuracy of the system.

Secondly, the occlusion and grouping of humans are common in real life scenarios. For a system to be considered real life application, it must be able to handle the occlusion, as well as the grouping of humans appears in the scene. Thus, occlusion handling mechanism has to be added into the proposed system to enhance the robustness of such system.

Notice that the distance, angle and size of an object in different part of the scene are different due to the use of single wide angle camera as input. The captured scene signal has to be calibrated early in the process chains for a better analysis works in latter processes. The defined movement direction classes (in P6, 8 directions and 1 directionless) might not be enough to represent the real movement direction of human appears in the scene. A more complete direction class has to be defined.

Fourth, P7 depends on the motion pixels to locate the head position might not be efficient when comes to crowded scene. Classification of head position using a classifier i.e. shapes of head (contour) and SVM as head classifier might increase the accuracy in locating the head of humans appear in

the scene. However, this classification process might reduce the overall speed of computation. Hence, a better yet fast approach has to be added in the proposed approach.

As for the fifth is the future work for P8. Similar to the movement direction classes, there are also 9 predefined head pose categories (8 head pose categories and 1 without head pose). This might not be enough to represent the head poses of humans in real situation. Thus, more categories or a better way to represent the head poses have to be defined. The rules that govern the classification of head poses might also be insufficient since there are only 32 predefined rules. More rules could be found and added into the group of governors for a better head pose estimation result. This might solves the hair colour and hair style problems at the same time (only dark coloured hair and normal hair style could be detected as hair of the person of interest).

When the object of interest is small in size, the proposed algorithm still can estimates his or her head pose category, however the accuracy might be lower. Image super resolution technique could be implemented for this kind of situation to increase the precision of head pose estimation.

Lastly, the proposed system aimed to function during daytime. Some improvement could be made that is to enable the system to function during night time. By automates the adjustment of threshold values of the system, the system could adjusts the threshold values automatically based on the

brightness of the targeted scene. This would generally increase the usability, capability as well as the value of proposed system significantly.

6.4 Conclusion

Computer vision and image processing are becoming popular in many areas such as security, business, advertising, psychology, etc. as stated in chapter 1. More and more smart systems have been developed that aim to aid or lessen the work load of humans, thus saving more valuable human resources for other purposes.

This proposed system aims to detect and track humans that appear in the scene, followed by estimating the head pose of that particular person for monitoring purpose. Such monitoring process could detects some actions i.e. looks around the surrounding area, constant focused on a particular location, and looks towards the direction of movement, that are commonly act by humans. This process could be further modifies (additional head movements) and implements in various applications to fulfil different needs, i.e. for business area, such system could identifies the most viewed location by customers who walk into a shop. The most viewed location can be used to place their new product for promotion, etc. to increase their sales.

To develop the proposed system, some related research works has been studied and summarized in chapter 2-Literature Review. This study enables

the generation of basic ideas about the implementation of proposed system as finalized in chapter 3-System Design and Methodology. Upon the forming of general process flow of proposed system, the details i.e. technique used in each processes were recorded in chapter 4-System Implementation.

All the information about the system testing was discussed in chapter 5-Result and Discussion using 5 carefully selected testing samples. Those 5 testing samples include 1 best case, 1 worst case, and 3 normal cases. The system was evaluated based on some predefined evaluation mechanisms that are suitable for the proposed system. All the results from the 5 cases were analysed and studied. Finally, those findings were summarized in chapter 5-Result and Discussion.

From chapter 5, the testing outcome of this system shows that the proposed system is capable of monitoring humans that appear in the scene for detecting some actions i.e. looks around the surrounding area, looks towards a particular direction, and looks towards the movement direction and acceptable computational speed for real time implementation.

This system has met the objectives and was able to finish on time as planned using Gantt chart in chapter 1. The system can perform well in real time basis, but there are also a few limitations as listed in section 6.3 *Future Work*.

Reference

- 100fps.com. (2011). Retrieved September 6, 2011, from http://www.100fps.com/how_many_frames_can_humans_see.htm.
- 2SEETV the Easy Way to Buy. (2011). Retrieved April 11, 2011, from <http://www.2seetv.co.uk/>.
- Answers.com WikiAnswers. (2011). Retrieved September 6, 2011, from http://wiki.answers.com/Q/What_is_the_minimum_number_of_frames_per_second_that_can_be_perceived_by_the_human_eye_as_motion.
- Ba, S. O., & Odobez, J. M. (2004). A Probabilistic Framework for Joint Head Tracking and Pose Estimation. *Proceedings of the 17th International Conference on Pattern Recognition*, 4, 264-267.
- Ba, S. O., & Odobez, J. M. (2007). From Camera Head Pose to 3D Global Room Head Pose Using Multiple Camera Views. *Proceedings of International Workshop of Classification of Events Activities and Relationships*.
- Balasubramanian, V. N., Ye, J., & Panchanathan, S. (2007). Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-7.

- Baranda, J., Jeanne, V., & Braspenning, R. (2008). Efficiency Improvement of Human Body Detection with Histograms of Oriented Gradients. *Second ACM/IEEE International Conference on Distributed Smart Cameras*, 1-9.
- Beleznai, C., & Bischof, H. (2009). Fast Human Detection In Crowded Scenes By Contour Integration and Local Shape Estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2246-2253.
- Beleznai, C., Frühstück, B., Bischof, H., & Kropatsch, W. (2004). Detecting Humans in Groups using a Fast Mean Shift Procedure. *Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition*, 71-78.
- Bhuvaneswari, K., & Abdul, R. H. (2009). Edgelet Based Human Detection and Tracking by Combined Segmentation and Soft Decision. *International Conference on Control, Automation, Communication and Energy Conservation*, 1-6.
- Buchsbaum, G. (1980). A Spatial Processor Model for Object Colour Perception. *Journal of The Franklin Institute*, 310(1), 1-26.
- Cao, C., Li, R., & Ge, L. (2010). Real-time Multi-hand Posture Recognition. *International Conference on Computer Design and Applications (ICCD)*, 1, 619-635.

- Chakraborty, B., Rudovic, O., & Gonzalez, J. (2008). View-Invariant Human-Body Detection with Extension to Human Action Recognition using Component-Wise HMM of Body Part. *8th IEEE International Conference on Automatic Face & Gesture Recognition*, 1-6.
- Chen, T. W., Hsu, S. C., & Chien, S. Y. (2007). Automatic Feature-Based Face Scoring In Surveillance Systems. *9th IEEE International Symposium on Multimedia*, 139-146.
- Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 564-577.
- Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 886-893.
- Denman, S., Fookes, C., & Sridharan, S. (2009). Improved Simultaneous Computation of Motion Detection and Optical Flow for Object Tracking. *Digital Image Computing: Techniques and Applications*, 175-182.

- Ding, X., Xu, H., Cui, P., Sun, L., & Yang, S. (2009). A Cascade SVM Approach for Head-Shoulder Detection Using Histograms of Oriented Gradients. *IEEE International Symposium on Circuits and Systems*, 1791-1794.
- Dong, L., Tao, L., Xu, G., & Oliver, P. (2009). A Study of Two Image Representations for Head Pose Estimation. *5th International Conference on Image and Graphics*, 963-968.
- Engelmore, R. S., & Feigenbawn, E. (1993). Expert Systems and Artificial Intelligence. *WTEC Hyper-Librarian*. Retrieved from http://wtec.org/loyola/kb/c1_s1.htm.
- ezCCTV.com Digital Surveillance & IP CCTV Specialists. (2011). Retrieved April 11, 2011, from <http://www.ezcctv.com/cat/CCTV-Cameras-14.htm>.
- Fisher, R., Santos-Victor, J., & Crowley, J. (2005). "Caviar: Context aware vision using image-based active recognition". Retrieved September 29, 2010, from <http://homepages.inf.ed.ac.uk/rbf/CAVIAR>
- Foytik, J., Asari, V. K., Youssef, M., & Tompkins, R. C. (2010). Head Pose Estimation from Images Using Canonical Correlation Analysis. *IEEE 39th Applied Imagery Pattern Recognition Workshop (AIPR)*, 1-7.

- Gavrila, D. M. (2007). A Bayesian, Exemplar-Based Approach to Hierarchical Shape Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8), 1408-1421.
- Ghaemini, M. H., Shabani, A. H., & Shokouhi, S. B. (2010). Adaptive Motion Model for Human Tracking Using Particle Filter. *20th International Conference on Pattern Recognition*, 2073-2076.
- Health Decision Strategies. (2011). Retrieved October 30, 2011, from <http://www.healthstrategy.com/epiperl/epiperl.htm>.
- HERMES dataset (2007). Retrieved September 29, 2010, from <http://www.cvmt.dk/projects/Hermes/head-data.html>
- Hu, N., Huang, W., & Ranganath, S. (2005). Head Pose Estimation by Non-Linear Embedding and Mapping. *IEEE International Conference on Image Processing*, 2, II-342-5.
- Hu, Y., & Huang, T. S. (2008). Subspace Learning for Human Head Pose Estimation. *IEEE International Conference on Multimedia and Expo*, 1585-1588.
- Huang, J., Shao, X., & Wechsler, H. (1998). Face Pose Discrimination Using Support Vector Machines (SVM). *Proceedings of 14th International Conference on Pattern Recognition*, 1, 154-156.

Huang, S., & Sun, B. (2010). An Algorithm for Real-Time Human Tracking under Dynamic Scene. *2nd International Conference on Signal Processing Systems*, 3, V3-590-V3-593.

Human Feature Extraction in VS Image Using HOS Algorithm. (2007). *University of Science & Technology of China*. Retrieved from <http://www.docstoc.com/docs/33915378/Human-Feature-Extraction-in-VS-image-Using-HOG-Algorithm>.

HumanEva dataset (2007). Retrieved September 29, 2010, from <http://vision.cs.brown.edu/humaneva/download1.html>

INRIA Person Dataset (2005). Retrieved September 29, 2010, from <http://lear.inrialpes.fr/data>

Ishii, Y., Hongo, H., Yamamoto, K., & Niwa, Y. (2004). Face and Head Detection for A Real-Time Surveillance System. *Proceedings of the 17th International Conference on Pattern Recognition*, 3, 298-301.

Isti, S., & Annury, C.S. (2005). Design of Product Placement Layout in Retail Shop Using Market Basket Analysis. *MAKARA Seri TEKNOLOGI* (pp. 43-47). Misri, G.

- Jia, H. X., & Zhang, Y. J. (2007). Fast Human Detection By Boosting Histograms Of Oriented Gradients. *Journal of Image and Graphics*, 683-688.
- Joachims, T. (1999). Making Large-Scale SVM Learning Practical. *Advances in kernel Methods – Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.
- Kruger, N., Potzsch, M., & von der Malsburg, C. (1997). Determination of Face Position and Pose with a Learned Representation Based on Labeled Graphs. *Image and Vision Computing*, 15, 665-673.
- Lablack, A., & Djeraba, C. (2008). Analysis of Human Behaviour in Front of A Target Scene. *19th International Conference on Pattern Recognition*, 1-4.
- Lablack, A., Zhang, Z., & Djeraba, C. (2008). Supervised Learning for Head Pose Estimation Using SVD and Gabor Wavelets. *10th IEEE International Symposium on Multimedia*, 592-596.
- Lades, M., Vorbruggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Wurtz, R., P., & Konen, W. (1993). Distortion Invariant Object Recognition in the Dynamic Link Architecture. *IEEE Transactions of Computing*, 42, 300-311.

- Lin, H. J., Chang, C. W., & Pai, I. C. (2009). Head Pose Estimation based on Nonlinear Interpolative Mapping. *2009 Joint Conferences on Pervasive Computing*, 89-94
- Liu, X., Lu, H., & Luo, H. (2009). A New Representation Method of Head Images for Head Pose Estimation. *16th IEEE International Conference on Image Processing (ICIP)*, 3585-3588.
- Lu, Y., Xu, D., Wang, L., Hartley, R., & Li, H. (2010). Illumination Invariant Sequential Filtering Human Tracking. *International Conference on Machine Learning and Cybernetics*, 4, 2133-2138.
- Lucas, B., & Kanade, T. (1981). An Iterative Image Registration Technique with An Application to Stereo Vision. *Proceedings of DARPA IU Workshop*, 121-130.
- Ma, B., Shan, S., Chen, X., & Gao, W. (2008). Head Yaw Estimation From Asymmetry of Facial Appearance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(6), 1501-1512.
- Ma, B., & Wang, T. (2010). Head Pose Estimation Using Sparse Representation. *2nd International Conference on Computer Engineering and Applications*, 2, 389-392.

Maggie Mae's Site, Exercise 8: Skin Detection Algorithms. (2009). Retrieved August 10, 2011, from <http://maggieroxas.multiply.com/journal/item/4>.

OpenCV2.0. (2009). Retrieved September 30, 2009, from <http://opencv.willowgarage.com/wiki/>.

Osuna, E., Freund, R., & Girosi, F. (1997). Training Support Vector Machines: An Approach to Face Detection. *Proceedings of IEEE Conference of Computer Vision and Pattern Recognition*, 130-136.

Rashid, M. E., Remya, S., & Wilsy, M. (2009). Fast Tracking of Humans in Frequently Occurring Entry, Exit and Occlusion Scenarios. *International Conference on Computer Technology and Development*, 2, 327-330.

Raytchev, B., Kimura, Y., Yoda, I., & Sakaue, K. (2010). Real-time 3D Head Pose Estimation Using Both Geometry and Learning. *17th IEEE International Conference on Image Processing (ICIP)*, 1525-1528.

Raytchev, B., Yoda, I., & Sakaue, K. (2004). Head Pose Estimation by Nonlinear Manifold Learning. *17th International Conference on Pattern Recognition*, 4, 462-466.

- Rowley, H., Baluja, S.,& Kanade, T. (1998). Neural Network-Based Face Detection. *IEEE Transaction on Pattern Analysis Machine Intelligent*, 20, 23-38.
- Saxena, V., Grover, S.,& Joshi, S. (2008). A Real Time Face Tracking System Using Rank Deficient Face Detection and Motion Estimation. *7th IEEE International Conference on Cybernetic Intelligent Systems*, 1-6.
- Shabani, A. H., Ghaemini, M. H.,& Shokouhi, S. B. (2010). Human Tracking Using Spatialized Multi-Level Histogram and Mean Shift. *Canadian Conference on Computer and Robot Vision*, 151-158.
- Shan, L., Tsechpenakis, G., Metaxas, D. N., Jensen, M. L.,& Kruse, J. (2005). Blob Analysis of the Head and Hands: A Method for Deception Detection. *Proceedings of the 38th Annual Hawaii International Conference on System Science*, 20c-20c.
- Sherrah, J., & Gong, S. (2001). Fusion of Perceptual Cues for Robust Tracking of Head Pose and Position. *Pattern Recognition*, 34(8), 1565-1572.
- Tang, S.,& Goto, S. (2009a). Partially Occluded Human Detection by Boosting SVM. *5th International Colloquium on Signal Processing & Its Application*, 224-227.

- Tang, S.,& Goto, S. (2009b). Human Detection using Motion and Appearance Based Feature. *7th International Conference on Information, Communications and Signal Processing*, 1-4.
- Tang, S.,& Goto, S. (2010). Histogram of Template for Human Detection. *IEEE International Conference on Acoustics Speech and Signal Processing*, 2186-2189.
- Thanh, N. D., Ogunbona, P.,&Li, W. (2009). Human Detection Based on Weighted Template Matching. *IEEE International Conference on Multimedia and Expo*, 634-637.
- Thombre, D. V., Nirmal, J. H.,& Lekha, D. (2009). Human Detection and Tracking Using Image Segmentation and Kalman Filter. *International Conference on Intelligent Agent & Multi-Agent Systems*, 1-5.
- Trecvid dataset (2004). Retrieved September 29, 2010, from <http://trecvid.nist.gov/trecvid.data.html>
- Tsai, C. C., Cheng, W. C., Taur, J. S.,& Tao, C. W. (2006). Face Detection Using Eigenface and Neural Network. *IEEE International Conference on Systems, Man, and Cybernetics*, 5, 4343-4347.

- Tsechpenakis, G., Metaxas, D., Adkins, M., Kruse, J., Burgoon, J. K., Jensen, M. L., Meservy, T., Twitchell, D. P., Deokar, A., & Nunamaker, J. F. (2005). HMM-Based Deception Recognition From Visual Cues. *IEEE International Conference on Multimedia and Expo*, 824-827.
- Tu, J., Fu, Y., & Huang, T. S. (2009). Locating Nose-Tips and Estimating Head Poses in Images by Tensorposes. *IEEE Transactions on Circuits and Systems for Video Technology*, 1, 90-102.
- Vatahska, T., Bennewitz, M., & Behnke, S. (2007). Feature-based Head Pose Estimation from Images. *7th IEEE-RAS International Conference on Humanoid Robots*, 330-335.
- Viola, P., & Jones, M. (2001a). Rapid Object Detection Using A Boosted Cascade of Simple Features. *Proceedings of IEEE Conference of Computer Vision and Pattern Recognition*, 511-518.
- Viola, P., Jones, M. (2001b). Robust Real-time Object Detection. *2nd International Workshop on Statistical and Computational Theories of Vision- Modeling, Learning, Computing, and Sampling*, 1-25.
- Viola, P., & Jones, M. (2002). Robust Real-time Object Detection. *International Journal of Computer Vision*, 137-154.

Wikipedia. (2011). Retrieved September 6, 2011, from http://en.wikipedia.org/wiki/Frame_rate.

Wilson, H., Wilkinson, F., Lin, L., & Castillo, M. (2000). Perception of Head Orientation. *Vision Research*, 40, 459-472.

Wu, J., Pedersen, J., Putthividhya, D., Norgaard, D., & Trivedi, M. M. (2004). A Two-Level Pose Estimation Framework Using Majority Voting of Gabor Wavelets and Bunch Graph Analysis. *Proceedings of Pointing 2004 Workshop: Visual Observation of Deictic Gesture*.

Wu, J., & Trivedi, M. (2008). A Two-Stage Head Pose Estimation Framework and Evaluation. *Pattern Recognition*, 41, 1138-1158.

Yin, X., & Zhu, X. (2006). Hand Posture Recognition in Gesture-Based Human-Robot Interaction. *IEEE Conference on Industrial Electronics and Applications*, 1-6.

Yun, F., & Huang, T. S. (2006). Graph Embedded Analysis for Head Pose Estimation. *7th International Conference on Automatic Face and Gesture Recognition*, 6-8.

- Zhang, J., Sun, H., Guang, W., Wang, J., Xie, Y., & Shang, B. (2010). Robust Human Tracking Algorithm Applied for Occlusion Handling. *5th International Conference on Frontier of Computer Science and Technology*, 546-551.
- Zhang, L., & Liang, Y. (2010). Motion Human Detection Based on Background Subtraction. *2nd International Workshop on Education Technology and Computer Science*, 284-287.
- Zhang, X., Zheng, N., Mu, F., & He, Y. (2009). Head Pose Estimation using Isophote Features for Driver Assistance Systems. *IEEE Intelligent Vehicles Symposium*, 568-572.
- Zhang, Z., Hu, Y., Liu, M., & Huang, T. (2007). Head Pose Estimation in Seminar Room Using Multi View Face Detectors. *Multimodal Technologies for Perception of Humans, International Workshop Classification of Events Activities and Relationships (CLEAR), ser. Lecture Notes in Computer Science, R. Stiefelhagen and J. Garofolo, Eds.*, 4122, 299-304.
- ZuWhan, K. (2008). Real Time Object Tracking Based on Dynamic Feature Grouping with Background Subtraction. *IEEE Conference on Computer Vision and Pattern Recognition*, 1-8.

Appendix A

PUBLICATION – A NOVEL NRBVS APPROACH FOR LONG DISTANCE HEAD POSE ESTIMATION

A Novel NRBVS Approach for Long Distance Head Pose Estimation

Che Yon Choo, Siak Wang Khor
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman
Kuala Lumpur, Malaysia

E-mail: joshua5367@hotmail.com, khorsw@utar.edu.my

Abstract—Head pose is an important element for automated surveillance system since it usually coincides with the gaze direction. It is essential for the system to know the focus-of-attention of an individual in order to obtain his/her object of interest. In this paper, we present a novel Nested Rule-Based Voting System (NRBVS) for long distance head pose estimation based on the regional color information of the head. First, the head image is divided into regions, then the color information of each region is retrieved and estimates the head pose based on the predefined rules. It is tested with 100 random images and the experimental result shows more than 80% accuracy.

Keywords: long distance head pose estimation, rule-based voting system, head pose estimation, regional color information

I. INTRODUCTION

Computer vision and image processing is becoming popular in many area such as surveillance purposes, business, advertising, psychology, etc. [3, 8, 9, 17, 18]. More and more smart system is developed which aims to aid or lessen the work load of humans, ultimately save more valuable human resources for other purposes.

There are a lot of components that can be used in smart system, such as human hand posture (robotic in industrial services) [2, 14]; face and head (in surveillance field) [3, 10, 19]; head pose (for human computer interaction) [4, 9], etc.. Among all the components, head pose is one of the useful and important component (in which head pose usually coincides with the gaze direction) [5]. For example, to detect the first region of focus of a customer who walk into a shop (for business purpose, to improve shop design); to calculate the number of person who attracted to a particular advertisement (for business purpose, to measure the advertisement effectiveness); to track and analyze sequence of region of focus of an individual (for surveillance purpose), etc.

Those smart system described need the use of overhead wide-angle cameras in order to obtain the information of the entire scene. This kind of camera usually is mounted at the corner just below the ceiling and it is best if the camera is capturing the whole concerned area.

Due to that purpose, the source of the smart system (video stream from CCTV system) is usually not in high quality (some of the CCTV system is not in high definition recording

[22, 23]) and this might cause the captured signal to be blurred. In addition, the long distance setting makes everything in the scene become smaller, and what make it worst is the inconsistency brightness and contrast of the input video caused by variant weather conditions (such as sunny day, cloudy day, etc.). This boost the difficulty to interpret the signal information received by the smart system. For this kind of scenario and environment setting, estimating head pose is a challenging task. The long distance setting causes the detection of face features (such as eye, ears, etc.) to fail and so for head pose estimation. Color information is more reliable compare to face features, shapes, etc. Therefore, estimating head pose in long distance has to be researched and developed.

Many head pose estimation technique performs well in close range estimation. Methods by scholars can be generalized into three main categories: appearance-based methods [1, 6, 7, 9], feature-based methods [4, 13] and manifold-learning & dimensionality reduction based approach [3, 11, 12, 15, 20].

To estimate head pose in long distance, a novel approach is proposed based on the color information of the head. The NRBVS first extract the percentage of skin and dark color within the 7 regions divided before hand, followed by the two session of voting process to estimate the head pose where each predefined voting rules have their own reasoning and believes for the vote given. The category with highest vote is the estimated head pose.

The rest of this paper is organized as follows. Section 2 discusses the related works of the head pose estimation approaches. Section 3 explains in depth about the architecture and methodology used in our proposed system. Section 4 presents the experimental results and lastly Section 5 concludes the paper with key findings and suggestions of future work in the similar area.

II. RELATED WORKS

While many head pose estimation technique perform well for close range estimation and under different illumination condition in more or less controlled environment, they would undergo degradation of performance under complex scene such as occluded condition, long range distance between camera and head, low light condition, etc. These problems

has been recognized by researchers and lots of research has been carried-out to resolve those problems, yet it still remains a highly challenging task since it required a combination of speed, accuracy as well as robustness for the implementation of such security system.

Head pose estimation methods generally can be grouped into three main categories: appearance-based methods, feature-based methods and manifold-learning & dimensionality reduction based approach.

Appearance based approach

Appearance-based methods [1, 6, 7, 9] formulate the head pose estimation problem as a pattern classification problem on image feature space. It used the whole face or head image rather than some distinctive features such as nose-tips and eyes for head pose estimation. Number of classes defines the accuracy of the pose estimation that can be achieved. Most of the appearance-based method suffers from the redundancy of the information cause by pose, illumination, expression, occlusion and background. Different testing subject would also resulting in different experimental result. However, despite those weaknesses, appearance-based method is less expensive in computation compare to feature based and dimensionality reduction based method in such a way that, it does not required additional steps such as detection of distinctive features (nose-tip detection, eye detection, etc.) or reduces the dimension of features by mapping it into the feature subspace.

Feature based approach

Feature-based methods [4, 13] refers to the used of distinctive features such as nose-tips, eye, ears, etc. for head pose estimation. It usually comes with a face or head detector to locate the location of face or head of an individual followed by feature detection such as eye detection or nose-tip detection to extract those distinctive features. The head pose is estimated based on the information of those features.

Manifold-learning and dimensionality reduction approach

To obtain a better representations of face images, [15] and [20] suggests the used of high-dimensional feature space of face image data as a set of geometrical related points lying on a smooth low-dimensional manifold in feature space. This suggestions brings the used of manifold-learning & dimensionality reduction based approach [3, 11, 12, 15, 20] for head pose estimation where linear or non-linear embedding of the face images is used as to delete irrelevant information. The best thing about this approach is the simplification of the overall classification process and resulting in obtaining a more accurate classification. However, this method is time consuming while evaluating and normalizing the unified embedding space and the error produced would be transmitted to the training process for interpolative mapping function and cause augmented error. To overcome this problem, [3] has made some improvement on the head pose estimation method by skipping the embedding process using the Isomap and replacing the training process with a supervised training scheme.

III. METHODOLOGY

Proposed NRBVS method for head pose estimation takes a head image as input and estimated head pose as output. The estimation result is decided based on the 2-session of voting process and 32 predefined rules. The overall structure of the proposed method is shown as in figure 1.

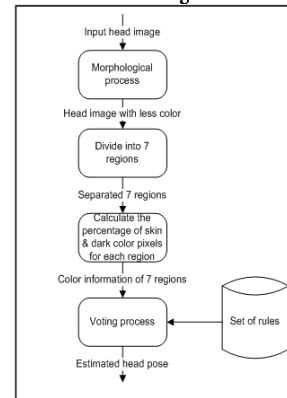


Figure 1. Overall structure of proposed system.

The head pose is categorized into 8 different postures as shown in figure 2. The estimation of head pose is based on the novel NRBVS approach which made use of the color information (dark and skin color) of the head image. In here, there are assumptions made which is all the input head image have black or dark hair color and Asian skin color (example as in figure 2).



Figure 2. 8 categories of head pose.

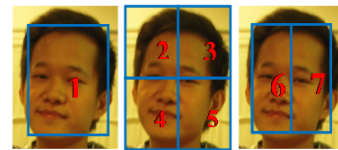


Figure 3. 7 regions divided of input image

Initially, the input image is divided into 7 regions and the region numbering is as shown in figure 3 above. The image is divided in this way is due to that the most important information is right at the center where normally is the location of the face. This dividing and numbering will be used by NRBVS approach. NRBVS approach is a novel approach which made use of the idea of democratic. In democratic country, the leader of the country is selected by all the members of the country itself through the voting system. Each people in the country have only one vote for them to vote for their desired candidate as leader. In this case, each vote is independent and each vote stands equal amount of weight in the final vote counting. The decision of choosing a leader is based on the majority and the result of voting is final. In short, the candidate with the most votes is selected as

leader. NRBVS approach is similar to this system where the candidate is the 8 different head poses and the voter is the set of predefined rules.

Just before the voting process, some essential information that is necessary for the set of rules to make their decision (vote) has to be generated. That information including the percentage of dark color pixels within each divided regions (shown in figure 3) and the percentage of skin color pixels within each divided regions. Once the information is generated, the set of rules will start voting based on the color information.

During the voting process, all the votes generated by rules are collected. The category with the highest vote will be chosen as the estimated head pose for the input image.

A. Color model

HSV color model is used for detecting skin and dark color region. Simple filtering to filter out non-skin and non-dark color region of the input image. The skin color is defined as hue less than 20 and saturation more than 60; where the dark color is defined as 30% of value of the color. These values are obtained by observation during training phase.

B. NRBVS engine

NRBVS engine is used to classify an input image's head pose based on the pixel's color information. For this NRBVS engine to be a success, a proper training is necessary. This training is meant for constructing a set of rules that used for the voting purposes. To construct such rules, 200 head images (self-captured, 25 images for each head pose, 8 head poses in total) are used. A total of 32 rules are produced from the training (16 rules for each voting session).

The whole voting process is divided into 2 sessions where the first session of voting is to classify the input image into one of the four main groups of head pose (refers to next section) and the second session is to classify the input image as one of the head pose in that specific group of head pose.

C. Grouping of head pose

As mentioned in the previous section, all the head pose (8 of them) are divided into four different groups based on the similarity (in terms of skin and dark color distribution) of the head poses. The purpose of this division is to narrow down and reduce the classification complexity (due to the similarity of head pose within a group and the significant differences between groups) and hence for a better and closer estimation of head pose for the input image.

The head poses in each groups share much similarity such as distribution of skin and dark color (For instance, group 3 consists of head poses which the center region of the head image consist of more than 80% of dark color). Figure 4 shows the information of 8 categories of head poses (obtained from average of training samples).

The main reason why there are only 4 groups of head pose is due to the uniqueness of some of the head poses. From figure 4, notice that some head poses are quite similar to another head pose (e.g. head pose category 1 and 2) which are suppose to be in the same group, and some of the head poses are different to another head pose (e.g. head pose

category 1 and 5) which are suitable to be the base of a group. Based on observation, the most suitable or the head poses with the significant differences are selected as the base of the group. The selected head poses are category 1, 3, 5, and 7.

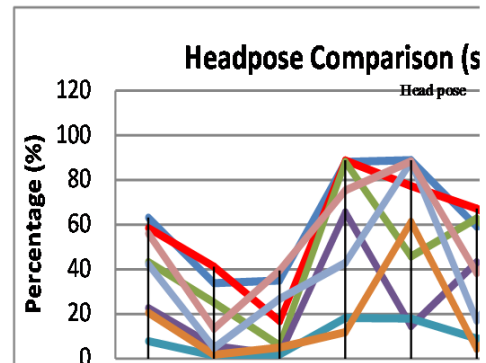


Figure 1. Comparison of 8 categories of head poses (skin color).

Observe that, each head poses are actually interrelated. Part of the characteristics of a head pose can be found on its neighboring head pose (e.g. characteristics of head pose category 2 can be found on head pose category 1 and 3). For instance, based on the figure 4 and 5, the graph pattern of the head pose category 1 and 2 are similar or with minor variation, but overall pattern of the information are similar. Therefore, head pose category 1 and 2 should be in the same group. The same goes to head pose 2 and 3, etc. By comparing non-base head pose (head pose category 2, 4, 6, and 8) to its neighboring base head pose (head pose category 1, 3, 5, and 7), the grouping of the head pose are done. The final grouping of head poses are in the table 1.

TABLE I. GROUPING OF HEAD POSES

Group	Head poses
1	Category 1, 2, & 8
2	Category 2, 3, & 4
3	Category 4, 5, & 6
4	Category 6, 7, & 8

D. Construction of rules

Rules are the main core of this NRBVS engine in which they represent the absolute statements for the whole voting process. This idea of voting system with the use of rules is actually some kind of inference in artificial intelligent in which the rules are said the facts or statements that are always correct and may not be disobey. This kind of inference normally used in artificial intelligent system for decision making. In this project, the rules are viewed as absolute instructions for certain actions to be done based on the given information.

There are total of 32 rules constructed in this project, 16 rules for each session of voting. The objective of rules in voting session 1 and 2 are different. In voting session 1, the rules are meant for finding the main group of head pose (where the input image belongs to) based on the given input image. The expected output from the voting session 1 is the group of head pose in which the given input image belongs

to. Next, the voting session 2 begins to decide which head pose the input image is, based on the rules. In this section, the construction of rules in each session will be showed.

Rules for voting session 1

The rules in voting session 1 are constructed based on the similarities between each head pose within the group and the differences between groups of head pose. Figure 5 shows the characteristics of head pose group 1 (refers to TABLE I for head pose grouping).

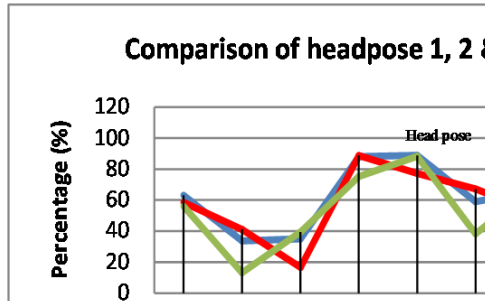


Figure 5. Comparison of head pose group 1 (skin color).

From the figure 5, observe that the patterns of the curves are similar since they are in the same group. From those observations, all the characteristics of each group of head pose can be summarize as in the TABLE II (skin color only).

TABLE II. SUMMARIZE OF HEAD POSE GROUP CHARACTERISTIC (SKIN COLOR)

Region	Percentage of skin color (%)			
	Group 1	Group 2	Group 3	Group 4
1	>40	>10 & <70	<30	>10 & <70
2	>10 & <50	<50	<15	<30
3	>10 & <50	<30	<15	<50
4	>60	>50	>0 & <70	-
5	>60	-	>0 & <70	>50
6	>30 & <80	>30	>0 & <50	<50
7	>30 & <80	<70	>0 & <50	>20 & <30
Comparison between region				
2 & 3	2 ≈ 3**	2 > 3	2 ≈ 3*	2 < 3
4 & 5	4 ≈ 5*	4 > 5	4 ≈ 5**	4 < 5
6 & 7	-	6 > 7	-	6 < 7

Remark:

*15% threshold

**special case, only valid for specific head pose

Refers to the remark, some of the characteristics are not shared among head pose within a group but belongs to a specific head pose only. However, this information is essential for the voting, therefore it is included as part of the characteristics that shared (not in real) by head poses within the group. With the characteristics obtained, rules can be generated for classifying groups. One of the rules that generated is illustrated in figure 6.

The blue rectangles in figure 6 is the instruction or statements that must be followed for decision making; whereas for the red rectangles are the decision made based on the instruction (in blue rectangle) and given information (input head image). The rule structure is from top to bottom. It starts at the first top blue rectangle and ends with a red rectangle.

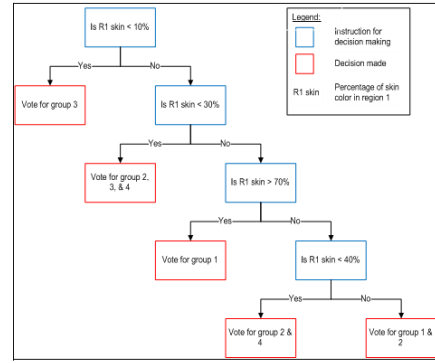


Figure 6. Example rule generated.

Refers to the percentage of skin color of region 1 in TABLE II and first top blue rectangle in figure 6, the only group of head pose that has lower than 10% skin color is group 3. Therefore, the first decision is to vote for group 3 if the input image satisfy the first statement. Then, check again for next possible statement. From TABLE II again, it is observable that group 2, 3, and 4 will have the percentage from skin color between 10-30% and this became the second statement for this rule. Using the similar way, all the statement and decision (for both skin color and dark color information) can be found and formed a complete rule. The rule structure is similar to a tree structure where the root will only have zero or two branches constructed using some conditional if-else choices (nested if-else) in the programming language. This is how the name nested rule-based (from NRBVS engine) is formed.

Other than forming rules based on percentage of skin or dark color, the characteristics between regions can also be used to form a rule. Figure 7 shows one of the rules that formed using the comparison characteristics between regions.

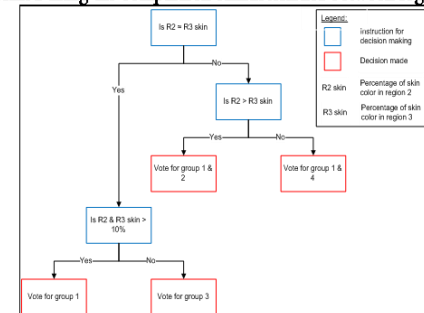


Figure 7. Example rule generated.

For relationship type of statement, there are only three possible outcomes. It is either two regions are similar in percentage of skin or dark color or one region contains higher percentage of skin or dark color. In figure 7, the rule is formed using this approach. All the rules in voting session 1 are formed using similar approach described previously.

Rules for voting session 2

In voting session 2, it is similar to that of session 1 but instead of classifying the input image into groups, it classify into specific head pose that the estimation engine believes.

The rules in voting session 2 are special. Each rules are designed for only specific combination of candidates (possible head pose), since the output from voting session 1 is the possible head pose (2 and above). For example, group 1 having the highest vote and group 2 having votes second to that of group 1. In this case, the possible head pose would be head pose category 1 and 2 (refers to E for candidate selection) and only the rules that specifically designed to classify whether the input image is head pose category 1 or 2 will be use. Figure 8 shows the rule for the example above.

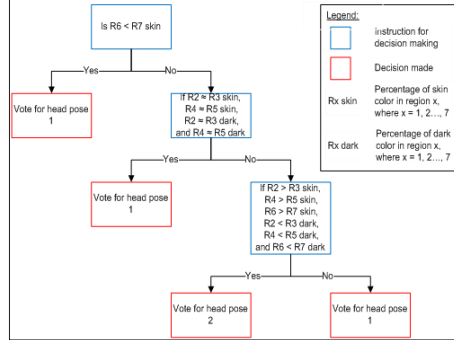


Figure 8. Example rule.

The method to construct the rules in voting session 2 is similar to that of in session 1. We observe the uniqueness of each candidates and form the rules in similar way.

A. Voting process

Before the discussion of voting process, the term "candidates" and "voters" have to be defined. In voting process, a candidate is a person, a thing, or anything that is being vote, and one of the candidate would win the election and being selected to do something; as for voters, is anything that made a decision on choosing the candidate which most suitable or fitting to be selected based on the information they have or what they believed.

In this process of voting, different sessions have different candidate and voters. For voting session 1, the candidate is the group of head pose (four of them) and the voter is the set of rules (16 of them in voting session 1); for voting session 2, the candidate is the specific head pose (could be different based on the input image and the voting result from voting session 1) and the voter is some of the rules out of 16 of them in voting session 2 (due to that not all the rules will be use depending on the situation). In the following discussion, the terms candidate and voters will refers to its meaning in voting session 1 and 2. The discussion starts with a simple flow diagram of the process of voting in figure 9.

As shown in figure 9, the voting process starts by calculating the image's features (percentage of skin and dark color in each region) based on the color model. Later on, the voters (discussed in D) will start to vote for the best candidates based on their own reasoning and believes and finally the voting result will be sorted and the top and second candidates will be selected and a new set of candidates will be chosen for the next session of voting.

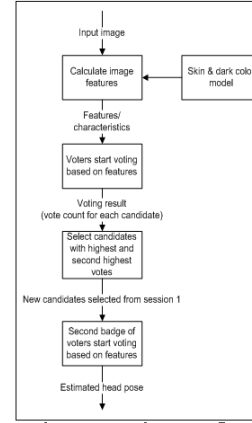


Figure 9. Voting process flow.

The selection of new set of candidates is based on the result from voting session 1. Since the head pose are interrelated, the voting result from session 1 will brings its meaning for defining the possible head pose of that input image. For example, the highest vote goes to group 1 and second goes to group 2. In this case, the input image can be said that it consists a set of characteristics that similar to group 1 and some of that of group 2. Therefore, the possible head pose for that input image could be that between group 1 and 2 (which is head pose category 1 and 2). Here, the head pose category 3 is not included since the highest vote goes to group 1, it is more similar to group 1 compare to group 2. This is how the candidates are selected for the voting session 2.

After the candidate for voting session 2 is selected, only specific voters will be allowed to vote for their best candidate, again based on their reasoning and believes. Unlike session 1, voting session 2 will produce the estimated head pose for the input image. This is how the whole voting process proceed using the rules created.

III. EXPERIMENT

To validate the method's effectiveness, we implemented the system in C++ on an Intel Core2Duo machine with 1.83 GHz processor and 2 GB RAM, and we captured 200 images (25 images for each head pose, 8 head pose in total) as training dataset and 100 different size random images (from internet) as testing dataset. The testing dataset includes models, peoples in posters, peoples having group activities in college, peoples in shopping mall, peoples in restaurant gathering, peoples in seminar, etc.

TABLE III shows some of the estimation result among the 100 testing samples and the justifications.

Figure 10 shows the overall head pose estimation result using NRBVS method. Out of 100 testing images 83% of them is classified correctly and 17% of them fail (some results and justifications are as shown in TABLE III) with average of 35 milliseconds estimation time per image.

TABLE III. HEAD POSE ESTIMATION RESULT

Image	Dimension (widthxheight)	Ground true (head pose category)	Estimation result (head pose category)	Justification
	32x39	5	5	Correct classification -
	33x40	5	5	Correct classification -
	50x57	8	8	Correct classification -
	28x30	1	1	Correct classification -
	32x39	3	2	Area left to the nose is dark in color. This might falsely identified as hair.
	33x35	7	7	Correct classification -
	85x107	8	8	Correct classification -
	95x92	1	5	Dominant dark color region caused by the lighting condition.

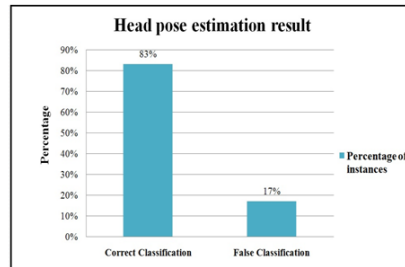


Figure 10. Head pose estimation result.

NRBVS engine is very sensitive to color. It is suitable for long distance estimation where the image is blurred or small. However, this characteristics may cause the estimation fails when the subject's hair color is different from dark color, different hair style, shadow caused by lighting conditions, accessory such as glasses and also the color model used

which detects the skin and dark color of the image. More and more training is needed to cover all those limitations.

III. CONCLUSION

This paper presented a novel nested rule-based voting system for head pose estimation. It achieved more than 80% accuracy with low amount of computational time using 100 random testing images. The NRBVS method is very sensitive to colors, it is suitable for long distance estimation where the image is blurred or small in which only color information might be useful compare to others such as facial feature. However, due to the sensitivity of NRBVS towards color information, it may cause the estimation fails when the subject's hair color is different from dark color, variety of hair style, etc. More and more training is needed to cover all those limitations. Additional rules and more region division can be implemented for more discriminative power for head pose estimation.

IV. REFERENCES

- [1] BingPeng, M., and TianJiang, W., "Head Pose Estimation Using Sparse Representation". *2nd International Conference on Computer Engineering and Applications*, 2010, vol. 2, pp. 389-392.
- [2] Cao, C., Li, R., and Ge, L., "Real-time Multi-hand Posture Recognition". *International Conference on Computer Design and Applications (ICDDA)*, 2010, vol. 1, pp. 619-635.
- [3] HweiJen, L., ChenWei, C., and IChun, P., "Head Pose Estimation based on Nonlinear Interpolative Mapping". *2009 Joint Conferences on Pervasive Computing*, 2009, pp. 89-94.
- [4] Jilin, T., Yun, F., and Huang, T. S., "Locating Nose-Tips and Estimating Head Poses in Images by Tensorposes". *IEEE Transactions on Circuits and Systems for Video Technology*, 2009, vol. 1, pp. 90-102.
- [5] LiGeng, D., LinMi, T., GuangYou, X., and Oliver, P., "A Study of Two Image Representations for Head Pose Estimation". *5th International Conference on Image and Graphics*, 2009, pp. 963-968.
- [6] Xuefao, Z., NanNing, Z., Fan, M., and YongJian, H., "Head Pose Estimation using Isophote Features for Driver Assistance Systems". *IEEE Intelligent Vehicles Symposium*, 2009, pp. 568-572.
- [7] BingPeng, M., ShiGuang, S., XiLin, C., and Wen, G., "Head Yaw Estimation from Asymmetry of Facial Appearance". *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, vol. 38(6), pp. 1501-1512.
- [8] Lablack, A., and Djenaba, C., "Analysis of Human Behaviour in Front of A Target Scene". *19th International Conference on Pattern Recognition*, 2008, pp. 1-4.
- [9] Lablack, A., ZhongFei, Z., and Djenaba, C., "Supervised Learning for Head Pose Estimation Using SVD and Gabor Wavelets". *10th IEEE International Symposium on Multimedia*, 2008, pp. 592-596.
- [10] Saxena, V., Grover, S., and Joshi, S., "A Real Time Face Tracking System Using Rank Deficient Face Detection and Motion Estimation". *7th IEEE International Conference on Cybernetic Intelligent Systems*, 2008, pp. 1-6.
- [11] YuXiao, H., and Huang, T. S., "Subspace Learning for Human Head Pose Estimation". *IEEE International Conference on Multimedia and Expo*, 2008, pp. 1585-1588.
- [12] Balasubramanian, V. N., JiePing, Y., and Panchanathan, S., "Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation". *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-7.
- [13] Vatahska, T., Bennewitz, M., and Behnke, S., "Feature-based Head Pose Estimation from Images". *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 330-335.
- [14] Xiaoming, Y. and Xing, Z., "Hand Posture Recognition in Gesture-Based Human-Robot Interaction". *IEEE Conference on Industrial Electronics and Applications*, 2006, pp. 1-6.

- [15] Yun, F., and Huang, T. S., "Graph Embedded Analysis for Head Pose Estimation", *7th International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 6-8.
- [16] Nan, H., WeiMin, H., and Ranganath, S., "Head Pose Estimation by Non-Linear Embedding and Mapping", *IEEE International Conference on Image Processing*, 2005, vol. 2, pp. II-342-5.
- [17] Shan, L., Tschepnakis, G., Metaxas, D. N., Jensen, M. L., and Kruse, J., "Blob Analysis of the Head and Hands: A Method for Deception Detection", *Proceedings of the 38th Annual Hawaii International Conference on System Science*, 2005, pp. 20c-20c.
- [18] Tschepnakis, G., Metaxas, D., Adkins, M., Kruse, J., Burgoon, J. K., Jensen, M. L., Mescrvy, T., Twitchell, D. P., Deokar, A., and Nunamaker, J. F., "HMM-Based Deception Recognition From Visual Cues", *IEEE International Conference on Multimedia and Expo*, 2005, pp. 824-827.
- [19] Ishii, Y., Hongo, H., Yamamoto, K., and Niwa, Y., "Face and Head Detection for A Real-Time Surveillance System", *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, vol. 3, pp. 298-301.
- [20] Raytchev, B., Yoda, I., and Sakaue, K., "Head Pose Estimation by Nonlinear Manifold Learning", *17th International Conference on Pattern Recognition*, 2004, vol. 4, pp. 462-466.
- [21] Baggenstoss, P.M., "Class-specific Classifier: Avoiding The Curse of Dimensionality", *IEEE Aerospace and Electronic System Magazine*, 2004, vol. 19(1), pp. 37-52.
- [22] "2SEETV the Easy Way to Buy". Internet: <http://www.2scctv.co.uk/> [April 11, 2011].
- [23] "Digital Surveillance & IP CCTV Specialists". Internet: <http://www.czccctv.com/cat/CCTV-Cameras-14.htm> [April 11, 2011].

Appendix B

PUBLICATION – LETTER OF ACCEPTANCE FOR MANUSCRIPT IN Appendix A

[ICIS 2011] Acceptance Notification

Dear Authors,

Thank you for your submission to ICIS2011, which will be held in Guangzhou during November 18-20, 2011. We are pleased to inform you that your paper:

ID: 10457

TITLE: **A Novel NRBVS Approach for Long Distance Head Pose Estimation**

has been accepted for publication in the proceedings of 2011 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2011). Congratulations! This year, we received more than 1000 submissions; only very outstanding paper can be accepted by the conference. Please revise your paper in detail according to the review results. **All papers accepted will be published in the IEEE categorized conference proceedings. All papers accepted will be included in IEEE Xplore and indexed by Ei Compendex and ISTP.**

Here are some important issues on registration and final paper submission:

- (1) At least one author of each accepted paper should register before Before September 30, 2011. Please visit: <http://www.ieee-icis.org/index6.asp>
- (2) Paper format: Note that your paper must be formatted according to the IEEE template files which can be downloaded from: <http://www.ieee-icis.org/index3.asp>
- (3) Please improve your paper according to the review results. The review information can be obtained by visiting the website <http://www.ieee-icis.org/login.asp> with your username and password.
- (4) Please submit your final papers (pdf file) and the signed completed copyright form to icis@ieee-icis.org before September 30, 2011. The blank copyright form can be found at: <http://www.ieee-icis.org/IEEECopyrightForm.doc>

Thank you very much for your contribution to this conference. We are looking forward to seeing you in Guangzhou, China.

Best Regards,

Shaozi Li, Technical Program Committee Chair

Shaozi Li

Wen Chen, General Chair

Wen Chen

ICIS2011 Organizing Committee.

August 31, 2011



