Development of situation recognition, environmental monitoring and patient condition monitoring service modules for hospital robots

A Thesis submitted to the Faculty of Engineering and Computing, School of Mechanical and Manufacturing Engineering, Dublin City University for the Degree of PhD

By

Md. Kabir Al Mamun

B.Sc., MEng.

School of Mechanical and Manufacturing Engineering

Dublin City University, Ireland

March, 2012

Research Supervisor

Dr. Tamas Szecsi

DECLARATION

I hereby certify that this material, which I now submit for the assessment on the programme of study leading to the award of PhD, is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____

Md. Kabir Al Mamun

I.D. Number: 53157745

Date: _____

ACKNOWLEDEMENTS

There are many individuals that I would like to thank for their help and support throughout this project.

In particular, I would like to acknowledge the contributions of Dr. Tamas Szecsi (Supervisor) Senior Lecture in the School of Mechanical and Manufacturing Engineering at Dublin City University, for his unceasing enthusiasm, interest, and constructive criticism. His expertise, availability to discuss ideas and willingness to give his knowledge were instrumental in the completion of this thesis. I owe him much gratitude.

I would like to give special thanks to Professor M.S.J. Hashmi (Head of School) for his motivation and encouragement.

I would like to thank the academic staff for all help they gave me over the course of my work. I am especially grateful to Liam Dominican, Alan Meehan and Keith Hickey for their technical support and discussion throughout this work and the secretarial staff for their support. I wish to thank all of my fellow postgraduate students at Dublin City University and elsewhere, for their support and friendship; especial thanks to Dr. Mojahidul Hoque, Mr. M. Hasanuzzaman, Mr. Kamrul Hasan, Mr. Tony Marrero Barroso, Mr. Colin Griffin, Mr. Robert Sherry, Mr. Safayet Rayhan, Mr. Conor Carmody, Mr. Robert Sherry, A.S.M.S Arefin (TCD), Mr. Faisal Siddiqui, Mr. Istiaq Mahmud and Mrs. Halima Akter.

I am grateful to Dublin City University which was a partner of the EU funded FP6 project called Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery (iWARD) (www.iward.eu) and a partner of the EU-funded FP6 Innovative production Machines and Systems (I*PROMS) Network of Excellence (www.iproms.org), without whose support this project would not have been made possible.

Special thanks are due to my parents and family for their endless support during my studies, without which I may not have come to this stage. Sincere thanks to Mrs. Halima Akter (my wife) for her love, care and encouragement.

I would also like to dedicate this thesis to my Parents Dr. Momtaz Begum and Mr. Obayedul Kabir.

ABSTRACT

Development of situation recognition, environmental monitoring and patient condition monitoring service modules for hospital robots

Md. Kabir Al Mamun

An aging society and economic pressure have caused an increase in the patient-to-staff ratio leading to a reduction in healthcare quality. In order to combat the deficiencies in the delivery of patient healthcare, the European Commission in the FP6 scheme approved the financing of a research project for the development of an Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery (iWARD). Each iWARD robot contained a mobile, self-navigating platform and several modules attached to it to perform their specific tasks.

As part of the iWARD project, the research described in this thesis is interested to develop hospital robot modules which are able to perform the tasks of surveillance and patient monitoring in a hospital environment for four scenarios: Intruder detection, Patient behavioural analysis, Patient physical condition monitoring, and Environment monitoring. Since the Intruder detection and Patient behavioural analysis scenarios require the same equipment, they can be combined into one common physical module called Situation recognition module. The other two scenarios are to be served by their separate modules: Environment monitoring module and Patient condition monitoring module.

The situation recognition module uses non-intrusive machine vision-based concepts. The system includes an RGB video camera and a 3D laser sensor, which monitor the environment in order to detect an intruder, or a patient lying on the floor. The system deals with various image-processing and sensor fusion techniques.

The environment monitoring module monitors several parameters of the hospital environment: temperature, humidity and smoke.

The patient condition monitoring system remotely measures the following body conditions: body temperature, heart rate, respiratory rate, and others, using sensors attached to the patient's body.

The system algorithm and module software is implemented in C/C++ and uses the OpenCV image analysis and processing library and is successfully tested on Linux (Ubuntu) Platform. The outcome of this research has significant contribution to the robotics application area in the hospital environment.

TABLE OF CONTENTS

DECLARATION I
ACKNOWLEDEMENTS II
ABSTRACT III
TABLE OF CONTENTS IV
LIST OF FIGURES VII
LIST OF TABLES XIV
LIST OF APPENDIX XV

Chapter 1 **INTRODUCTION** 1

1.1 Background of the research project 2
1.2 Scenarios 3
1.2.1 Environmental monitoring 3
1.2.2 Intruder recognition 3
1.2.3 Detection of a patient lying on the floor 4
1.2.4 Patient physical condition monitoring 4
1.3 Structure of the thesis 5

Chapter 2 **LITERATURE SURVEY** 6

2.1 Introduction 7
2.2 Robots in healthcare 7
2.3 Scenarios 11
2.3.1 Environment monitoring 11
2.3.2 Situation recognition 13
2.3.3 Patient physical condition monitoring 15
2.3.4 Vision System 17
2.3.5 Human Detection 18
2.3.5.1 Application and Techniques 18
2.3.5.2 General architecture of human detection systems 21
2.3.5.2.1 Pre-processing 21
2.3. 5.2.2 Foreground Segmentation 23
2.3.5.2.3 Object Classification 26
2.3.5.2.4 Verification/Refinement 30
2.3.5.2.5 Tracking 31
2.3.6 Sensor integration and fusion 34
2.3.7 Summary 36

Chapter 3 **MODULE HARDWARE DESIGN** 38

3.1 Hardware Integration with Mobile Robot Platform 39
3.1.1 Mobile Robot Platform 40

3.1.2 Module box....42
3.1.3 Computing hardware....42
3.1.4 Power supply, data communication hardware....43
3.1.5 Sensor selection considerations....44
3.2 Environmental condition monitoring module....45
3.2.1 Sensor selection....46
3.2.1.1 Temperature Sensor....46
3.2.1.2 Humidity sensor....49
3.2.1.3 Smoke detector....50
3.2.2 Data acquisition hardware selection....52
3.3 Situation recognition module....55
3.3.1 Camera selection....58
3.3.2 Distance measurement hardware selection....64
3.3.3 Sensor selection....66
3.3.4 Experimental equipment....68
3.4 Patient physical condition monitoring....68
3.4.1 Sensor selection....69
3.4.2 Data collection....71

Chapter 4 **MODULE SOFTWARE DEVELOPMENT**....72

4.1 Introduction....73
4.2 Module Communications....73
4.2.1 Orca Interface....73
4.2.2 Module Interface Functions....75
4.2.2.1 Basic Module....75
4.2.2.2 Environment Monitoring Module....76
4.2.2.3 Situation Recognition Module....76
4.2.2.4 Patient Condition Monitoring Module....77
4.3 Environmental condition monitoring module....77
4.3.1 Driver software....77
4.3.2 Module algorithm....78
4.3.3 Module software....79
4.4 Situation recognition module....79
4.4.1 Driver software....80
4.4.1.1 Camera....80
4.4.1.2 PMD (3D) laser sensor....80
4.4.2 Face detection....81
4.4.2.1 Face Detection Using Haar-cascade and Region of interest Selection....82
4.4.2.2 Face Detection (Using Skin Colour Segmentation)....84
4.4.3 Badge detection....91
4.4.3.1 Contour analysis approach....93
4.4.3.2 Haar-Classifier approach....94
4.4.4 Sensor fusion....98
4.4.5 Intruder monitoring....101
4.4.5.1 Module algorithm....101
4.4.5.2 Module software....102
4.4.6 Monitoring patients on the floor....103
4.4.6.1 Module algorithm....103
4.4.6.2 Module software....104

4.4.6.3 Face detection and Colour segmentation....106
4.4.6.4 Badge detection....109
4.4. 6.5 Image Fusion and Background subtraction (Blob extraction)....109
4.4.6.6 Blob Image Analysis....112
4.4.6.7. Positive Sample Data preparation for ANN 113
4.4.6.8 Negative sample Data for ANN....124
4.4.6.9 Artificial Neural Network (ANN)....128
4.5 Patient condition monitoring module....130
4.5.1 Driver software....131
4.5.2 Module algorithm....131
4.5.3 Module software....133
4.5.3.1 Equvital Sensor Data Conversion....135

Chapter 5 **TEST RESULTS AND ANALYSIS**....137

5.1 Introduction 138
5.2 iWARD review meeting tests....138
5.3 iWARD hospital tests....139
5.4 Post-iWARD module tests....140
5.4.1 Environmental Condition Monitoring Module 140
5.4.2 Situation recognition module....144
5.4.2.1 Intruder monitoring..... 144
5.4.2.2 Recognising patients on the floor....151
5.4.3 Patient physical condition monitoring module 154
5.5 Plug-and-Play tests.... 156

Chapter 6 **CONCLUSIONS AND FUTURE WORK**....157

6.1 Thesis summary 158
6.2 Conclusions 159
6.3 Future work... 160

PUBLICATIONS ARISING FROM THIS WORK....162
REFERENCES....163

List of Abbreviations.... 196

List of Software.... 200

LIST OF FIGURES

Figure 2. 1: Robotics applications for hospitals per country 8

Figure 2. 2: The Care-o-bot (© Fraunhofer IPA) 8

Figure 2. 3: The uBOT-5. 9

Figure 2. 4: (a) Security robot and (b) Hospital robot for delivery 10

Figure 2. 5: Interceptor environmental monitoring system. 11

Figure 2. 6: Surveillance robot ideal for civil defence operations, developed by iRobot.....13

Figure 2. 7: Robotic platform for security guards with sensors marked. 14

Figure 2. 8: (a) Nexus – 4, (b) Procomp Infiniti 17

Figure 2. 9 : Hierarchy of templates used in the Chamfer System. 29

Figure 2. 10: (a), (b), (c), (d), and (e) Haar wavelets and Haar-like features, applied at specific positions of a pedestrian sample in. 29

Figure 2. 11: Histograms of oriented gradients in. (a) The descriptor block. (b) Block placed on a sample image. (c) and (d) HOG descriptor weighted by positive and negative SVM weights. 29

Figure 2. 12: Part-based classification using gradient-based features 29

Figure 2. 13: First five edgelet features selected by AdaBoost approached 29

Figure 2. 14: Pose invariant algorithm used in). (a) Input image. (b) Part-template detections. (c) Pose and shape segmentation. (d) Cells grid used for HOG computation. (e) HOGs. (f) Cells relevant to HOG. 29

Figure 3. 1: Functional modules integrated with a robot base. 41

Figure 3. 2: Installation of a module box on an iWARD robot. 42

Figure 3. 3: Gumstix tool kit. 43

Figure 3. 4: Back side of the Module drawer. 44

Figure 3. 5: Hospital condition monitoring system & hardware. 45

Figure 3. 6: Basic centigrade sensor used in the Environmental condition monitoring module. 48

Figure 3. 7: Humidity sensor 808H5V5. 50

Figure 3. 8: Interior of a basic ionisation smoke detector. 51

Figure 3. 9: Environmental monitoring module drawer for robot base. 52

Figure 3. 10: RSIOADXR RS-232 DAQ card. 54

Figure 3. 11: Sensor setup with a breadboard during the experimental stage. 54

Figure 3. 12: Sensors for motion detection and distance measurement. 55

Figure 3. 13: Sensors for face detection and image analysis. 56

Figure 3. 14: General graphical overview of the sensors of the vision system to recognise a patient lying on the floor. 56

Figure 3. 15: TV7240 IP camera. 58

Figure 3. 16: Phillips web camera. 59

Figure 3. 17: Canon VC-4 PTZ camera. 60

Figure 3. 18: Axis 214 PTZ IP camera. 60

Figure 3. 19: (a) Images of a chessboard being held at various orientations (b) chessboard used for camera calibration in this research. 61

Figure 3. 20: Camera viewing angle calculations. 63

Figure 3. 21: O3D100 PMD 3D laser sensor. 66

Figure 3. 22: KC7786 PIR sensor module. 67

Figure 3. 23: Situation recognition module (the computing device is inside the drawer). 67

Figure 3. 24: Laser sensor and video camera assembled on an experimental rig. 68

Figure 3. 25: Patient condition monitoring system & hardware. 69

Figure 3. 26: EQ-01 Sensor Electronics Module (SEM). 71

Figure 3. 27: EQ-01 Monitoring Belt and EQ-01 Sensor Electronics Module (SEM). 71

Figure 4. 1: Orca2 interface. 74

Figure 4. 2: Flow diagram of the environmental condition monitoring module software algorithm. 78

Figure 4. 3: Input Images of different real time scenes (a), (b) and (c) frontal face, (d) profile face. 82

Figure 4. 4: Face detected and ROI selected for the input images shown in Figure 4.3. 84

Figure 4. 5: Value components (from the HSV colour space) of the input images from Figure 4.3-‘c’ and ‘d’. 89

Figure 4. 6: Detected skin blobs (binary images) of the input images (from Figure 4.3-‘c’ and ‘d’) 89

Figure 4. 7: Detected skin colour mask (RGB) of the input images (from Figure 4.3-‘c’ and ‘d’) 90

Figure 4. 8: General overview of the colour segmentation method 91

Figure 4. 9: Grey and equalised histogram-images for the input images. 92

Figure 4. 10: Black and White (binary threshold) image for the input images (from Figure 4.3). 92

Figure 4. 11: Detected Badge Image for the input images shown in Figure 4.3; (a, b, d) contains detected badge contours (c) no badge found. 94

Figure 4. 12: (a) Badge sample image (b) Background image (contains no badge). 95

Figure 4. 13: Output of the performance utility of the badge detection classifier. 96
Figure 4. 14: Badge detection using a classifier; (a, b, d) Badge detected and (c) No badge detected. 97

Figure 4. 15: (a) Video (RGB) input image, (b) distance value converted to Intensity image (PMD), (c) distance value displayed with a colour map & (d) perspective view of the 3D scenes. 100

Figure 4. 16: Distance value displayed with colour map and index. A patient is lying on the floor and his image is vertically projecting the real time horizontal scene. 'A' marks the black hair of the human model. 100

Figure 4. 17: Flow diagram (starts from blue blocks) of the Intruder monitoring module software algorithm. 101

Figure 4. 18: Flow diagram (starts from blue blocks) of the detection of patient lying on the module software algorithm. 104

Figure 4. 19: Face and badge centre line orientation relative to the x-axis of the image frame. 105

Figure 4. 20: Binary image of the segmented dress colour of the input image in Figure 4.3- 'c' and 'd'. 107

Figure 4. 21: Colour mask image of the segmented blob of the input image in Figure 4.3-'c' and 'd'. 108

Figure 4. 22: Segmented binary blob of the full body (target object), where input images are from Figure 4.3-'c' and 'd'. 108

Figure 4. 23: Segmented colour blob of the full body (target object), where input images are from Figure 4.3-'c' and 'd'. 109

Figure 4. 24: Image fusion: (a) Original camera image, (b) Original laser (PMD 3D0 image, (c) Stretched camera image and (d) Perspective warp image. 110

Figure 4. 25: Resized laser intensity image for two different orientations of the lying person, presented in figure 4.3-'c' and 'd'. 111

Figure 4. 26: Resized camera image of binary blob to match the original size (64x50)of the PMD image for two different orientations of the lying person, presented in figure 4.3-‘c’ and ‘d’. 111
Figure 4. 27: Resized (640x480) laser image for two different orientations of the lying person, presented in figure 4.3-‘c’and ‘d’ after applying distance threshold. 112

Figure 4. 28: Projected image (fusion of colour and distance) for two different orientations of the lying person, presented in Figure 4.2-‘c’ and ‘d’. 112

Figure 4. 29: Filtered blob contours of the input images presented in Figure 4.3-‘c’ and ‘d’. 114

Figure 4. 30: (a) Freeman chain moves are numbered 0–7; (b) contour converted to a Freeman chain-code representation starting from the back bumper. 115

Figure 4. 31: Three differently shaped rectangular objects (a, b and c) with identical Freeman chain code histograms (d, e and f). 116

Figure 4. 32: (a) Contour of the rectangular shape object, (b) DFCC (including 'same directions' and 'different' directions), (c) DFCC (excluding changing in the same direction) and (d) DFCC (sums of all left and right turns in a weighted manner). 117

Figure 4. 33: (a) Contour of the ‘L’ shape object, (b) DFCC (with 'same' and 'different' directions), (c) DFCC (excluded changing in same direction) and (d) DFCC (sums of all left and all right turns in a weighted manner). 118

Figure 4. 34: (a) Contour of the ‘Stairs’ shape object, (b) DFCC (with 'same' and 'different' directions), (c) DFCC (excluded changing in same direction) and (d) DFCC (sums of all left and all right turns in a weighted manner). 118

Figure 4. 35: (a) and (b) extracted features for the patient lying on the floor detection process, (c) and (d) shows the convexity defects histograms for the images (a) and (b). In image (a), ‘u’ and ‘v’ indicate the start and the end points of the i^{th}-convexity defect; the depth d is the distance from the farthermost point of the convexity defect to the convex hull segment. 122

Figure 4. 36: HS Histogram for input image (from Figure 4.23). Both histograms are the same

due to the same skin and dress colour of the two input images. 123

Figure 4. 37: Raw (RGB) input images (Left: Human sample) and (Right: Non-Human sample). 125
Figure 4. 38: (a) and (b) B&W blob image segment, (c) and (d) Colour blob segment, (e) and (f) the extracted contour sequence of the input images in Figure 4.37. 125

Figure 4. 39: Contour-related (FCC and DFCC) Histograms. 126

Figure 4. 40: (a) and (b) Convexity defects of the input image blob contours (Figure 4.38), (c) and (d) convexity defects histogram of the respective images (a) and (b). 127

Figure 4. 41: The hue and saturation histogram of the obtained colour mask segment..... 127

Figure 4. 42: General diagram of the Neural Network model. The system uses a feed forward neural network with three layers. The input layer is composed by one set of input neurons for each used descriptor (FCC, DFCC etc.). 129

Figure 4. 43: Flow chart of the patient condition monitoring module. 132

Figure 4. 44: VSDS Viewer 134

Figure 4. 45: Sensor electronic module Viewer 135
Figure 4. 46: A normal ECG waveform 136

Figure 5. 1: Testing of the iWARD robots during the Final review meeting (February 2010, San Sebastian) 139

Figure 5. 2: Testing of the iWARD robots during the first and second Evaluation meetings (October 2009, Newcastle, and November 2009, San Sebastian). 140

Figure 5. 3: GUI screen shot of the Environmental monitoring module testing; (a) Results show temperature readings with ‘no detection’ of smoke (b) temperature readings with detection of smoke (smoke created under a controlled environment). 141

Figure 5. 4: Sample graph of temperature sensor readings with their respective time stamps. 142
Figure 5. 5: Sample graph of humidity sensor readings with their respective time stamps. . 142

Figure 5. 6: GUI of the Intruder detection scenario. 144

Figure 5. 7: GUI of the Intruder detection scenario shows point clouds of the target object projected in 3D space. 144

Figure 5. 8: Target frame containing a human with a badge. 145

Figure 5. 9: False detection of a face and incorrect ROI labelling using the haar face detection classifier. 146

Figure 5. 10: (a) Target frame containing a human and a badge (camera to badge distance is 1.5 m, good lighting condition), (b) face detection and ROI labelling, (c) Detected badge using the developed classifier and (d) Detected badge using contour analysis. 146

Figure 5. 11: (a) Target frame containing a human and a badge, (b) face detection and ROI labelling and (c) Detected badge using contour analysis. 147

Figure 5. 12: (a), (b) Target frame containing a badge rotated at ±50º relative to the horizontal axis and (c), (d) Detected badge using contour analysis. 148

Figure 5. 13: Schematic diagram for extreme badge positioning relative to the camera. 148

Figure 5. 14: (a), (b) Target frame containing a face and a badge at extreme left and right of the camera FOV, (c), (d) Detected badge using the developed badge classifier and (e), (f) Detected badge using contour analysis. 149

Figure 5. 15: (a) Target frame containing multiple persons and badges at extreme left and right of the camera FOV, (b) Face detection and ROI labelling (multiple), (c) Detected badge using the proposed badge classifier and (d) Detected badges using contour analysis. 150

Figure 5. 16: ANN Training error rate per iteration. 151

Figure 5. 17: Examples of positive blob samples (patient lying on the floor). 152

Figure 5. 18: Examples of negative blob samples (no patient lying on the floor) (a, b, c) Non-target class human objects and (d, e, f, g, h) Non-target static class objects. 153

Figure 5. 19: Example of a graph of heart rate. 155

Figure 5. 20: Example of a graph of respiration rate. 155

Figure 5. 21: Example of a graph of skin temperature. 156

LIST OF TABLES

Table 2. 1: Different Learning Algorithms Used in Human Detection System. 27
Table 2. 2: Different Tracking Approaches Used in Human Detection System 33
Table 3. 1: shows typical calibration results for the Phillips USB camera generated in YML format during the experimental stage. 62
Table 4. 1**:** Comparison between raw RGB and Equalised image in various lighting conditions.....86
Table 4. 2 Histogram matching results (H1 = Histogram of low lighting condition, H2 = Histogram of moderate lighting condition and H3 = Histogram of Good lighting condition)..... 87
Table 4. 3**:** BGR values comparison in various lighting condition..... 88
Table 4. 4**:** FCC and DFCC characteristic values for Figure 4.30. 119
Table 4. 5**:** Different chain codes histogram for different positive samples. 120
Table 4. 6: Different ANN descriptor values for training samples.....124
Table 4. 7: Different ANN parameter values for target example (no patient lying on the floor) samples.....128
Table 4. 8: Interface application Layer 131
Table 4. 9: Sample hexadecimal data converted into ASCII 135
Table 5. 1: Data file 'Environment_Normal_Values.txt'.....141
Table 5. 2: Example Log File.....143
Table 5. 3: Results of recalling the ANN. 154
Table 5. 1: Sample of the Equvital sensor data logging chart.....155

LIST OF APPENDIX

Appendix A (Software and Experimental Results)…….……..i

Orca interface definition for the basic module ii
Orca interface definition for the Environmental condition monitoring module iv
Orca interface definition for the situation recognition module.....vi
Situation recognition (Scenario: intruder monitoring) module Log file ix
Artificial Neural Network (ANN) PARAMETERS.TXT file.....x
Table A-1: ANN results of recalling the Test samples xi
Table A-2: Equvital Sensor data conversion (Hexadecimal to ASCII).....xii

Chapter 1

INTRODUCTION

1.1 Background of the research project

Using robots in hospital environments for assisting nurses in their everyday tasks is not uncommon in some developed countries like Japan and Canada. It is now aimed to bring these ideas into Europe and put robots into European hospitals and healthcare centres to overcome the shortages of healthcare staff. The shortage of healthcare staff has been a big issue over the past decade and without the aid of robots hospitals would simply be overcrowded with patients and with stressed out workers. As a result, this would lead to errors occurring, patients waiting too long for assistance and a lack of hygiene throughout the hospitals. As time progresses this problem would not alleviate itself but rather increase the problems that are already occurring.

At the end of 2006, DCU teamed up with nine other European research centres, universities and hospitals and together founded the iWARD consortium, which aimed to develop a hospital robot swarm system. iWARD stands for 'Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery'. The research project was financed by the EU FP6 scheme. The aim of the project was to develop a prototype of the system containing several robots and a realistic hospital model by the end of 2009.

The five major tasks of the robot system were cleaning, surveillance, guidance, delivery and patient monitoring. The whole system was created as the result of several work packages, including the development of the robot platforms, functional modules, software platform, navigation system, communication system, human-robot interface, swarm application software, and others. While researchers in the other partner organisations were developing the robot base and its control system, the human–robot interface, the software platform and the swarm application software, the work package for the development of the functional modules for the robots was led by the principal investigator from DCU, Dr. Tamas Szecsi, a senior lecturer at the School of Mechanical and Manufacturing Engineering. Each functional module was developed as a stand-alone, self-contained, interchangeable unit that had a common interface to the mobile, self-navigating robot platforms.

After the successive testing of the prototype system in two hospitals and completion of the iWARD project in February 2010, the DCU team continued the development of the functional modules in order to enhance their functionality and make them more robust. The material

described in this thesis covers the development of three functional modules both during and after the iWARD project.

1.2 Scenarios

Out of the all the scenarios that were implemented in the iWARD project, this thesis deals with four. They are explained in the section below.

1.2.1 Environmental monitoring

In this scenario the moving robot equipped with sensors performs routine monitoring of the parameters of the hospital environment: ambient temperature, humidity, smoke/fire. The robot cyclically accesses the sensors' readings and based on the current location of the robot it evaluates the sensor data using the pre defined database. The database contains the allowable ranges of the environmental parameters for each location. The shared knowledge of the robot system is updated with the sensor readings. If smoke/fire is detected an alarm signal is generated and event is raised that informs the robot system. If the temperature and/or humidity/temperature level is out of range the module raises a corresponding event that is evaluated by the robot system. The nurses' station is also informed so that a staff member can take action. Environmental monitoring should be active all the time.

1.2.2 Intruder recognition

In this scenario, the task of the robot is to recognise a potential intruder in the hospital. When the robot detects motion in its vicinity it attempts to decide whether the source of the motion is a person who is entitled to be in the hospital (a staff member or a patient), or it is a potential intruder. If a potential intruder is detected an event is raised which is analysed by the robot system. The images of the person are passed on to security for further action. If a potential intruder gets too close to the robot another event is raised due to the possibility of the person attempting to damage the robot itself.

Since it would be extremely difficult to automatically perform such task in a crowded environment, intruder recognition is only performed night time.

1.2.3 Detection of a patient lying on the floor

In a hospital environment it could happen that a patient has fallen on the floor either consciously or unconsciously. Within this scenario, the robot, why patrolling the hospital, analyses the readings of sensors and attempts to detect a person on the floor. If the robot finds such a person it raises an event that is analysed by the robot system. The nurses' station is also informed in order to take action.

Since it is highly unlikely that during daytime the robot can detect a fallen person before he/she is given help, this type of detection is only required during night time when there is only a limited presence of staff and patients in corridors and other areas.

1.2.4 Patient physical condition monitoring

In this scenario the robot remotely collects information about the physical condition of a patient, including body temperature, heart rate, respiratory rate, and others. The data is stored in the shared knowledge of the robot system and can be used by staff members. If a patient's condition is critical the robot raises an event and informs the robot system and the nurses' station.
This scenario can be performed either routinely, while the robot is patrolling, for each patient in the vicinity of the robot, or the robot can be ordered to take measurements from a specific patient.

In order to assist hospital staff, the objective of this research is to develop three robot service modules as follows:

(i) Environmental Condition Monitoring
(ii) Situation Recognition (including Intruder Monitoring and Recognising Patients on the floor)
(iii) Patient Physical condition Monitoring

1.3 Structure of the thesis

The rest of the report is divided into a number of chapters.

Chapter Two is a review of literature relevant to this research. It first describes the state-of-the-art of different service robots and their applications, followed by a brief description of module sensors based on the scenarios of the research. Then a detailed description of different research results and techniques on human detection follows, as that is the core component of this research. A critical analysis of existing solutions is provided, together with the requirements for an improved system.

Chapter Three describes the hardware development and sensor selection used in this research. It includes the specification and design of the functional modules. It also contains details about hardware integration with the robot base utilising a “Plug and Play' mechanism.

Chapter Four describes the algorithms and software specifications and development for each individual module. The operation of each module is explained with a flow chart of the system. At the end of this chapter, details of the developed driver software and Orca interface functions are also included.

In Chapter Five results from test of each module are presented and discussed.

Finally, Chapter Six summarises the major contribution from the results of this research and gives recommendations for future work in the researched area.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

With the development of robotics technologies, the application of robots has been extended more and more widely, i.e. in manufacturing cars, industrial assembly lines, health-care, and so on. "The robot is required not only to complete some large-scale assemblage tasks in work, but also to achieve complex missions in various environments" [1]. However, the research described in this thesis is interested in health-care robots and hospital robots which are able to perform the scenarios listed in the previous chapter. A detailed overview of the background and modules related to this research is outlined in [2] and is described in the following chapter. This chapter presents the state-of-the-art of solutions related to this thesis.

2.2 Robots in healthcare

The impact of robotics applications in healthcare could be huge in terms of health, societal and economic benefits. Without compromising the quality of care, robotics applications could offer the promise of sustainable and affordable health provision. Some products are already available, like the surgical robot Da Vinci. Clear roadmaps are required for the scale of research and development still needed to transform the challenges that exist – technological, financial, ethical, and social – into practical and beneficial solutions [3]. Figure 2.1 shows a graphical overview of the robotics applications per country based on a European commission research project [4] where the USA is in top position and Europe still has open scope in this field.

Presently, doctors and nurses can look after their patients using monitoring devices. This is especially valuable for remote monitoring when the patient is at home. However, when unusual situations are detected, it is often difficult to tell from a distance how urgent the situation is. Robotics applications will enhance communication and identification of alarming situations. The future will see more remote monitoring and this will help care to become more affordable and efficient. Figure 2.2 shows a mobile robot assistant Care-O-bot® [3] that is able to assist humans in their daily life.

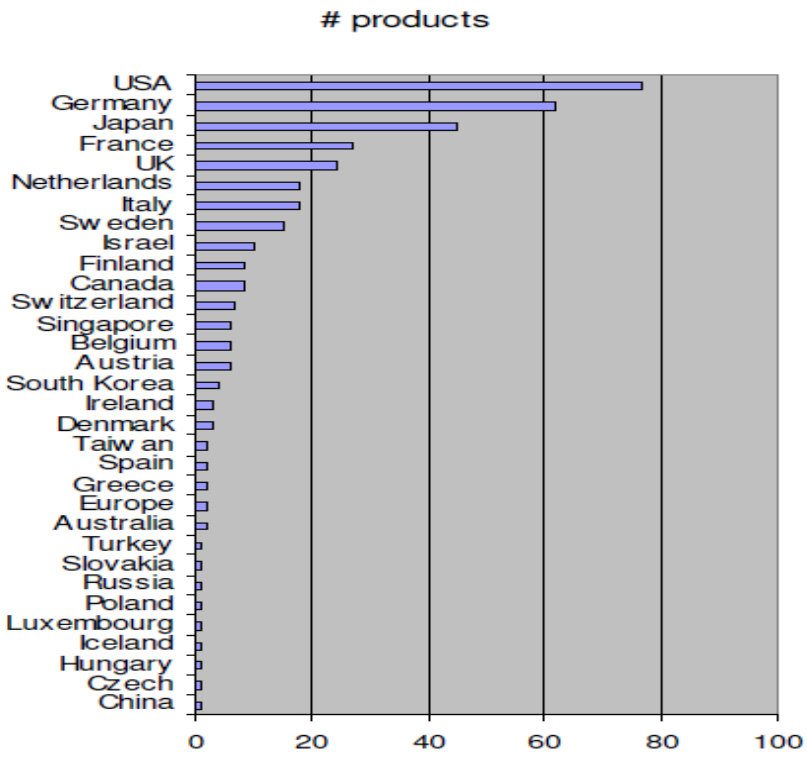


Figure 2. 1: Robotics applications for hospitals per country [4].

Figure 2. 2: The Care-o-bot (© Fraunhofer IPA) [3].

The Skillgent Robot Learning and Behaviour Control System is a complete control system for autonomous service robots, including a behaviour control and coordination system, task and skill learning functions, a powerful computer vision system, a social human-to-machine interface, a database for storing knowledge and other integrated components [5]. The

Skillgent Robot is able to perform several tasks, mainly patient condition monitoring, delivery, remote consultation, guidance, environment monitoring, and surveillance system and so on [5].

Self-navigating mobile robots can be used in many hospital/medical applications: transport in the X-ray region, chemical gas detection system of toxic residues or hazard points, remote inspection for detection of high temperature and radiation, aided-radiotherapy [6,7,8]. Shieh proposed a design concept of an intelligent hospital service robot (IHSR) aimed at saving human resources and also improving the hospital services [9]. Takahashi [10] has been developing the MKR (Muratec Keio Robot), an autonomous Omni-directional mobile transfer robot system for hospital environment to support a delivery system. Rafflin and Fournier [11] presented FIRST, a Friendly Interactive Robot for Service Tasks, designed to carry heavy loads in hospitals.

The uBOT-5 [12] is the result of project ASSIST, which can quietly follow its owner around the house, take care of the cleaning, give reminders about medication, help with shopping and make communication with doctors. The uBOT-5 also has the ability to recognise when its owner has fallen or become unresponsive, allowing it to immediately dial 911 for medical attention and relay important information to caregivers en route [13]. This robot is designed to work in home environment, instead of in a hospital environment. Figure 2.3 shows an image of uBot-5.

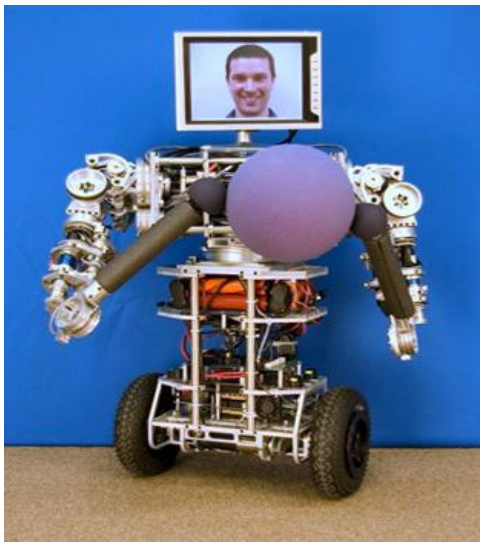


Figure 2. 3: The uBOT-5 [12].

Mobilerobots [14] commercially offers different types of robots for different applications and some of them are for hospital environment. Although they offer security and delivery robots for hospital environment, they are separate robots which cost more. Figure 2.4 (a) and (b) show a security robot and a hospital robot for delivery application respectively.

(a)   (b)

Figure 2. 4: (a) Security robot and (b) Hospital robot for delivery [14].

To date, researchers have developed numerous approaches to collaborative robot control to solve different application problems. Cao et al. [15] deserve credit for attempting the categorisation of different architectures in the cooperative, multi-robot domain. There has been a decade and a half of research aimed at providing robots with a behavioural or task-based decomposition of the control system since Brooks [16] introduced it as an alternative to the traditional functional approach. The application of behaviour-based robotics to groups of robots has been explored extensively in the latter half of the 1990s (e.g., [17]) and architectures for cooperative control have recently been introduced [18]. Much work is focused on how different behaviours have to be combined to achieve a desired effect that solves a given problem. Proposed solutions include hard-coded or learned weights [19]. Attempts have also been made to analytically determine the minimum information requirements for solving a task [20] and for automatic generation of robot teams [21]. However, most of these systems have not tackled the problems of distributed collaborative behaviours and distribution of resources across robots.

2.3 Scenarios

The following section includes scenario-based literature survey of existing solution and systems.

2.3.1 Environment monitoring

Laboratories, blood banks, pharmaceuticals, blood products and general hospital environment all require temperature and humidity to be maintained at constant levels to meet health, storage and quality requirements. Such systems which monitor the environmental condition parameters (i.e. temperature, humidity and smoke/fire event) are not a new idea. But functional modules based as stand-alone, self-contained, interchangeable units which have a common interface to mobile, self-navigating robot platforms are not commonly available systems. The module-based environmental monitoring system must provide dynamic data compared to the static, traditional approaches. The Skillgent robot [5] is capable of monitoring environmental conditions along with other tasks, but it is not a standalone module, which is one of the major features of the present study.

The Jacarta interceptor[22] is designed to protect data centres, server rooms and racks by preventing computer equipment damage and data loss due to over-heating, excess moisture, water ingress, fire, power failures, security breaches, and many other potentially catastrophic environmental threats. Figure 2.5 shows one of the environmental monitoring systems designed by [22]. Whereas DICKSONDATA [23] provides a wide range of environmental monitoring devices specially designed for hospital environment applications.

Figure 2. 5: Interceptor environmental monitoring system [22].

In fact, there are many companies [mentioning just a few of them 22, 23,24,25] that are commercially producing environmental monitoring systems for different application areas with different ranges of features; but modular systems are still very uncommon. However, a list of different monitoring devices used in hospital environment is reported in [23].

Review of monitoring systems and sensors:

There is no one-size-fits-all solution for environmental monitoring. In fact, there are too many data loggers, chart recorders, thermometers and wireless monitoring instruments available off the shelf, but only a small number of these are best match systems for any particular hospital environment. A few things need to be considered [26] when developing such a system to fit the specific application.

i. The regulatory requirements for monitoring temperature and/or humidity vary from one application to another.

ii. How the differences in staff patterns and staff responsibilities from one department to another make one or another "optional" feature (e.g. ability to transfer data with USB stick or the ability to show a continuous display of logger readings on computer monitor at all times) more relevant to successful environmental monitoring in that particular hospital department.

iii. Upfront costs of monitoring instruments need to be weighed against the ongoing costs for operating these instruments, especially in terms of labour time requirements and associated costs.

iv. Ease-of-use features need to be considered as ways to ensure that monitoring procedures are adhered to, and that use of the instruments is within reach of the skill set of department staff members.

However, it is clear that each of the environmental monitoring systems needs the relevant sensors to get the data and also needs a data acquisition device (DAQ) to convert the analogue data to digital. For example smoke, temperature and humidity sensors with a DAQ device for this research application. Hence this research analyses a wide range of sensors which is summarised in the next chapter.

2.3.2 Situation recognition

As stated earlier, there are two types of situation recognition in this research: intruder monitoring and recognition of patients on the floor. Since to-date there have been no dedicated systems for the second task, especially in a hospital environment, the scenario-based background research is limited to intruder monitoring. However, the 'patient on the floor' scenario will be reviewed later as part of the general human recognition techniques.

Surveillance (Intruder monitoring)

The development of fully automated surveillance systems based on mobile multifunctional robots is an active research area [27,28,29,30].With the traditional approach, using CCTV systems, surveillance monitoring is also not a new concept. But again, functional modules based as a stand-alone, self-contained, interchangeable unit which has a common interface to the mobile, self-navigating robot platform is not a common system. The module-based intruder monitoring system must provide dynamic data compared to the static traditional approaches. Robotic research is in a mature state and ready to focus on complete mobile robotics applications. The Skillgent robot [5] is capable of performing surveillance monitoring along with other task, but still that is not a standalone module which is another major feature of this research. There are several other companies (namely Mobilerobots [14] and iRobot [31]) that design robot platforms and systems (Figure 2.6) which can be used for surveillance monitoring purposes in various environments.

Figure 2. 6: Surveillance robot ideal for civil defence operations, developed by iRobot [31].

The research in the AASS Learning Systems Lab aimed at building a Robotic Security Guard [32] for remote surveillance of indoor environments. Biber [33] reported that their research relates how a robot will learn to patrol a given environment, acquire and update maps, keep watch over valuable objects, recognize known persons, discriminate intruders from known persons, and provide remote human operators with a detailed sensory analysis. The design concept is based on augmenting remote human perception with superhuman sensory capabilities, as shown in Figure 2.7.

Figure 2. 7: Robotic platform for security guards with sensors marked [32].

Review of module sensors

Both sub-modules (Intruder detection and Patient on the floor detection) need to detect humans, which is a major task and is categorised as a vision system. Garcia [34], Miaou [35], NaitCharif [36], Girondel [37], and Boulay [38] present different methods to recognise a human body by analysing image data. However, the iWARD research application environment is different from the above reported work: the input image background and foreground are dynamic. Vision sensors can provide more information for a mobile robot about the environment than most of other sensors, and then it is attracting more and more attentions [39]. As vision is recognised as an essential sensing function for robots, various

types of special and general purpose image processing hardware have been developed to improve the performance of visual object sensing [40,41,42].

Time of flight (TOF) cameras can be used as position sensing devices in robotic applications. Among the most important sensors in autonomous mobile robotics are the ones for distance measurement, specifically to measure the distance to obstacles around for navigation. Sonar, laser, and infrared sensors are heavily used for this purpose. They are active sensors in that they emit a signal and calculate the distance to the nearby objects by processing the reflected signal. A compass can be used to determine the robot‘s absolute orientation. A further step could be the interfacing to a receiver module for the satellite-based global positioning system (GPS), which is very useful to get global knowledge about the world, but it is complex and can only work outdoors in unobstructed areas [43]. Various camera-like techniques for measuring distances between objects and a sensor’s location exist in literature. Prominent examples based on triangulation and/or image correspondences are laser triangulation, and structured light and stereo vision [44,45]. Time-of-flight (TOF) sensors, on the other hand, compute the distance to an object by estimating the elapsed time between the emissions of light from an active illumination to the arrival at the pixel of a sensor chip. Relatively new camera-like approaches, like the Photonic Mixing Device (PMD) [46,47] and the Z-Cam [48], are still being developed and only a few prototypes are available. However, a camera and a distance measurement device are the root sensors for this module. More details on camera and distance measuring device selection are reported later in the next chapter.

2.3.3 Patient physical condition monitoring

The Care-o-bot [3] and Skillgent Robot [5] are capable of monitoring patient physical condition remotely, although none of them have the standalone modular feature. They are designed with built in features. On the other hand, there are a large number of systems to monitor patient physical condition (i.e. vital signs) available commercially. They are not integrated with autonomous mobile platforms.

As part of the iWARD project, a remote physical condition monitoring system (stand alone, modular, plug and play with autonomous robot platform), otherwise known as a telemetry device, was needed to monitor patients for the following: body temperature, heart rate,

respiratory rate, and so on. Since this project was developed by a group of scientists and engineers, more information was needed to understand what other physiological signs were needed to be monitored. Some of the nurses from the School of Nursing in DCU suggested heart rate, temperature, respiration, pulse rate, blood pressure and electroencephalography (EEG) for monitoring the brain.

According to Barbara [49], the four vital signs of the body are body temperature, pulse, respiration rate and blood pressure. Blood pressure can only be measured with a separate device containing a sphygmomanometer, a blood pressure cuff and a stethoscope, and this may cause discomfort to the patient when used on a regular basis. With this in mind it was very clear to distinguish what is considered important for monitoring patients on a regular basis and did not limit the products that were researched. On failing to find a device that could monitor body temperature or body motion meant that a monitoring system would have to be engineered from conception, using raw materials, and would result in time and money being wasted in developing and testing it. As a result, this may end up with some technical failures and delay the development of the project.

Hence this study carried out a survey to select a product or system which is available on the market that can sense the vital sign of the patient. In fact there are quite a few numbers of companies that are commercially producing physical condition monitoring systems for different application areas with different ranges of features. The Infinity Omega-S integrates an Infinity Kappa monitor, which provides real time vital signs monitoring at the point of car [.50].The EQ-01 sensor unit [51] features a system of fabric-embedded, intelligent sensors to measure medical grade physiology in real time. The sensor unit is a non – invasive, remote physiological welfare monitoring system designed to protect the lives of personnel operating in hazardous or harsh environments such as the military, emergency services and even sports professionals. It can also be used for telemedicine and if used correctly can actually be used for clinical monitoring. The NeXus-4 [52], Figure 2.8 (a), is small and light and can supports 4 input channels for: Electroencephalography (EEG) (to measure the brain), Electromyography (EMG) (to measure the muscles), Electrocardiogram (ECG/EKG) (to measure the heart), Electrooculography (EOG) (to measure the eye), Blood Volume Pulse (BVP), skin conductance, respiration, skin temperature and heart rate (HR) and sends the information wirelessly via Bluetooth.

(a) 

 (b)

Figure 2. 8: (a) Nexus – 4 [52], (b) Procomp Infiniti [53]

The ProComp Infiniti [53], Figure 2.8 (b), is a small, light, non – invasive telemetry device that can monitor up to eight body physiologies. It is designed for a hospital environment and can transmit the data in real time via Bluetooth. The first two sensor channels provide signals for viewing RAW EEG, EMG and EKG/ECG signals. The remaining six channels can be used with any combination of sensors, including EEG, EKG/ECG, RMS EMG, skin conductance, heart rate, blood volume pulse, respiration, goniometry, force, and voltage input. The next chapter includes more details on sensor selection.

2.3.4 Vision System

Cameras are probably the most complex sensors utilised in robotics applications. They have not been used extensively in robotics until recently due to high computational and memory requirements, as well as high costs. Extensive research is going on to incorporate vision into robotics, due to its ability to supply the robot and/or a system with rich information about its environment, which may not be provided even by combining several other sensors. Current robots are using either global or local vision. In global vision, the camera is external to the robot, placed in such a way that it can see a region large enough to enclose both the robot and its environment. In local vision, the robot has its own camera doing all the vision processing onboard. The latter approach is more difficult to implement especially for small robots due to space, memory, and processing power constraints. However, it is natural to have visual

perception onboard, since no animal or human has global vision; rather, every living being has its own sensors.

Therefore, the trend is to have local vision on the robots, which will get easier with computing cost being embedded in them. On the other hand, global vision is much easier to realise, and also has some implementations, such as in-factory environments.

Combining ToF camera with RGB camera (video camera) is the recent trend in robotics applications (i.e. vision systems, object recognition) [54].

2.3.5 Human Detection

Human detection is one of the key roles to an increasing number of applications that runs in human inhabited environments or needs to interact with a human (i.e. user). As an example, cars provided with pedestrian protection systems require some type of human detection capability [55]. In the same way, a self navigated robot, such as a social robot [56], needs to detect humans to perform tasks like aid for rehabilitation in hospitals [57], assistance in offices [58], or guidance in museums [59].
Both people detection, tracking and human pose estimation has a large variety of applications (i.e. automotive safety, surveillance, video indexing, virtual reality, intelligent control, human-computer interaction and model-based coding and so on). For these applications, a system first needs to detect the objects of interest and then track them across different frames while maintaining the correct identities. The two major difficulties in performing these tasks are: (a) changes in appearance of the objects with viewpoint, illumination and clothing, and (b) partial occlusion of objects of interest by other objects.

2.3.5.1 Application and Techniques

There is a popular technique, histogram of oriented gradients (HOG) in human detection. Due to be one of the first papers, Dalal [60] is remarkable which reports impressive results on human detection. HOG used in [60] as low-level features, which were shown to outperform features such as wavelets [61], PCA-SIFT [62] and shape contexts [63]. Detection speed is another important issue. Zhu [64] proposes a rejection cascade using HOG features to improve it. Zhang [65] propose a multi resolution framework to reduce the computational

cost. Begard [66] reports the problem of real-time pedestrian detection by considering different implementations of the Adaboost algorithm (**AdaBoost**, short for Adaptive Boosting, is a machine learning algorithm). Detection is a pre-requisite for tracking in general. Tracking by detection has been a focus of recent work [67,68,69].

Human detection and tracking system development has been going forwards for several years. Many real-time systems have been developed [70,71,72,73,74]. However, there are still several challenging technologies that need more research: foreground segmentation and false alarm elimination. Skin colour is one of many cues which are used to segment foreground in some applications (i.e. human hand detection). Skin colour has been utilised in combination with other cues to obtain better performance [75].

Foreground segmentation techniques include optical flow computation [76] and background subtraction. Although optical flow computation can provide better performance, it is a computationally expensive method and not suitable for real time systems [77]. However, Tian and Hampapur combine these two techniques [77] to solve the problem of computation consuming. Initially the background subtraction technique is used to locate the motion area, and then applied the optical flow computation only on the motion area to avoid false foreground pixels. Generally, the motion information is extracted by thresholding the difference between the current image and the background image. The background can be modelled as Gaussian distribution N (μ; 2σ); this basic Gaussian model (GM) can adapt to gradual light change by recursively updating the model using an adaptive filter [72]. However, this GM will fail to handle dynamic backgrounds, such as water wave and tree shaking, or more particularly to the present research scope (i.e. when the vision system is mobile which creates more complex dynamic scenes for both the background and the foreground). To solve the problem of multiple backgrounds, the models such as the mixture of Gaussian (GMM) [78], Nonparametric Kernel [79], and codebook [80] are provided recently. Although these algorithms are effective for modelling multiple backgrounds, they require more memory and more computation.

Human recognition problems can be solved using two relatively new techniques: neural networks [81] and model-based solutions [71,82]. The neural network is a powerful tool for pattern recognition. In [81], the BP network was used to recognize a pedestrian. Model-based

human recognition analyses the shape of objects and distinguishes a human from other objects. In order to track people, the human model has to be created which includes the human features, such as the colour, aspect ratio, edge, velocity etc. Occlusion is a significant problem in human tracking. In order to solve the problem of occlusion, Kalman filter-based method [83] and appearance-based tracking method [84,85] were proposed. To evaluate the performance of the system, Erden provides a new method [86]. Most of the algorithms developed in previous works were based on RGB colour space while other type of colour spaces (like HSV) give better detection results in the colour based detection systems (i.e. skin colour detection) [87]. Furthermore, there is an extensive list of works dedicated to the problem of human detection [88].

When the platform is mobile, however, as in the case of a mobile robot, none of the above mentioned background subtraction methods can be applied. An important aspect of human detection is that the system should be able to detect a human under different poses; however, most current solutions still rely on detecting human faces [89,90,91,92]. This has the disadvantage that a human can only be detected when is facing the camera. In particular, in the case of a mobile robot this leads to a loss of several social aspects. For instance, if the robot wants to initiate a conversation, the user has to already be paying attention to it, which is not always the case. Similar problems might arise in the case of a pedestrian detection system where the human can be in any pose with respect to the detecting system. In this way, although important advances have been achieved in the development of algorithms for human detection, there is still space for further improvements, particularly in the case of a mobile platform that needs to detect humans under different poses [93]. Furthermore, many applications also require real time operation, stressing the need for an efficient human detection system that can provide a timely detection.

The state-of-the-art in human detection systems can be divided into two main categories: i) methods that require background subtraction as first step to detect the interesting objects. ii) methods that perform the detection using uncontrolled moving cameras. This research method belongs to the last category; hence, this research concentrates on the review in methods that do not rely on background subtraction techniques. For a more extensive review refer to [88,94]. Gavrila proposed a pedestrian protection system for moving vehicles [95,96]. In those works, human bodies are detected using a shape-based method known as the

Chamfer system. Every detected shape is then passed to a previously trained neural network as a verification step using texture as the feature. As stated by the authors, this system requires more accuracy for use in the real world. A statistical approach for describing the shape of humans using clusters is proposed in [97] and later generalised it for detecting humans and vehicles [98]. In 2000, an interesting work [81] was proposed where human detection is accomplished using the complete human body. It uses a neural network fed with the intensity gradient of the objects. The general architecture of a typical human detection system is outlined in the following section.

2.3.5.2 General software architecture of human detection systems

While doing this literature review, it was found that the following modules are generally proposed for human detection: pre-processing, foreground segmentation, object classification, verification/refinement, and tracking.

2.3.5.2.1 Pre-processing

The pre-processing module includes tasks such as exposure time, gain adjustments, and camera calibration, to mention a few.

(i)Review

Although low-level adjustments, such as exposure or dynamic range, are normally not described in the literature, some recently published papers have targeted image enhancement through these systems. Real-time adjustments are a recurring difficulty in this area, especially in complex scenarios, such as short tunnels, narrow streets, and the rapid motion of the scene. Nayar [99] presents some approaches for performing a locally adaptive dynamic range: fusion of different exposures, spatial filter mosaicing and pixel exposures, multiple image/pixel sensors, etc. Besides, during recent years, solutions exploiting High Dynamic Range (HDR) images [100,101] have gained interest in robotic vision systems due to their potential to provide high contrast in the aforementioned scenarios. In fact, HDR cameras are

useful for night-time vision because of the near-infrared spectrum feature.

Another major topic in the processing module is camera calibration. There are few approaches that tackle both intrinsic and extrinsic on-board self-calibration [102,103]. The most common approach is to initially compute the intrinsic parameters, and then to assume that they are constant while the extrinsic parameters are continuously updated. This procedure is often referred to as camera pose estimation.

The existing approaches can be divided into two categories: monocular-based and stereo-based. In the former case, the algorithms are mainly based on the study of visual features. In [104,105], Broggi corrects the vertical image position by relying on the detection of horizontal edges oscillations. The horizon line is computed according to the previous frames. A comparative study of different monocular camera pose estimation approaches has been presented in [106] which included horizontal edges, feature-based, and frame difference algorithms. In 2006, Hoiem [107] presented a probabilistic framework for 3D geometry estimation based on a monocular system. Other approaches work in Euclidean space. For instance, Sappa [108] proposed fitting 3D data points to a plane. In the Euclidean space, classical least-squares fitting approaches can be followed, while in the v-disparity space, voting schemes are generally preferred (e.g., Hough transform). Recently, Ess [109, 110] proposed the use of pedestrian location hypotheses together with depth cues to estimate the ground plane, which is used to reinforce new detections tracks.

(ii) Conclusion

Stereo-based approaches give more robust results in camera pose estimation compared to monocular approaches. Furthermore, the global error increases with time as long as the estimation depends on previous frames known as the drift problem in such monocular-based approaches. On the other hand, stereo-based approaches (both disparity and 3D data) do not accumulate errors and also feed information about the object's distance from the camera (on board mobile platform). It is not clear whether disparity based approaches are better than 3D data-based approaches. Each approach presents advantages, disadvantages, and limitations.

2.3. 5.2.2 Foreground Segmentation

Foreground Segmentation (i.e. Object of interest or candidate generation), extracts regions of interest (ROI) from the image to be sent to the classification module, avoiding as much background regions as possible. Although some papers do not contain a specific segmentation module (e.g., exhaustive scanning), these techniques are of remarkable importance not only to reduce the number of candidates, but also to avoid scanning regions like the sky. The key to this stage is to avoid missing target objects; or else, the following modules will not be able to fix the error. The term human size constraints (HSCs), which refers to the aspect ratio, size, and position that candidate ROIs must fulfil to be considered to contain a human (e.g., in [111], humans are assumed to be around 1.70 m, with some standard deviation, e.g., 20 cm, tall with a 1:2 aspect ratio; hence, ROIs are constrained to these parameters).

(i) Review

The Exhaustive scanning approach [60,112] is one of the simplest candidate generation procedures; it selects all of the possible candidates in an image according to HSC, without explicit segmentation. This procedure has two main drawbacks: (i) the number of candidates is large, which makes it difficult to execute in real-time, although some proposals have recently studied this problem [64, 65,113], and (ii) many irrelevant regions are passed to the next module, which increases the potential number of false positives. Hence, other approaches are used to perform precise segmentation.

2D-based: Miau [114, 115] uses a biologically inspired attentional algorithm that selects ROIs according to colour, intensity, and gradient orientation of pixels. In several works it is observed that the vertical symmetries in the visible [116,117, 118, 119] and Thermal Infrared (TIR) spectra are used alone [104, 120] or as a complement to stereo imaging [117]. Some implementations include single thresholding [121], double image and hot-spots-based thresholding [122], and adaptive intensity based thresholding [123,124]. Hyper mutation networks [125], which use a multilayer neighbourhood pixel classification, are considered in more sophisticated approaches like [126,127] to classify pixels as foreground/ background.

Background subtraction: Background subtraction is one of the common approaches for foreground segmentation, particularly under those situations with a quite static background. It detects moving regions in an image by counting the difference between the current frame and the reference background image in a pixel-by-pixel fashion. Although it is simple, it is extremely perceptive to changes in dynamic scenes derived from lighting and extraneous events etc. Hence, it is highly dependent on a good background model to reduce the influence of these changes [128,129,130], as part of environment modelling.

Temporal differencing: Temporal differencing makes use of the pixel-wise differences between two or three consecutive frames in an image sequence to extract moving regions. Lipton [131] detects moving targets in real video streams using temporal differencing.

Motion-based: Meyer [132, 133] computes the displacement vector field to initialise a contour based tracking algorithm, called active rays, for the extraction of articulated objects. Inter frame motion and optical flow [134] have been used for foreground segmentation, primarily in the general context of moving obstacle detection. More detailed discussion of optical flow can be found in Barron's work [135]. In [136], Leibe presents a real-time Structure-from-Motion based approach for ground plane estimation.

Stereo: Foreground segmentation using depth data is based on the idea that the human body is located in a 3D environment. Methods are either based directly on estimated 3D data for the scene [137,138,139,140,141] or indirectly by combining different camera views after features have been extracted [142,143,144, 145]. Many authors [116,146,147] have made use of the v-disparity representation [148] to identify ground and vertical objects. Disparity map analysis together with HSC is also used to extract candidates [117,149,150].

The use of multimodal stereo analysis to generate object of interest, i.e., combining visible spectrum (VS) and TIR to perform stereo [151] or a VS stereo pair matched with TIR imaging [147] is a recent proposal. This approach is an example of sensor fusion (detailed in Section 2.3.6). It is worth mentioning here because of its potential to widen the range of working conditions, e.g., VS pair for daytime and a TIR pair for night-time.

Of course, besides the basic methods described above, there are some other approaches for foreground segmentation. In many systems the colour of a human is represented as either a colour histogram [152 , 153 , 154 , 155 , 156 , 157] or a mixture of Gaussian (MoG) [158,159,160,161].

(ii) Conclusion

2D based analysis does not provide satisfactory results at this stage. Parameters such as symmetry are not that much consistent; hence additional cues (i.e. depth) are very useful. Stereo-based approaches have several merits: (i) nice accuracy within the working range of human detection, (ii) robust to environmental variability (e.g., lighting condition in VS and/or temperature in TIR), and (iii) feed helpful information to other modules (e.g., depth value to the tracking module). There are few negative features of those systems as follows: (i) expensive computing cost, (ii) blind areas in non textured sections, and (iii) needs post-processing in order to filter out regions with similar disparity and position to fit the object/human size and aspect ratio [81].

The temporal differencing approach is highly adaptive to dynamic environments, but typically not good enough for extracting all the relevant pixels. Optical-flow-based approaches can be used to detect independently moving objects even in the presence of camera motion. On the other hand, colour histograms are usually compared using the Bhattacharyya distance, which can be improved by weighting pixels close to the centre of the human higher than those close to the border [152,157].

It is clear that in such an oriented application problem, the use of scene prior knowledge plays a key role. Research on more complicated algorithms based on pre-attentive cues and context should be noted: Torralba [162,163] and Hoiem [107, 164]. The major roles of perspective, scene object dependencies, surfaces, and occlusions in object detection are presented in these papers. In addition, active sensors (e.g., laser scanners, PMD), which can estimate distances without high computation times, are also likely to be exploited in specific human detection tasks.

2.3.5.2.3 Object Classification

In the two situation recognition scenarios, the programme receives a list of images that may contain a target object (i.e. a human). The aim of object classification is to classify these images as ones containing a target object (a human) or a non-target object (any other object). This is done in order to minimise the number of false positives and false negatives during the actual recognition process.

(i) Review

Most object classification approaches are entirely 2D and can be classified as either silhouette matching or appearance-based.

Silhouette matching: a binary shape model is the simplest one reported in [117], after symmetry-based segmentation; an upper body shape is compared to an edge modulus image by simple correlation. The Chamfer System is another complex approach where a silhouette-matching technique is used [95,165]. This system consists of a hierarchical template-based classifier (Figure 2.9) that matches distance-transformed ROIs with template shapes in a coarse-to-fine manner. This technique has also been exploited for TIR images in [127]. Also in the TIR spectrum, the research described in [166] uses probabilistic template matching on a multi scale basis.

Appearance-based methods: These processes include the definition of a description of image features, followed by training of a classifier using Region of Interest (ROI) known to contain examples (human) and counterexamples (non-human). Typical learning algorithms used to detect humans are listed in Table 2.1. Following a holistic approach (i.e., target object is detected as one piece) in [55, 95], a classifier is proposed that uses grey-scale pixels as features and an artificial neural network (ANN) with local receptive fields (LRFs) [167] as the learning machine that classifies the candidate ROIs generated by the Chamfer System. In [71 149], Zhao and Thorpe use image gradient magnitude and a feed-forward neural network. The research in [112] reports that the Haar wavelets (HWs; e.g. compute the pixel difference between two rectangular areas in different configurations) are used as features to train support vector machines (SVMs) with pedestrians. Examples of HWs features are shown in Figure 2.10a, 2.10b, and 2.10c; they can be seen as a derivative at a large scale. Viola and Jones [168,

169] propose Adaboost cascades as a learning algorithm to exploit Haar-like features (the original HWs plus two similar features, Figure 2.10d and 2.10e), for surveillance-oriented pedestrian detection. These features have been reasonably successful for object recognition.

Table 2. 1: Different Learning Algorithms Used in Human Detection System.

Learning Algorithm	**Definition**	**Properties**	**Used Features**
SVM [170]	Finds a decision boundary by maximizing the margin between the different classes.	Decision boundary can be linear. Data can be of any type, i.e. scalar or vector features, intensity images etc.	Intensity images [124, 171] Haar wav. [112, 172, 173] HOG [60,121,174,175] Edgelet [174], Shape Cont. [175]
Adaboost [176,177,178, 179,180]	Constructs a *strong* classifier by attaching *weak* classifiers in an iterative greedy manner. Each new classifier is focused on misclassified instances.	Speed optimisation thanks to the use of cascades. Can be combined with any classifier to find weak rules (e.g. With SVM [64]). Few parameters to tune.	Haar [111, 127, 154] HOG [64, 175], EOH [111] Edgelet [69 ,174] Shape Cont. [175]
Neural Networks [181]	Different layers of neurons provide a non-linear decision making.	Many configuration parameters to choose. Raw Data is often used.	Intensity image [182] Gradient image. [81 ,146]

The research in [127] combines Haar-like features with the Chamfer System using TIR imagery. Another recent work [111] makes use of Real Adaboost to select the best features with a set of Haar-like and edge orientation histograms (EOHs; [183]); features which can classify regions of interest in the visible spectrum. Both Haar-like features and EOH can make use of the integral image representation [168], which computes the sum of pixels of a region in just four memory accesses. The research in [60] presents a human classification scheme that uses SIFT-inspired [184] features, known as histograms of oriented gradients (HOGs), and a linear SVM as a learning method. A HOG feature is shown in Figure 2.11.
The method described in [185] has outperformed the state-of-the-art detectors by using a multi-scale histogram of oriented edge energy features quite similar to HOG and the Intersection Kernel SVM. The performance of short segments (up to 12 pixels long) of lines or curves, referred to as edgelets, as features for Adaboost for VS images are reported in [186]. In this case, a mask is attached to each feature, in order to provide pixel-wise segmentation (Figure 2.12). Other features and learning algorithms used in the literature include the gradient magnitude and quadratic SVM [146], Four Directional Features and Gaussian kernel SVM [150], and intensity image with Convolution Neural Networks [168] or with an SVM [124, 171].

HWs and a quadratic SVM to independently classify four human parts (head, legs, right arm, and left arm) are used in [172], then a linear SVM is used to combine the classifications of these parts. In [187], 13 overlapping parts are used (Figure 2.13), as described by SIFT inspired [184] features, and ridge regression is applied to train the classifier of each part. The research in [69] proposes the use of different body parts (i.e. full body, head-shoulder, torso, and legs) and three view categories (i.e. front/rear, left profile, and right profile) to train a nested-weak-classifier Adaboost [165].

The research work [188] sums the classification score of the ROI and six different dynamic parts and the authors use what they call latent SVM and HOG. Dolla´r [189] extends the aforementioned Multiple Instance Learning method to a part-based scheme called Multiple Component Learning, using Haar features. Here, gradient magnitude and orientation features are used. Notably, both approaches [188, 189] avoid the task of manually annotating parts since they are automatically determined by the method. The research [190] recently proposed an approach that combines some of the aforementioned distinct concepts to a greater or lesser extent, i.e., silhouette, appearance, holistic, and parts-based (Figure 2.14).

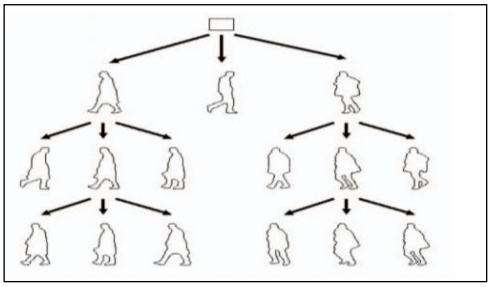


Figure 2. 9 : Hierarchy of templates used in the Chamfer System [165].

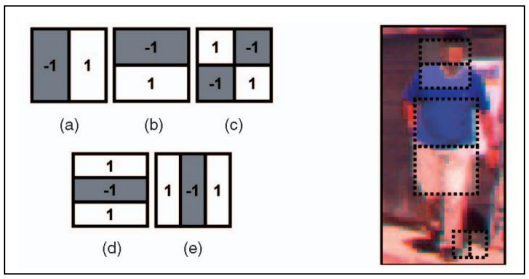


Figure 2. 10: (a), (b), (c), (d), and (e) Haar wavelets and Haar-like features, applied at specific positions of a pedestrian sample in [111, 112, 127, 172].

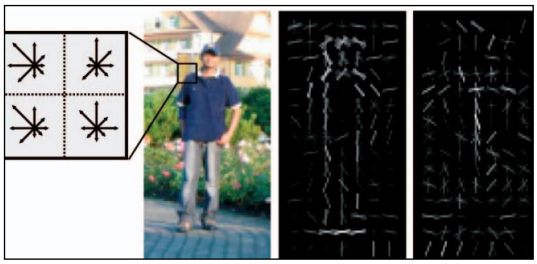


Figure 2. 11: Histograms of oriented gradients in [60]. (a) The descriptor block. (b) Block placed on a sample image. (c) and (d) HOG descriptor weighted by positive and negative SVM weights.

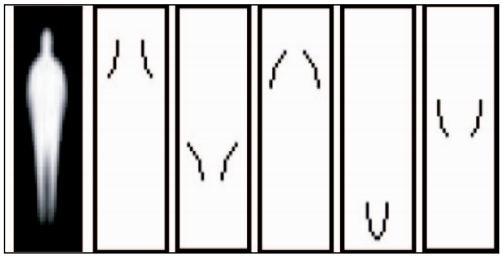


Figure 2. 13: First five edgelet features selected by AdaBoost approached in [186].

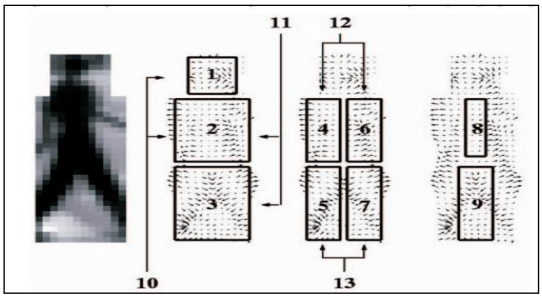


Figure 2. 12: Part-based classification using gradient-based features in [187].

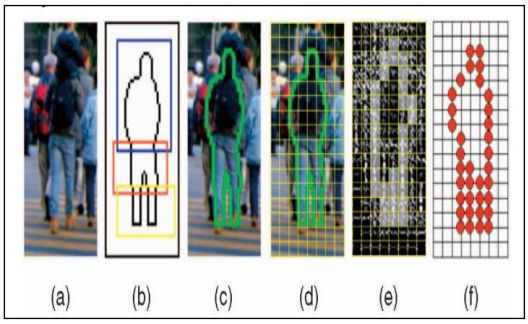


Figure 2. 14: Pose invariant algorithm used in [190]). (a) Input image. (b) Part-template detections. (c) Pose and shape segmentation. (d) Cells grid used for HOG computation. (e) HOGs. (f) Cells relevant to HOG.

(ii) Conclusion

Two conclusions can be highlighted here: HOGs and shape context features are the best option; they are independent of the learning algorithm, and feature combination significantly improves detector performance. In recent years, however, the lack of comparisons has been amended due to reported proposal both in detectors [60] and databases [60], which have been established as a de facto baseline. In fact, many of the techniques proposed within the last few years [185, 188, 189, 189,191] use this benchmark, which makes it feasible to gain insights into the classification module. Given the number of papers presented in recent years, it is not possible to point to one method as the best option.

2.3.5.2.4 Verification/Refinement

Typically an object recognition system contains another module that verifies and refines the ROIs as target object; i.e. human. This step filters out the false positives, using criteria that do not have common characteristics with the classifier, and then performs a fine segmentation of the target object to feed accurate distance measurement or to support the subsequent tracking module.

(i) Review

In the research works [55, 95], detection performance is validated by applying cross correlation between the left side image of a stereo pair and the isolated silhouette computed by the Chamfer system (e.g. shape matching using distance transforms) in the right side image.

For refinement, one important algorithm that gives single detection per target is clustering or non-maximum suppression (e.g. gives estimates of the image gradients). Assuming that

classifiers offer a peak at the correct position and scale of the target and weaker responses within the region, in [192] mean shift [193] approach was used to find the minimum set of ROIs that best fit to the target object (i.e. Pedestrians/human) in the image frame. In [194] proposed are two algorithms. The first generates an activation map, where high-confidence detections point to their neighbourhoods as invalid for new detections. This system is based on a parts-dependent classifier. The second approach constrains the parts which are pre-set to detect only one, and hence non-maximum detections are discarded by iteratively decreasing their confidence.

(ii) Conclusion

This module should be complementary to the classification module. In fact, authors often refer to the reported techniques as two-steps detection, in the sense that verification steps are tied with the output of the classification (i.e., to the characteristics of its false positives). More important is that the stereo information tends to be used only with 2D image-based classification. Additionally, it is quite normal to expect that if larger numbers of cues are used in verification then they provide better results, e.g., Stereo/depth imaging may be used in combination with classification confidence, symmetry, or gait.

2.3.5.2.5 Tracking

The most basic and most important task of a vision system is to use a tracking module to follow a detected target (i.e. a human) over time. This serves as the groundwork for various higher-level applications in many domains (i.e. surveillance, motion capture and so on). This step has various purposes: avoiding false detections, predicting the future positions of current target, thus feeding the foreground segmentation module with pre-candidates, and, at a higher level, making useful inferences about behaviour of the target object (e.g., human walking direction). This module normally involves matching objects in successive frames using features (i.e. points, lines or blobs). The methods of tracking are grouped into four key

categories [195] (Table 2.2): region-based tracking, active-contour-based tracking, feature-based tracking, and model-based tracking. It should be noted that these categories are not absolute in that algorithms from different categories can be combine together [196].

(i) Review

The use of two Kalman filters is proposed in [55, 197], one for lateral motion and the other for longitudinal motion. Both filters determine the speed and acceleration of detected objects. Later, [55, 95] used an α-β tracker (i.e. modified Kalman filter) based on the bounding box drawn from the verification stage. Modified versions were adapted with pre-defined steady-state gains and a constant velocity model. The Euclidean distance between a bounding box centroid, shape dissimilarities (to avoid duplicating the same objects), and the Chamfer distance were used as cues. The research in [119] used ROI overlapping to merge tracks and [147] includes Bayesian probability to give certainty, trajectory, and speed of targets on top of Kalman filter over the time.

Recently, the research [136] proposed the use of a colour model called ‘event cone’ i.e., the space-time volume in which the trajectory of a tracked object is sought. The research [198] proposes the use of network flows to optimise association of detections to tracks.

The research [199] presents tracking of multi-body combined with the implicit shape model detector [200] and the stereo-odometry-based tracker of [110]. In [201] target object detection was executed using a part-based detector, in addition to that a Gaussian process latent variable model was used to compute the temporal consistency of detections over time. Finally, Table 2.2 outlines a general overview of the tracking system.

Table 2. 2: Different Tracking Approaches Used in Human Detection System [195].

Tracking Approaches	**Definition**	**Research Area**
Region-based	Track objects according to variations of the image regions corresponding to the moving objects.	*Vehicle Tracking*- JPL [202], Path system [203] *People Tracking*- colour and gradient feature used with adaptive background subtraction method [128]
Active-contour-based	Track objects by representing their outlines as bounding contours and updating these contours dynamically in successive frames.	*Active-contour-based* [172,204, 205 , 206 , 207, 208] Geodesic active contour [209] Kalman filter for non- rigid (i.e. People) object tracking [210]
Feature-based	Perform recognition and tracking of objects by extracting elements, clustering them into higher level features and then matching the features between images.	*Global feature-based* (include centroid, perimeters, areas, some orders of quadratures and colours) [211 , 212 , 213] *Local feature-based (*include line segments, curve segments, and corner vertices) [208, 214] *Dependence-graph-based* (include a variety of distances and geometric relations) [215] *Combined Approaches* (i.e. shape, texture, colour, and edge features with Kalman filter) [216].
Model-based	Track objects by matching projected object models, produced with prior knowledge, to image data. The models are usually constructed off-line with manual measurement, CAD tools or computer vision techniques.	*Model-Based Human Body Tracking:* Stick figure [217] 2-D contour [218, 219] Volumetric models [220, 221, 222] Hierarchical model [223] *Model-Based Vehicle Tracking:* [224, 225 ,226 ,227 , 228, 229]

(ii) Conclusion

Tracking represents an important aspect of transforming an intruder detection algorithm in surveillance systems. However, this module has not gained as much consideration as other modules; each paper showed its own proposal and no comparisons are outlined. Hence, it is not easy to extract conclusions. Tracking includes different useful mathematical tools (e.g. Kalman filter, Condensation algorithm, dynamic Bayesian network, geodesic method and so on).

In contrast to region-based tracking approaches, active contour-based approaches describe objects in more simple and effective manner and the computational complexity is reduced. In case of disturbance or partial occlusion, these approaches may track objects constantly. But the tracking accuracy is restricted at the contour level.

Feature-based approaches can adapt effectively and rapidly to allow real-time processing and tracking of multi objects. However, dependence graph-based approaches are not recommended to be used in real-time tracking due to its high computational cost. By using information on object motion, local features and dependence graphs, they can handle partial occlusion. However, there are a number of limitations in these approaches: recognition rate of objects based on 2-D image features is low and also unable to improve 3-D pose of objects.

Model-based tracking approaches have the following advantages compared to others: by using the prior information of the 3-D contours, they are intrinsically robust. These approaches can find better results even under occlusion. The structure of human body, the constraint of human motion, and other prior knowledge can be fused in relation to human body model tracking.

2.3.6 Sensor integration and fusion

There are various approaches as to how to combine the output data of different sensors for a robotic application. On the other hand, combining sensor outputs, termed sensor fusion, is an important task. The incoming data from various sensors might support each other, or can be

competitive. Moreover, they might be arriving asynchronously, since some sensors take longer to process than others. Therefore, it is vital to design the sensory system carefully considering the types of sensors in use. Sensor fusion is still an open research area.

From the aforementioned review it is clear that most of the techniques utilise cameras. They are the most commonly used sensors, due to the high potential of visual features, high resolution, and colour cues. However, the performance of image analysis depends on cluttering and illumination, among many other factors. Furthermore, the VS can be affected by glaring sources of light, while TIR can be influenced by other hot objects/ heating source, changing weather conditions, year/season, etc. [230].

The research [231] introduced the combination of a laser sensor with a TIR. It includes a shape extraction process to set ROIs, using Kalman filtering as the data fusion algorithm. The research in [232] presents segmentation and tracking clusters of points along the 1D laser scanner dimension while categorisation is done using the laser sensor data. Human features are extracted with a stereo pair camera in [233]. Data-Fusion of PMD-Based Depth-Information and RGB-Images was carried out in [234] for object recognition. The process highly depends on fast parallel Graphics Processing Units (GPUs) resources.

Combining VS and TIR spectra has also been proposed in [235], where v-disparity is computed using VS, and then, foreground segmentation is carried out in both the VS and TIR. Finally, symmetry and template matching are used to classify, verify, and refine final detections in the TIR. The research [147] evaluates tetra and tri-sensor systems that utilise both the VS and TIR.

Fusion of sensors and active sensors (i.e. VS/TIR/PMD), which are used to obtain complementary information, is being investigated in the context of on-board human detection. The merits and demerits of various sensors can be complemented in order to enhance the overall system performance. Active sensors emit signals and observe their reflection from the surrounding objects, for example, laser scanners emitting infrared light. Typically, these sensors are suitable to detect objects and provide greater range estimates ought to larger distances relative to passive sensors.

Review

It is obvious that future vision systems will highly benefit from the use of multi-cameras [236,237, 238]. The cooperation between multiple cameras relies greatly on fusion of data from each camera. Data fusion is primarily feature-level based rather than image-level based or decision-making-level based. Sensors for surveillance include audio equipment, infrared, ultrasonic, radar and so on along with cameras. Each of these sensors has its own features and characteristics. Surveillance using multiple sensors seems to be an extremely interesting subject. A Photonic Mixing Device (PMD) is based on the principle of time-of-flight and measures full-range depth information in real-time applications. Unfortunately, PMD-based devices have still limited resolution (e.g. 64 x 48 pixels) and offer only grey level information. However the resulting combined RGBZ-data not only enhances the visual result, but also represents a basis for advanced data processing in e. g. object recognition [234].

2.3.7 Summary

To date not a single swarm-based robotics application found, not only in a hospital environment but also for general-purpose use. Schlegel [239] gives a detailed description of the swarm system of hospital robots for the iWARD project. To improve the quality of healthcare, these focal issues emerge: fast identification and location of patients needing immediate attention; reduction of human errors; effective cleaning in hospitals; wider reach of specialist medics, possibly attending patients remotely. To achieve this, iWARD presents a robot swarm delivering support to oversee activities in healthcare environments, providing a multipurpose, cost-effective and scalable solution to enhance quality of healthcare. Four major tasks of the system are: attendance, recognition, communication and support (assisting/cleaning). Attendance means to monitor hospital wards by robots acting as a dynamic swarm. Recognition points out that the swarm is able to recognise patients or objects needing attention, providing immediate information about the location and needs of the concerned patients. The robots can be equipped with different adaptable hardware components for floor cleaning and delivery of food, linen, medicine etc. All mobile robots are capable of providing patients and visitors with guidance and information. It provides easy to

use but high tech interaction interfaces like voice control through mobile and fix-mounted robots. The swarm based approach unburdens the nursing staff from the details of robot control and central coordination - reducing the complexity of robot control to that of a chat, having the swarm negotiating which robot to be uses for each job, shortening the reaction time, reducing human error and increasing efficiency to provide better patient care. This research is part of the iWARD project; its aim is to develop the aforementioned three modules: Environmental condition monitoring, Situation recognition and Patient physical condition monitoring (remotely).

Modular robotics applications have very limited uses in hospital environments. The selected module sensors (mounted on the autonomous robot platform) provide dynamic data for all the proposed modules in this research, as opposed to approaches using stationary sensors. Hence this research is a state-of-the-art of the robotic application in the health care sector.

However, the outcome of the literature review significantly influenced this research in selecting an approach which can solve the target task in a cost effective way.

Chapter 3

MODULE HARDWRE DESIGN

3.1 Hardware Integration with Mobile Robot Platform

The iWARD philosophy suggests that instead of large, specialised robots, general-purpose, small robots with attached service modules are to be used. This has the following advantages:

- Easy configuration
- Easy expansion (with new and existing modules)
- Low cost
- Easy optimisation of task schedules

According to these principles, service modules are attached to a mobile, self-navigating robot platform; these modules are used to execute various scenarios in the hospital. This leads to the following requirements to service modules:

- Interchangeability: each module should have a common (standard) interface to the robot.
- Plug-and-Play: once inserted, a module should automatically register itself with the hardware manager of the robot and should start the execution of the scenario automatically.
- Self-contained: each service module should have its own sensors, computing devices and other hardware and software to execute a scenario.
- Powersave mode: in order to save battery power, all non-essential equipment should be switched off.
- No HRI: none of the modules should have a human-robot interface. A module should not be accessed directly. Instead, the HRI of the robot base should be used.
- Orca component: each module software piece should be developed as an Orca component that communicates to other components using pre-defined interfaces.

One of the major design features of the iWARD robot swarm is that there is no central computer in the system. Instead, the robots are equipped with a level of intelligence to

negotiate and schedule tasks between themselves dynamically – the robot swarm is a self-organising system. The robots themselves decide which of them is most suitable to perform a given task at a given time. The interchangeable robot modules can be attached to the robot bases according to demand. Artificial intelligence learning mechanisms support many robot functions. The objective of this research is to develop the robot modules that are to be attached to the mobile robot bases. Each module has to be a stand-alone, self-contained unit. Researchers at the other partner institutions have developed the robot base and its control system, the human–robot interface, the software platform and the swarm application software.

These functional modules would operate while attached to a robot base in a hospital environment, patrolling the hallways and patient wards for human movement and report any unusual activity defined as the modules tasks to a central terminal and to the SWARM system.

Sensor fusion and Image Processing through a cost effective approach are two primary features of this research. The features of the final product of this research also include environmental condition monitoring, automatic intruder detection and tracking, patient lying on the floor detection and patient physical condition monitoring.

These modules are equipped with different sensors which are detailed in the following section. It individually explains the proposed sensor selection for each module.

In order to accomplish hardware integration and plug-and-play facilities, the following requirements to the functional modules were defined:

3.1.1 Mobile Robot Platform

The developed functional module prototypes are integrated with the Mobile Robot Platforms. In iWARD, two types of mobile robot platforms were developed by the University of Newcastle: one is based on the commercially available Pioneer 3 robot [14], and the other was developed from scratch. Figure 3.1 illustrates the modules and sensors developed as part of this thesis, attached to a robot base built by Newcastle University [240] in one of the iWARD facilities.

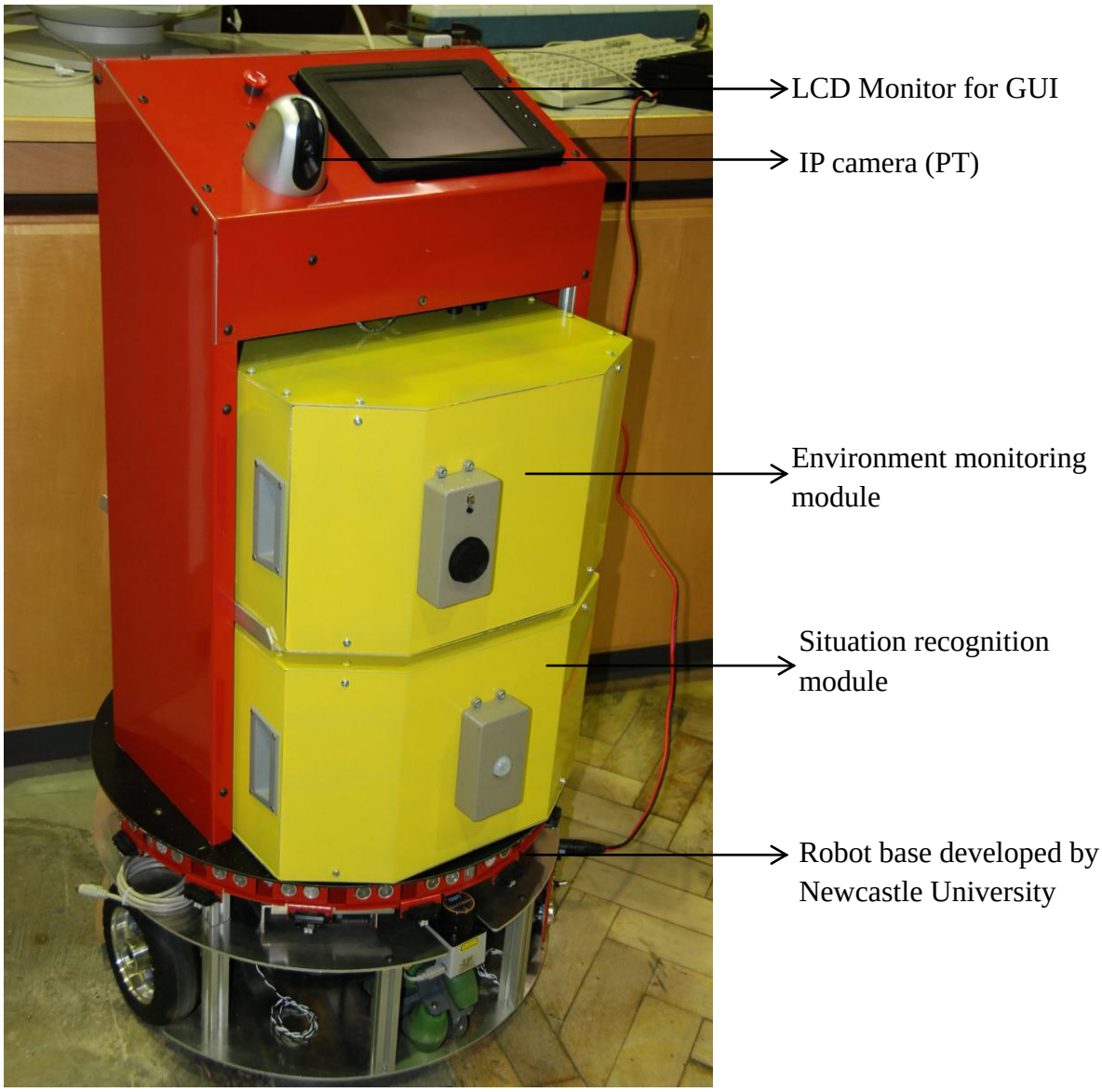


Figure 3. 1: Functional modules integrated with a robot base.

The robot platforms are equipped with motors with 500-tick encoders, ultrasonic (sonar) sensors for object sensing, a 1-D scanning laser sensor for distance measurement, and bumpers for collision avoidance. Communication with other robots and the nurses' terminals is done through local WiFi connection.

3.1.2 Module box

One of the iWARD project partners, Newcastle University [240] designed a steel frame skeleton on top of the mobile robot platforms. The frame also includes a locking mechanism that prevents accidental or unauthorised removal of the module boxes. The design of the module boxes follows the specifications of the robot base. As a compass is not used any more for navigation purposes, the boxes were made of aluminium sheet metal (Figure 3.2) in the DCU workshop facility. Figure 3.4 also illustrates the self locking mechanism of the module drawers to the robot base.

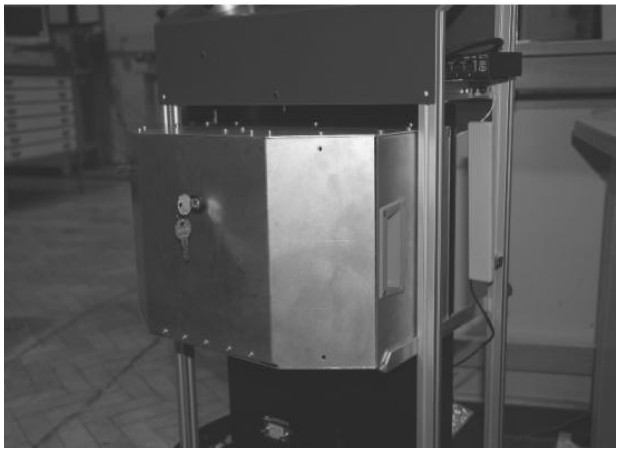


Figure 3. 2: Installation of a module box on an iWARD robot.

3.1.3 Computing hardware

To be self-contained, all modules should have their own computing power. The following are the requirements to module computers:

- Small size: so that it leaves sufficient room for other equipment in the module box.
- Low power: in order to save battery power.
- Selection of ports: to connect various sensors to them.
- Linux operating system: to integrate with the robot on-board computers.
- Low cost.

To satisfy the above-mentioned requirements, Gumstix computer was selected. Gumstix [241] is an embedded, single board, small, open source-hardware computer. Figure 3.3 illustrates a Gumstix tool kit used for the service modules. It has on-board memory, three serial ports, one USB port, and is powered by 5V DC power. Analogue modules sensors can be connected to a DAQ device, and the DAQ device is connected to the Gumstix through a serial port. It comes with a Linux kernel installed in it. All module computing is performed by this computer: plug and play of the module, drivers for sensors and actuators, communication of the module with the robot's computer, and execution of scenarios. All module software is cross-compiled for the Gumstix XScale processor. The Gumstix is powered up automatically once the module box is inserted into the robot.

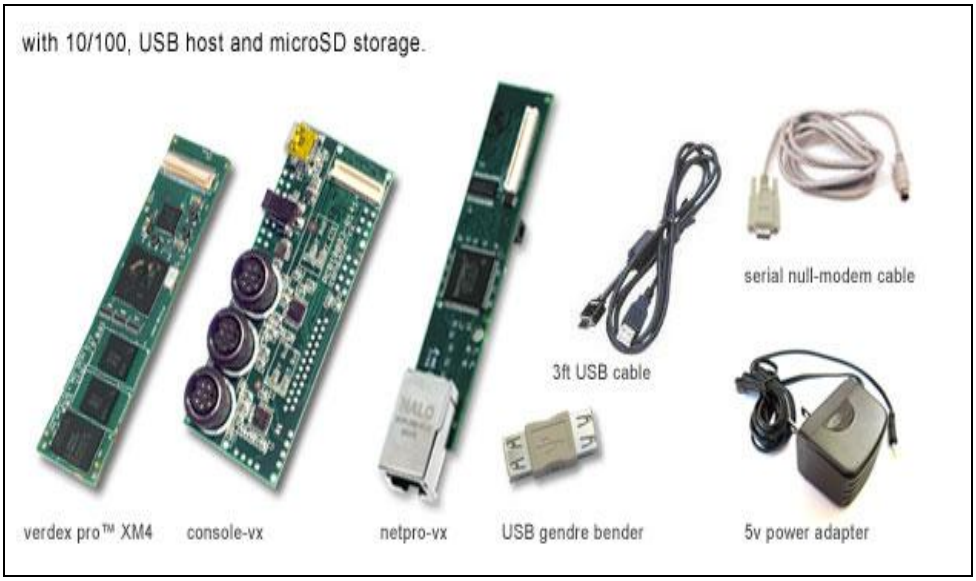


Figure 3. 3: Gumstix tool kit [241].

3.1.4 Power supply, data communication hardware

Since all service modules are standalone and detachable, the power and data connection couldn't be wired all the time. It needs to be connected when the specific module is on service. For easy connection, the modules are powered through a Strix female power jack (power connector) (Figure 3.4). The power connector supplies the power required for the electrical devices (+5V DC and +12V DC) inside the box, drawn from the robot's main battery power supply unit. The Ceep (Ethernet) connector (Figure 3.4) is used to implement

an easy Ethernet connection with the robot's computer. The plug and play term refers to a mechanism which is configured with an electrical circuit and it informs the swarm system about which module is in operation. These features help to make the system modular based.

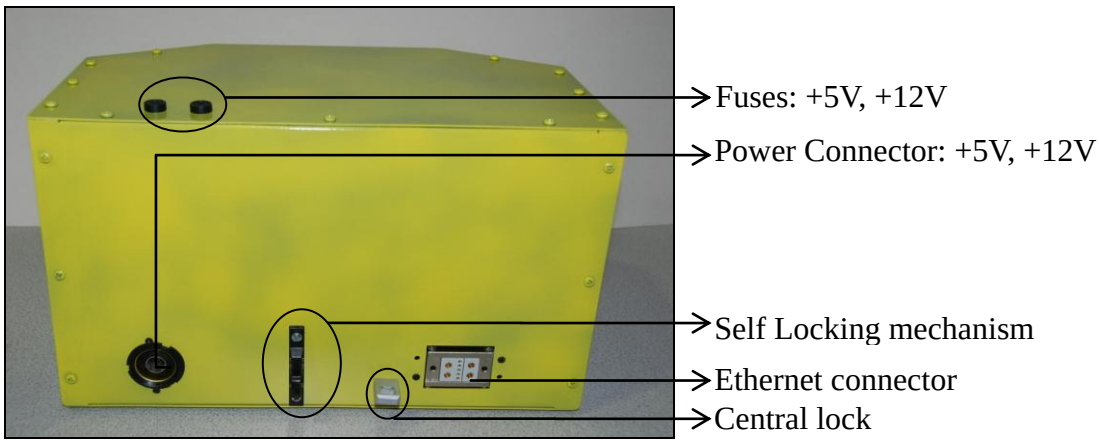


Figure 3. 4: Back side of the Module drawer.

3.1.5 Sensor selection considerations

Sensor selection for the prototypes of this research considered the following factors:

- Type of interface/communication (i.e. USB, Serial/Parallel, Wi-Fi/Bluetooth, Ethernet)
- Software compatibility (i.e. Player, Orca2 and Linux Ubuntu OS supported)
- Requirement of power supply
- Output/Input signal (voltage, current)
- Cost effectiveness
- Capacity or range of each sensor (minimum and maximum, accuracy, resolution).

Hardware selection is scenario-based. Hence the following section briefly describes the sensors of each module individually as follows:

3.2 Environmental condition monitoring module

The environmental condition monitoring module is responsible to collect the data from the attached sensors and compare them with the pre-defined values. In case of any event (i.e. smoke/fire, low/high temperature & humidity), the module informs the robot system to take the necessary action. In this research temperature, humidity and smoke detection sensors are used for the prototype environmental monitoring module. Other sensors can easily be added if necessary.

Figure 3.5 shows the schematic representation of sensors (i.e. temperature, humidity, smoke detector) attached to DAQ hardware, which is then connected to the embedded computer.

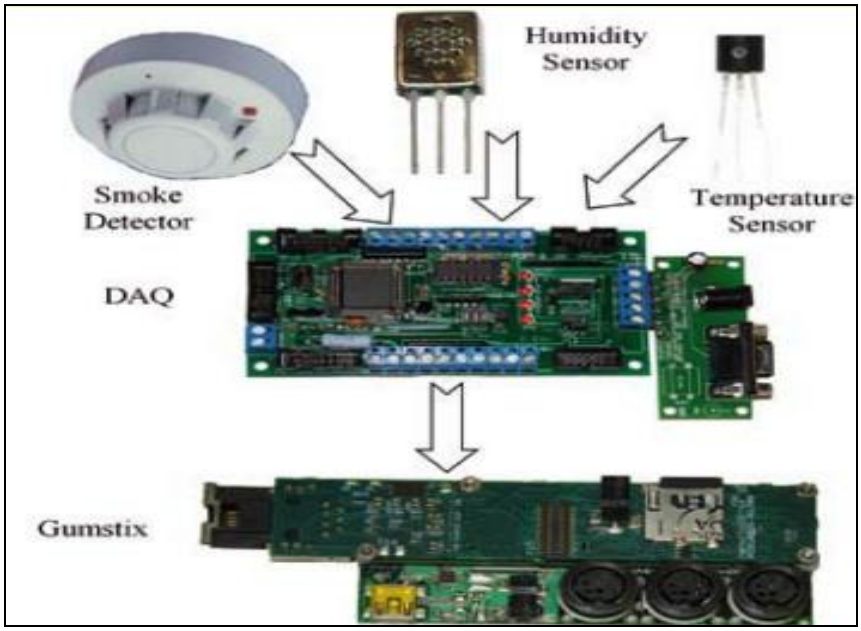


Figure 3. 5: Hospital condition monitoring system & hardware [2].

The scenario of the environmental monitoring system requires the following module specification:

- The system needs to be running continuously as environmental monitoring is not a planned activity.
- The module has to be equipped with a temperature, humidity and smoke detector.

- The robot needs to be able to report its position to relate the measured values to their location.
- A data acquisition (DAQ) device needs to process the analogue output of the above sensors.
- All drivers need to operate the DAQ device in the Linux operating system.
- Threshold values for parameters are stored in the robot's database providing the base for correct interpretation of environmental conditions.

3.2.1 Sensor selection

Sensors detect physical phenomena and convert these to corresponding electrical signals that are comparatively easy to measure, but they do not generate an electrical signal naturally; they are supplemented with electronics to produce the measurable signal. Off-the-shelf measurement sensors are available that offer accuracy within a good range. Before purchasing these sensors a literature survey was conducted to determine the suitable sensors for all of the modules in the current scope.

3.2.1.1 Temperature Sensor

The following are the requirements of the temperature sensor that can be used with the above specified system:

• The temperature sensor needs to operate in a relatively small and quite general temperate range as it will only be used to measure ambient temperature. For instance, a temperature sensor with a range between 0ºC C to 60ºC can easily perform the task.
• The sensor should provide relatively higher accuracy and precision. Expected accuracy is minimum 0.5ºC.
• A temperature sensor that can provide high linear response is preferred. Analogue voltage output is preferred for reducing costs. Using DAQ with a serial port connection is easy for programming in C++.
• For the ease of connectivity, a temperature sensor with a 3 pin (V_{out}, V+ and V_{Ground}) connection is preferred.

• The sensor should have low impedance output and low self heating to avoid systematic errors.

The temperature sensor is presumed to perform in a mild and controlled hospital environment. In other words it does not need to measure any extreme temperature or even any rapid changes. So a quite simple, relatively cheap but accurate temperature sensor should well be able to perform the task for this project.

The most common temperature measurement sensor types are as follows: Thermocouples, RTDs, Thermistors, Infrared, Bimetallic and Solid State. Each type has its own merits and demerits. Hence a careful consideration is essential to select the sensor to be used in this study.

Thermocouples are rugged, and accept a wide range of temperatures, but they have a number of drawbacks: low signal levels, long-term stability, and noise. Response tends to be very time-consuming. The measurement is actually temperature difference, not an absolute temperature level, so an additional measurement is required to set up the reference temperature. Regardless of their limitations, thermocouples are used with success in a variety of applications.

Negative temperature coefficient Thermistors change resistance dramatically in response to temperature changes and usually used for temperature measurements, but they have some limitations: they are vulnerable to physical damage and chemical contamination, need careful protective packaging, limited range compared to other thermal sensors and this often limits how they can be used. Response is reasonably fast, but usually limited due to the packaging. The thermal characteristics are highly nonlinear, hence need corrections.

Resistive thermal devices (RTDs) are generally between thermocouples and thermistors in terms of speed, ruggedness, signal level and temperature range. If not severely stressed, they have good long term stability. They cost more, but offer superior linearity and good accuracy using standard curves without calibration. Though most common metals exhibit RTD effects, platinum alloys have the best range and performance, and are by far the most popular.

Infrared sensors are very useful for measuring extreme temperatures through a viewing port, under conditions that would rapidly destroy other sensor types. The accuracy, stability, and repeatability are not very good, so they need a lot of attention and high cost.

Bimetallic types are too bulky, slow, and vulnerable to mechanical and electrical interference for most applications, but are compatible.

Solid state devices use the thermal properties of semiconductor junction voltages to detect temperatures. They are appealing because sensor, power-gain amplifiers, and "linearisation" can be fabricated on the same chip. The drawbacks are that the operating range of the electronics limits the sensing range, and any calibration other than offset adjustment is impractical. But these are extremely handy for measuring ambient temperatures around circuit boards, and popular for "cold junction compensation" in combination with thermocouples. Accuracy is moderate and they are not associated with high cost.

A solid state type temperature measuring sensor is suitable for the current module based on the overview of the different types of temperature sensors discussed above. There are numerous manufacturers and suppliers in the marketplace who sell solid state temperature sensors. The LM35 [242] (Figure 3.6), is a solid state type temperature sensor selected for the Environmental condition monitoring module. It is a precision integrated-circuit temperature sensor, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. A data sheet for the sensor is available online [242].

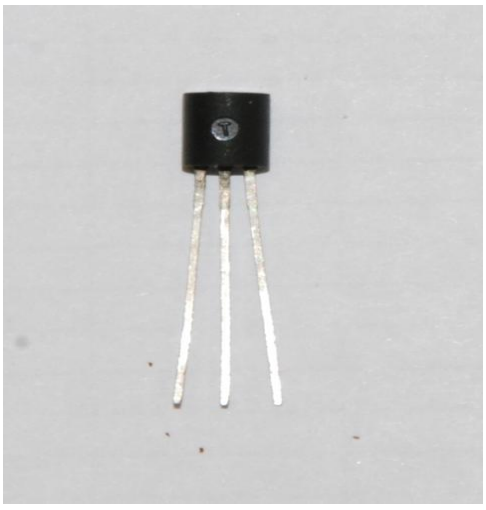


Figure 3. 6: Basic centigrade sensor used in the Environmental condition monitoring module.

3.2.1.2 Humidity sensor

The following are some features and specifications that a humidity sensing device should posses in order to fit in this system.

- Relatively high accuracy; approximately 3%-5% is expected.
- Normal operating temperature range; +10ºC - +40ºC is presumed
- Repeatability
- Interchangeability
- Long term stability
- Quick responding time
- Voltage output is preferable for the sake of data acquisition and programming
- Three-pin interface is preferred for convenient connectivity
- Small size
- Cost effectiveness.

Today, humidity sensors work somewhat differently (i.e. compared to the classic age), in that they compute the difference in electrical conductivity or temperature between moist and dry air. There are three types of sensors commonly used to measure humidity: Resistive Sensors, Thermal Conductivity Sensors and Capacitive Sensors. Each type has its own merits and demerits.

Resistive humidity sensors measure the change in impedance of a specific medium (i.e. salt or conductive polymer) in response to changes in humidity. Resistive sensors comprise noble metals (gold, silver, platinum or palladium) wound in something like a plastic or ceramic tube, or deposited on a substrate using photo resistance. The sensor's respond time is about 10 to 30 seconds, and operating temperatures are up to 100˚ C. Resistive sensors have a major drawback that they consume time to recover when they get wet.

Thermal conductivity humidity sensors measure the difference in radiated heat between a resistor that is exposed to the atmosphere, and another one needs to be encapsulated in a controlled environment. Their respond time is very fast and they tolerate much higher temperatures than other types of sensors (up to 300˚ C).

The relative humidity is proportional to changes in the capacitance of a thin film, measured as the dielectric constant of a Capacitive Sensor. These sensors are also able calculate the dew

point. Measurement time is 30-60 seconds, and operating temperatures are up to 200˚ C. But these are extremely handy and popular for measuring ambient humidity around circuit boards. They are less expensive compared to other types.

Based on the overview of the different types of humidity sensors discussed above, a capacitive type humidity sensor is the most suitable for the environment monitoring module. Humidity sensor 808H5V5 [243] module humidity transmitter is designed based on a capacitive humidity sensor (Figure 3.7) which is selected for the Environmental condition monitoring module. A data sheet is available online [243].

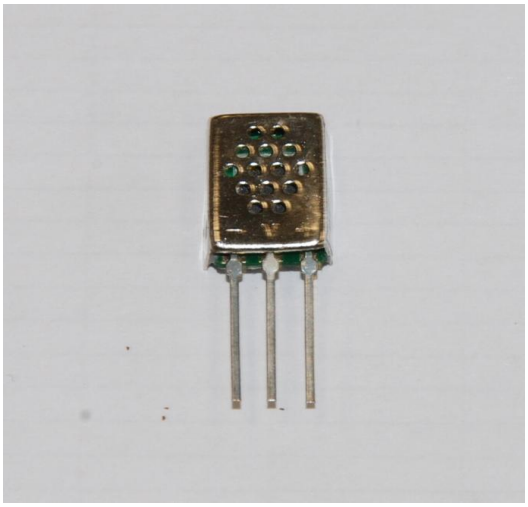


Figure 3. 7: Humidity sensor 808H5V5.

3.2.1.3 Smoke detector

The most common smoke detection types are as follows: Optical, Air-sampling, Carbon monoxide & carbon dioxide detection based and ionisation.

Photoelectric smoke detection is generally more responsive to fires that begin with a long period of smouldering. Although optical alarms are highly effective at detecting smouldering fires and do provide adequate protection from flaming fires, they react slower to rapidly growing fires than ionisation alarms. The type and sensitivity of the photodiode or optical sensor, and type of smoke chamber differ between manufacturers.

To trigger an automatic fire response, such as a gaseous fire suppression system, in mission-critical areas, such as archives or computer server rooms, air-sampling detectors may be used. Most of them are capable of a higher sensitivity than spot type smoke detectors and provide multiple levels of alarm thresholds, such as Alert, Action, Fire 1 and Fire 2.

Some smoke alarms use a carbon dioxide or carbon monoxide sensor in order to detect hazardous products of combustion. The major drawback is that not all smoke detectors of this category are actually able to raise alarm of poisonous levels of those gases in the absence of a fire.

An ionisation type smoke detector is usually cheaper to manufacture than an optical smoke detector, but, in many cases they are rejected due to their being more prone to false alarms than photoelectric smoke detectors [244]. It can detect too small particles of smoke which are normally not visible with naked eye. Ionisation detectors are weaker in high air-flow environments. However, they are superior when detecting flaming fires, which can be categorized by combustion particles from 0.01 - 0.3 microns.

Due to the different levels of detection capabilities between detector types, manufacturers have designed multi-criteria devices which cross-reference the separate signals to both rule out false alarms and improve response times to real fires [245] (i.e. Photo/heat, photo/CO, and even CO/photo/heat/IR).

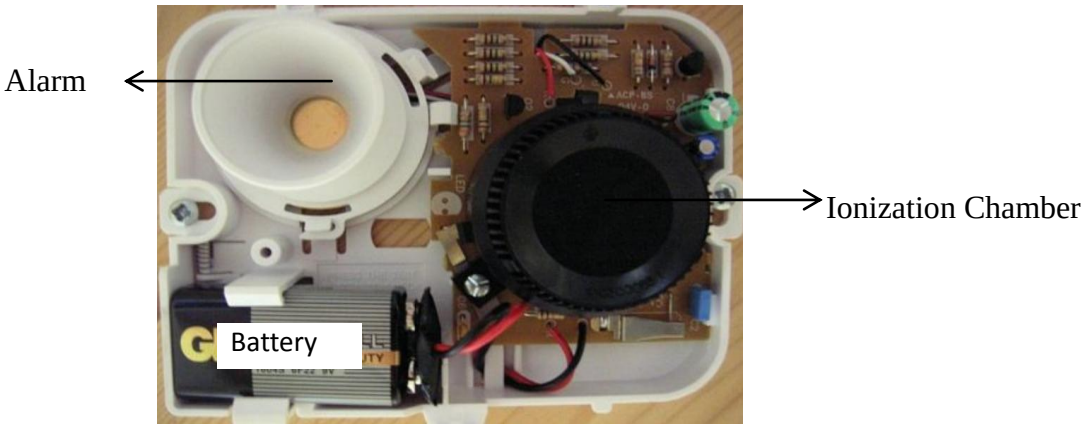


Figure 3. 8: Interior of a basic ionisation smoke detector.

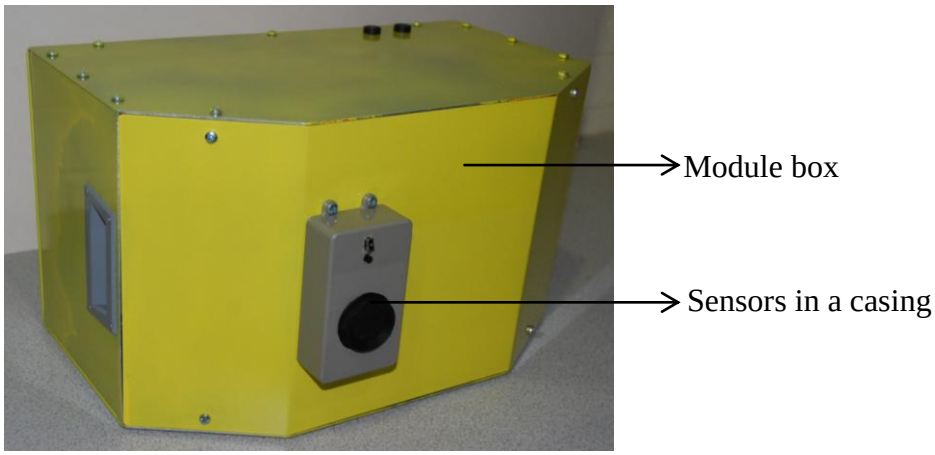


Figure 3. 9: Environmental monitoring module drawer for robot base.

Due to the lower price and superiority when detecting flaming fires, an ionisation type smoke detector was selected for the environment monitoring module (Figure 3.8). It includes a tiny mass of radioactive americium-241, which is a source of alpha radiation and not considered to be a health hazard [246]. The radiation passes through an ionisation chamber, which is an air-filled space between two electrodes, and allows a small, constant current between the electrodes. Any smoke that enters the chamber absorbs the alpha particles, which reduces the ionisation and interrupts this current and activates the alarm.

In this research the physical alarm system is not necessary accept for system testing purposes. Hence the traditional alarm is not included (Figure 3.9) in the module. If the sensor is activated the alarm message is sent to the swarm to take necessary action.

3.2.2 Data acquisition hardware selection

Data acquisition systems, as the name implies, are products and/or processes used to collect information to document or analyse some phenomenon. Data acquisition systems (abbreviated with the acronym **DAS** or **DAQ**) typically convert analogue waveforms into digital values for processing. The components of data acquisition systems include:

- Sensors that convert physical parameters to electrical signals.

- Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values.
- Analogue-to-digital converters, which convert conditioned sensor signals to digital values.

There are different types of Data acquisition systems available in the marketplace: Wireless, USB, Serial Communication Data acquisition Systems and Data acquisition Plug-in Boards.

Wireless data acquisition systems can minimise the cost and time consumed by field wiring of system sensors. These systems consist of one or more wireless transmitters sending data back to a wireless receiver connected to a remote computing device (i.e. CPU). But the data transmission rate is not as fast as with other types and performance could be poor due to obstacles between transmitter and receiver.

The Universal Serial Bus (USB) offers several advantages over usual serial and parallel connections, including higher bandwidth (up to 12 Mbits/s) and the ability to provide power to the peripheral device. Since USB connections supply power, only one cable is required to link the data acquisition device to the computer, which most likely has at least one USB port. Hence initial experiments were performed with a LabJack (U3) [247] USB DAQ device. The only drawback observed was that the cross compilation of the driver was not compatible with Gumstix due to its default kernel settings.

Computer data acquisition boards plug directly into the computer bus. Major advantages of using boards are speed and cost (i.e. the overhead of packaging and power is provided by the computer). Their major drawback is that they are primarily for IBM PC and compatible computers. Also, the Gumstix does not have a bus structure.

Serial communication data acquisition systems are a good choice when the measurement needs to be made at a location which is distant from the computer and cost is low. Different communication standards are available: RS232 is the most common but only supports transmission distances up to 20 meters. RS485 supports transmission distances up to 200 meters.

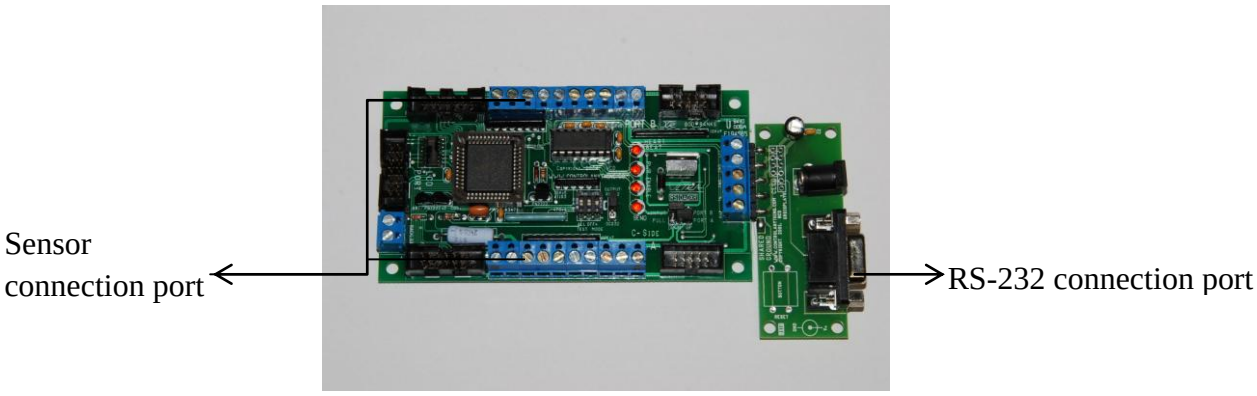


Figure 3. 10: RSIOADXR RS-232 DAQ card.

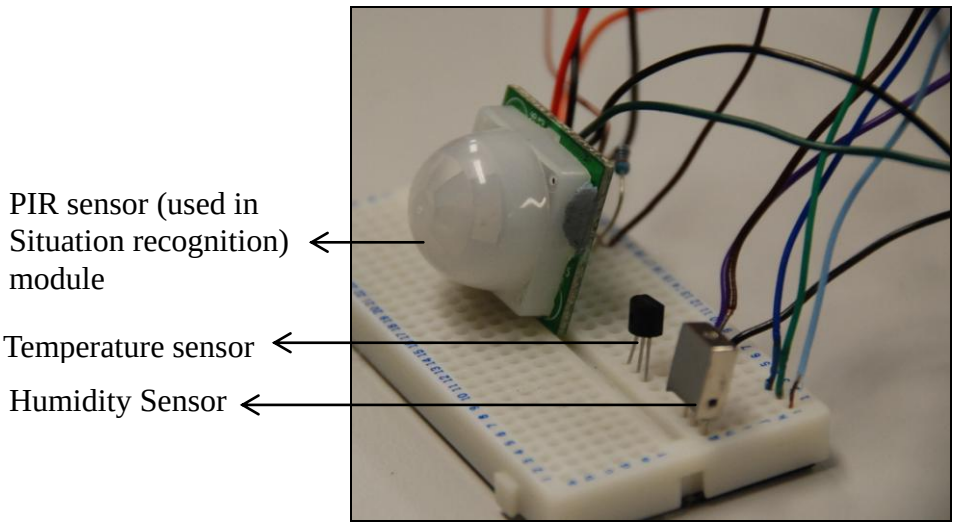


Figure 3. 11: Sensor setup with a breadboard during the experimental stage.

Since transmission distance is not an issue in a module box, and the Gumstix has three RS232 serial ports, as an alternate choice to USB, an RSIOADXR RS-*232* (Figure 3.10) with *A/D* 8-Channel *Dual XR Controller*, 10-Bit, A/D Dual XR Controller with RS-232 Interface [248] is used as a DAQ hardware is selected for the environment monitoring module. Its main features include: 16 On-Board Relays and 16 User Programmable TTL/CMOS Data Lines (Shared with 8- Channel, 8-Bit/10-Bit Analogue to Digital Converters). All data ports are detachable if they are not needed. Figure 3.11 illustrates sensor setup with a breadboard during the experimental stage. The smoke detector is not visible in the figure. This setting was used to test the module sensors and the DAQ device during the pre-development stage of the module.

The RSIOADXR DAQ kit did not include a Linux-compatible device driver, so it had to be developed based on the signal descriptions.

3.3 Situation recognition module

As was explained earlier, this module caters for two scenarios:

(i) Surveillance (Intruder monitoring)

Intruder monitoring is responsible for identifying an illegal person in the hospital environment, after detection tracking the person, and storing the person's image. Finally, it should report the incident to the robot system. This scenario requires two tasks to be solved: motion detection with distance measurement, and face detection with image analysis. The following sensors and hardware are needed to accomplish these tasks.

Figure 3.12 and 3.13 show the sensors (PIR, Distance measurement sensor, Camera) which could be used to detect any intruder in the hospital environment. A vision system featured with face detection and contour analysis image processing technique can be used to detect the unique badge that every legal person in the hospital should wear. If a face is detected but the badge can not be found relative to the face, the system considers the detected face as a potential intruder, which is reported in details in the next chapter.

Figure 3. 12: Sensors for motion detection and distance measurement [249].

Figure 3. 13: Sensors for face detection and image analysis [249].

(ii) Detection of a Patient lying on the floor

This scenario deals with the detection of a patient who has fallen and lying on the floor. Recognition of a human body lying on the floor is a unique task for a vision system. The principal sensor necessary for this task are shown in Figure 3.14.

Figure 3. 14: General graphical overview of the sensors of the vision system to recognise a patient lying on the floor [2].

Many techniques and several sensors for this scenario can be shared with the intruder detection scenario; indeed, this is the reason for including both as one stand-alone service module. The requirements of the situation recognition module are as follows:

- The system operates mainly at night time.
- The PIR sensor needs to remain in operation when the system starts.
- The system needs to be able to get location information from the robot and communication is needed with the robot's speed control software to stop/start when it is needed.
- The module has to be equipped with a vision system. Since the position and orientation of the moving robot relative to the patrolled environment changes all the time, the camera has to be with a pan and tilt mechanism. This allows pointing the camera towards the detected person without having to move/rotate the robot. For getting the best image from a distance a camera with at least 5 times optical zoom (15-75 mm focal length equivalent on a 35 mm film camera) is preferable. Without optical zoom only basic image recognition (face recognition) is feasible. As the vision system needs to produce quality image in various lighting conditions, a camera with programmable exposure settings is preferable. If only auto-exposure mode is available, in poor lighting conditions the camera may attempt to boost its sensitivity by overcompensating (large apertures) which may lead to colour distortion and the colour separation algorithms not being able to function. The vision system needs to be able to produce streamed video image as well as still images captured from the stream.
- The camera needs to have a wide viewing angel for viewing a large portion of the environment.
- A 3D distance measurement system is needed for getting the distance information in a predefined angle in the path of the robot. This output is to be compared with the camera image in order to separate the image of the human from background images.
- The robot needs to be able to report its position where it found the intruder or any patient lying on the floor after detection.

3.3.1 Camera selection

One of the major sensor components for this research, where digital image processing is highly involved is a camera. A vision system can't run without a camera. There are a wide range of cameras (used in image analysis) available on the market from different manufacturers. Based on the requirements in the previous section and as a result of an extensive review of available video cameras, several cameras were used in iWARD and in the research that followed it:

- ***Abus Digi-Lan TV7240 IP camera***

The TV7240 (Figure 3.15) [250] is a low-cost pan-and-tilt IP camera. In the iWARD project video cameras were used not only for the situation recognition module but also for navigation, for the remote consultation scenario, and as part of the human-robot interface. This meant that each robot was permanently equipped with a video camera. In order to share equipment, service modules that required a video camera did not have one installed as part of the module, but instead used the one that was permanently installed. For budgetary reasons this low-cost video camera was selected. Although it does not have mechanical zoom, it was a reasonable selection to prove the concept of iWARD.

Figure 3. 15: TV7240 IP camera [250].

Since the TV7240 is an IP camera, it has a built-in web server and thus all communication with the camera is done by directly accessing it through the internet. This includes streaming

images from the camera and controlling it. This type of camera is also more flexible to be used in an Orca component.

- ***Philips ToUcam Pro II***

The Philips camera (Figure 3.16) was used for experimentation purposes during the iWARD project and after that. It is a compact USB-based analogue camera. Its exposure rate can be set manually, which was required for the situation recognition module. Since it is not an IP camera, a device driver was developed for accessing it (streaming and controlling). This camera does not have a pan/tilt/zoom mechanism, but was good enough for experimenting with various imaging techniques.

Figure 3. 16: Phillips web camera.

- ***Canon VC-C4***

After the completion of the iWARD project, the robot service modules undergone further development. In order to improve the image recognition a tracking abilities of the situation recognition module, a high-quality pan-tilt-zoom (PTZ) camera was needed. This allows image recognition from a larger distance and object tracking without the need for rotating/moving the robot. The VC-C4 (Figure 3.17) [251] was selected as a good-quality PTZ camera. It has a 16x zoom and streams video in NTSC format. Since the VC-C4 only provides analogue video output, a frame grabber is used with it. This transforms the analogue image to digital and transmits it through a USB port. A device driver in C++ was developed to stream video, and to control the camera through a serial port. The VC-C4 was used for experimentation purposes only.

Figure 3. 17: Canon VC-4 PTZ camera [251].

Figure 3. 18: Axis 214 PTZ IP camera.

- ***Axis 214 PTZ IP camera***

In order to eliminate image translation, and for easier access and control of the camera, for the final design of the situation recognition module the Axis 214 PTZ IP camera (Figure 3.18) [252] was selected. Its 18x motorised zoom, 0.3 lux minimum illumination (colour mode), 704x576 pixel resolution, built-in web server, and full control makes is an ideal choice for high quality imaging and object tracking.

Camera calibration

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. In practice, no lens is perfect. This is mainly for reasons of manufacturing ability. It is also difficult to mechanically align the lens and imager exactly. There are two main lens distortions: *Radial distortions* arise as a result of the shape of the lens, whereas *Tangential distortions* arise from the assembly process of the camera as a whole. The process of camera calibration includes two types of models: the camera's geometry model [253] and a distortion model [253] of the lens. These two informational models define the intrinsic parameters and distortion vectors of the camera. Camera calibration means a transformation process which characterises a particular camera using mathematical calculations which find the quantities internal to the camera that affect the imaging process. These parameters include:

- Position of image centre in the image
- Focal length
- Different scaling factors for row pixels and column pixels
- Skew factor
- Lens distortion (pin-cushion effect)

OpenCV [254] provides several algorithms to help to compute the camera intrinsic parameters. The actual calibration is done via the cvCalibrateCamera2() function. In this routine, the camera is targeted on a known structure that has many individual and identifiable points. By viewing this structure from different angles, it is possible to compute the location and orientation of the camera relative to each image as well as the intrinsic parameters of the camera (see Figure 3.19-a). In order to provide multiple views, during these experiments the used object was rotated and translated (Figure 3.19-b).

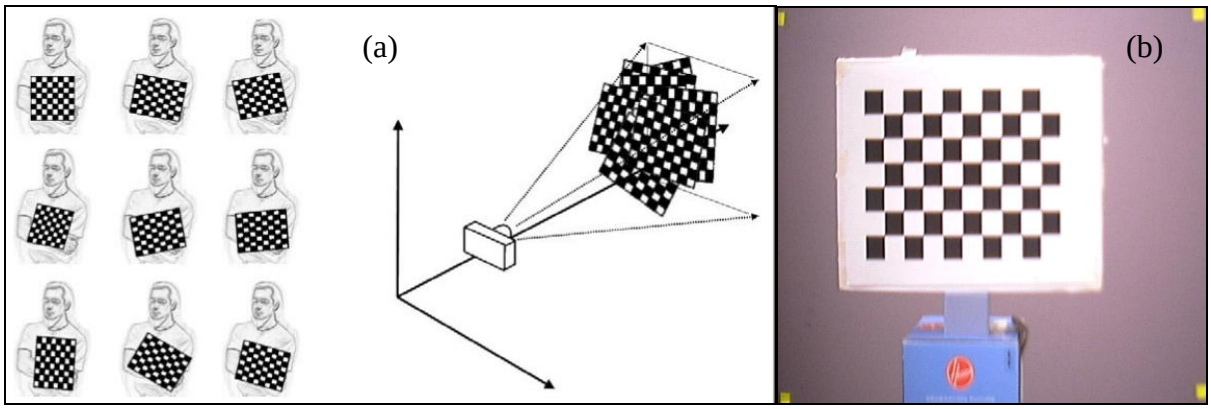


Figure 3. 19: (a) Images of a chessboard being held at various orientations [253] (b) chessboard used for camera calibration in this research.

The calibration step provides the internal parameters for each camera: the focal length f_c, the principal point c_c, the skew α_c, and the distortion coefficients k_c. These allow correction of the distortion in the input images (using the 'undistort' algorithm in *OpenCV*). Particularly, the results contain the camera intrinsic matrix, the distortion coefficients, rotation and translation vectors that define the object's (i.e., the chessboard's) position and orientations. The distortion coefficients (k1, k2, p1, p2, and k3) are the coefficients from the radial and tangential distortion equations which help to correct such distortion. The camera intrinsic matrix allows transformation of real-time 3D coordinates to the image's 2D coordinates. The camera matrix

can also be used in a reverse operation, but in this case we can only compute a line in the three-dimensional world to which a given image point must correspond. The algorithm OpenCV [254] uses to solve for the focal lengths and offsets is based on the works in [255] and [256] to solve for the distortion parameters [253]. Table 3.1 shows typical calibration results for the Phillips USB camera generated in YML format during the experimental stage.

Table 3. 1: shows typical calibration results for the Phillips USB camera generated in YML format during the experimental stage.

%YAML:1.0. calibration_time: "Thu 20 May 2010 14:46:20 IST"
image_count: 10
image_width: 640, image_height: 480
board_width: 6, board_height: 8
square_size: 2, flags: 0

camera_matrix: !!opencv-matrix
rows: 3, cols: 3, dt: d
data: [1.0620485829121696e+03, 0., 294.3815190540113917, 0.,
1.0549024461215070e+03, 473.3151834447116926, 0., 0., 1.]

distortion_coefficients: !!opencv-matrix
rows: 1, cols: 4, dt: d
data: [0.2945735145756788, -0.6564721062111361, 0.0468938656975349,
-6.4126889163699250e-03]

avg_reprojection_error: 0.1902364412943522
per_view_reprojection_errors: !!opencv-matrix
rows: 1, cols: 10, dt: d
data: [0.1888151168823242, 0.2043771743774414, 0.1965344746907552,
0.1874901453653971, 0.1806224187215169, 0.1655991872151693,
0.1921892166137695, 0.1915963490804037, 0.1765368779500326,
0.2186034520467122]

A set of 6-tuples (rotation vector + translation vector) for each view
extrinsic_parameters: !!opencv-matrix
rows: 10, cols: 6, dt: f
data: [-1.85121393, -1.88564038, -0.53807497, -1.78957784,
-18.08477592, 59.56589890, -1.84628880, -1.88268471, -0.53758430,
-1.67119050, -18.09909439, 59.70355606, -1.83639503, -1.86979532,
-0.55218804, -1.71885145, -18.36124039, 58.94250488, -1.83887339,
-1.87294090, -0.54902714, -1.70800173, -18.31420708, 59.10222626,
-1.84558964, -1.88247275, -0.53473294, -1.66491044, -18.17015266,
59.52830505, -1.84819639, -1.88593984, -0.52709448, -1.60870707,
-18.18239212, 59.81133270, -1.84334123, -1.88164198, -0.53416443,
-1.59329534, -18.12213135, 59.66894531, -1.84591746, -1.88563967,
-0.52805537, -1.56062162, -18.10741425, 59.85022736, -1.84568250,
-1.88488042, -0.52943319, -1.55628753, -18.12622452, 59.84625244,
-1.82364035, -1.85949290, -0.56025887, -1.58443582, -18.42090988,
58.95009995]

The internal parameters of the other cameras were calculated in a similar manner.

The camera viewing angle is another important parameter for image processing. For the situation recognition module, a wide angle for viewing a large portion of the environment is necessary. The viewing angle also defines the necessary orientation angle of the camera relative to the horizontal direction. In this research, the viewing angle (both horizontal and vertical) was calculated using a triangulation technique:

- An image was taken where the four corners were marked (Figure 3.19) with reflective patches to get the image width (W) and height (H).
- The distance (D) between the camera lens and the centre of the image was measured.
- Then equation (1) was used to calculate the half viewing angle theta (θ) (Figure 3.20) either for the vertical or horizontal direction.

$$Tan\theta = \frac{a}{D} \ldots (1)$$

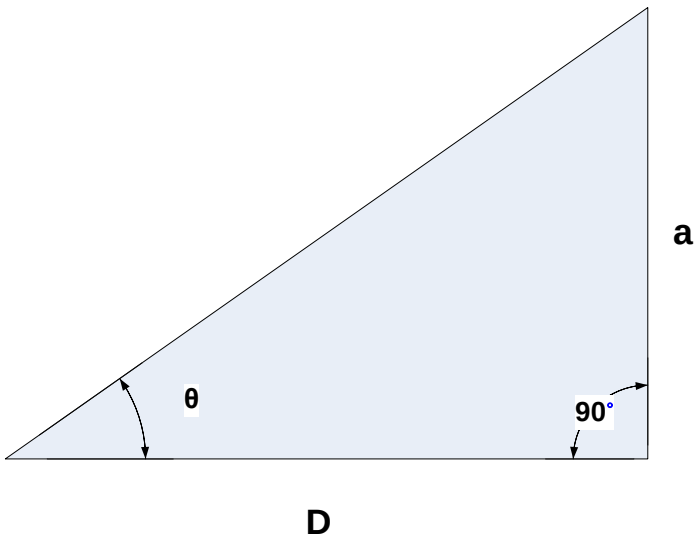


Figure 3. 20: Camera viewing angle calculations.

In the figure, ‘a’ is the half of the image width or height (vertical/horizontal direction), ‘D’ is the distance of the image from the camera, and theta (θ) is the half viewing angle.

As an example, the obtained field of view (FOV) of the Phillips camera is 36.2º in the horizontal and 27.8º in the vertical direction. The PMD laser sensor's measurements are: 41.2º horizontal and 31.7º vertical FOV.

These calibration results are used during the vision system software development process which is detailed in the corresponding chapter.

3.3.2 Distance measurement hardware selection

In addition to identifying the presence of a person in an image, in the context of the situation recognition module it is also necessary to measure their distance from the robot. This depth measurement provides additional information for the image recognition system. Combining colour with depth is a **sensor fusion** technique that makes is easier to separate the image of a foreground object (possibly a person) from the background.

Distance measurement is a broadly used approach for location estimation. There are three major approaches for measuring the distance to an object from a given reference point: Time-of-flight, Time-Difference-of-Arrival (TDOA) and Image processing.

The Time-of-flight technique measures the time 't' taken for some signal (i.e. laser beam or infrared) to traverse the path between two points (the reference point and the object). If the speed of the signal is 'v', the distance 'd' is given by $d = v \times t$. For example, Global positioning systems (GPS) use the time of flight of radio frequency (RF) signals to approximate the distance between GPS satellites and the GPS receiver [257].

TDOA-based approaches calculate the distance between two given points using two signals with different speeds that pass through the same path between the two points. Two-way ranging does not involve synchronization among nodes however; this method may give inaccurate range information for asymmetric channels [258]. Consider two signals A and B with speeds vA and vB sent simultaneously by a transmitter. If vA > vB, then signal B lags behind signal A as they propagate. Lets 't' represent this time lag at a receiver positioned at a distance 'd' from the transmitter. Then,

$$d = \frac{t}{\left(\frac{1}{v_A} - \frac{1}{v_B}\right)} \qquad \text{... (2)}$$

For example, the Cricket device uses TDOA of RF and ultrasonic signals to measure the distance to the reference points.

Image processing (stereo vision) techniques are commonly used in passive auto-focus cameras. Here a microprocessor moves a motor-driven lens located in front of a Charge Coupled Device (CCD) sensor array until a focused image of a target is formed on the sensor. Once the image is in focus, for a given lens, the distance between the CCD sensor and the lens 'l' defines the distance 'd' to the target; the distance 'd' can be computed from 'l' and the 'F' number of the lens using the laws of geometric optics [259].

Each of the above mentioned techniques has some drawbacks, such as ToF is not suitable under direct sunlight, TDOA computes distances with errors in the presence of obstacles between the transmitter and receiver, and Image analysis techniques involve high computation cost. However, the ToF (laser sensor) technique can measure the distance between the robot and the person, thus enabling the camera to point at the person more accurately without changing the robot's position. A low-cost and fast alternative to standard techniques like laser scanners or stereo vision is the distance measurement with modulated, coherent infrared light based on the Photo Mixing Device (PMD) technique [260]. Hence for the situation recognition module a laser sensor was selected for distance measurement.

PMD 3D laser sensor

The O3D100 PMD 3D laser sensor from Ifm Electronics (Figure 3.21) [261] is the first industrial 3D sensor that can detect objects in three dimensions at a glance. It is based on the technology of time-of-flight measurement. The measurement and the evaluation of the time-of-flight are integrated on one chip. The resolution of the sensor is 64 x 48 pixels. Each pixel of this matrix evaluates its distance to the object. The image of the object on the matrix and the respective distance values correspond to a 3D image. In effect, the sensor provides a distance matrix. When combined with the readings of a camera using image fusion techniques, the laser sensor provides depth information for each coloured pixel of the camera image. Sen-

sor fusion enables easier separation of the image of the human body as the laser sensor's imaging is only affected by distance and not by colour.

Figure 3. 21: O3D100 PMD 3D laser sensor.

Besides the distance matrix, the laser sensor can also light intensity. In this mode, the sensor measures light intensity of the reflections from the target object. This can be used for calibration purposes by applying highly reflective patches on the target object. Those patches then show up as bright spots in the intensity matrix and can be used to identify object boundaries in the distance matrix. The O3D provides two interfaces; one is an Ethernet connection for parameterising the sensor and obtaining data. The other one is a process connection providing digital input/output lines and an analogue output. The communication with the O3D is based on two TCP/IP ports. The first port configures the camera settings based on an XMLRPC server-protocol. The second port transfers image-data based on a TCP/IP byte stream socket.

3.3.3 Sensor selection

PIR Sensor

In order to save computing power, some of the image recognition and analysis tasks can be triggered by an occupancy sensor in the presence of a human being within the range of the

robot. Passive infrared (PIR) sensors are the most commonly used occupancy sensors. There are different types of PIR sensors available on the market; they vary in detection range, mounting option, power supply and so on. After conducting a review of different PIR sensors, the KC7786 (Figure 3.22) [262] was selected. It is a PIR sensor module which is very small in size and has a built in amplifier and logic circuit. The Fresnel lens is included for the detection of human body movement at a distance of 5 meters. This is suitable for automatic illumination systems and intruder alarm systems.

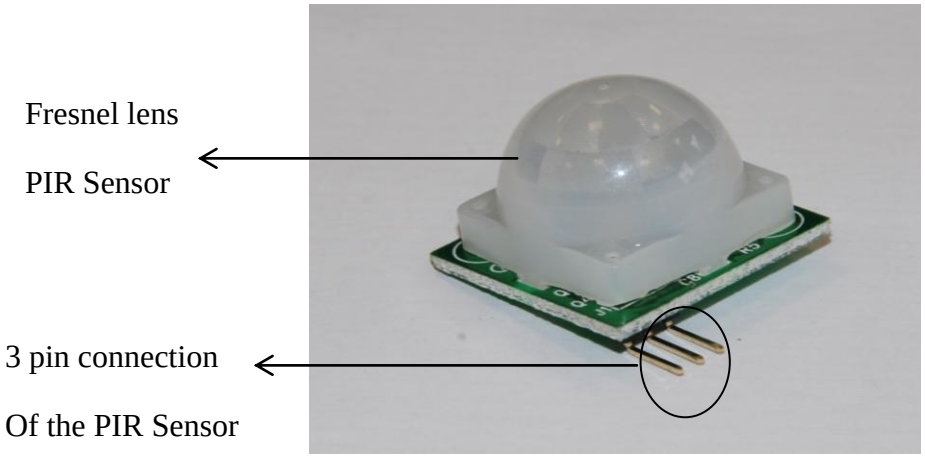


Figure 3. 22: KC7786 PIR sensor module.

Figure 3.23 shows the situation recognition module with one PIR sensor. Note that in the iWARD system there was no laser sensor, and the video camera was mounted on the robot base itself so that it could be used for other purposes as well, like navigation. However, in the post-iWARD final version of the module a laser sensor was added, and together with the higher quality Axis camera they were mounted on the module box.

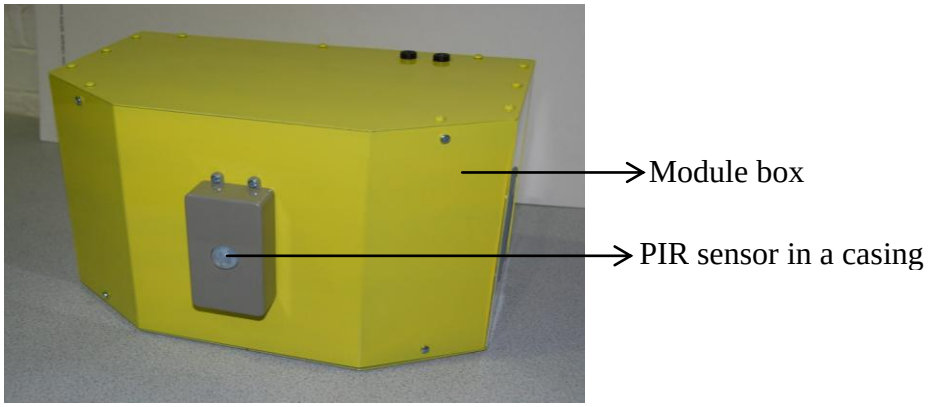


Figure 3. 23: Situation recognition module (the computing device is inside the drawer).

To get the output of the PIR sensor as a digital signal, the same RSIOADXR data acquisition device that was shown in Figure 3.10 is used.

3.3.4 Experimental equipment

In order to experiment with sensor fusion, that is to add depth information to a coloured image, an experimental setup rig was designed (Figure 3.24). To maximise the useful depth data, the cameras should be placed as close as possible to each other. In this case, the video camera is mounted beside the laser sensor.

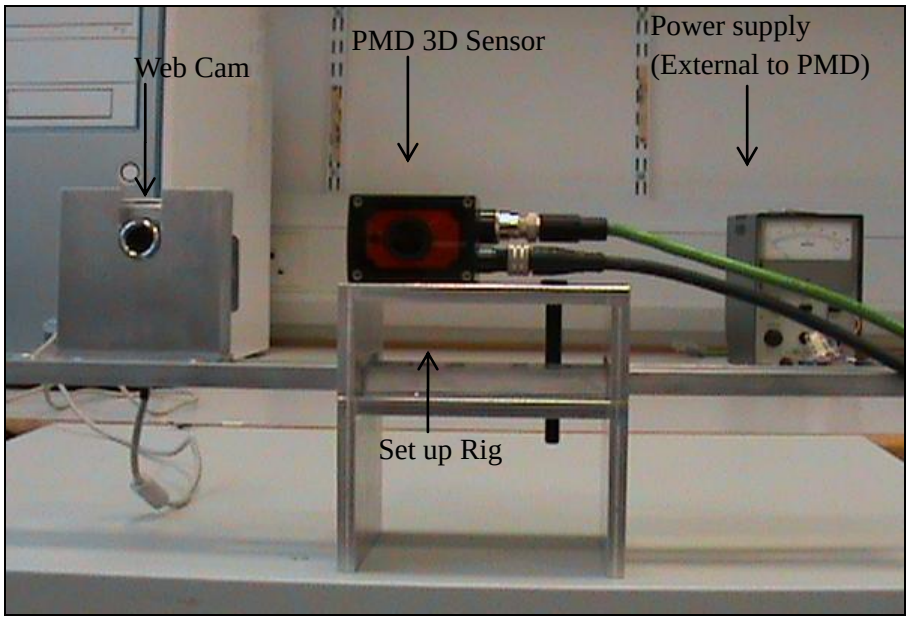


Figure 3. 24: Laser sensor and video camera assembled on an experimental rig.

3.4 Patient physical condition monitoring

Patient physical condition monitoring is a critical component of medical care. Monitoring of vital signs is also of interest of businesses with dangerous working conditions. Monitoring the physical condition of patients is the task of this module. According to the module specifications, vital signs of a patient that need to be measured include body temperature, heart rate, ECG, respiration rate, body acceleration, and others). These data need to be processed and

analysed to make a decision regarding the patient's health condition (reported more detail in the next chapter). Figure 3.25 shows the principle of measuring vital signs of a patient. The monitoring system compares the obtained values with the pre defined normal range values to monitor the patient's physical condition.

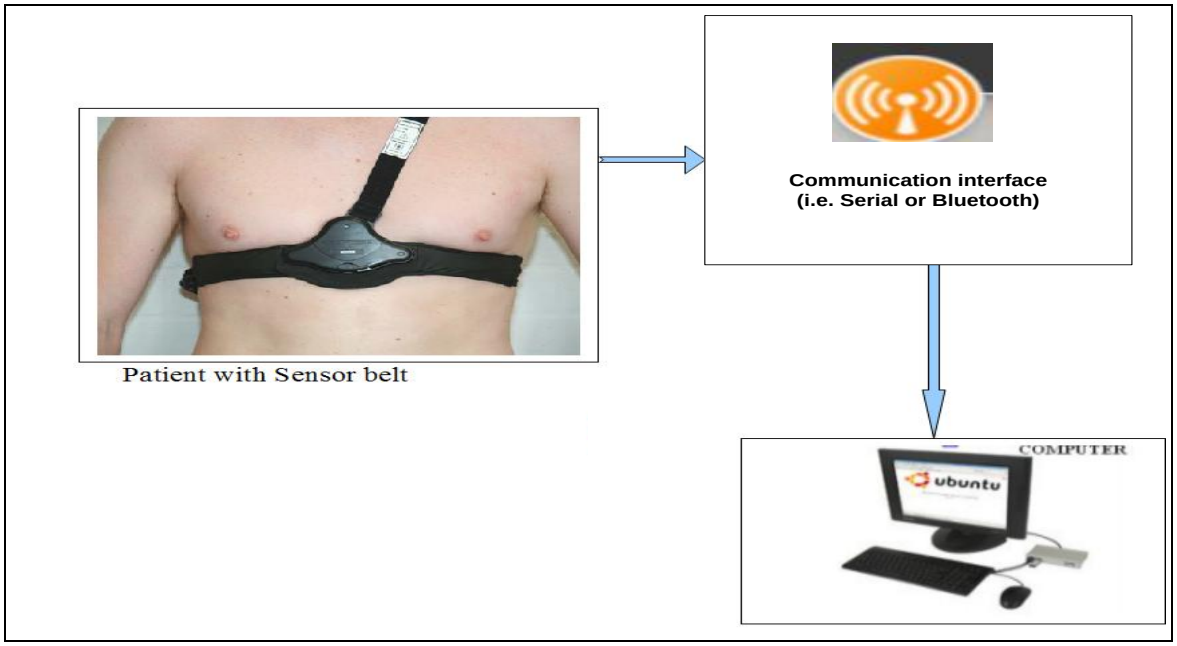


Figure 3. 25: Patient condition monitoring system & hardware.

3.4.1 Sensor selection

There are several integrated sensor solutions on the market for monitoring human condition:

- ***Equivital™ system [51]:***

The small, light, body-mounted, wireless multi-sensor module collects data (e.g. heart rate, respiration effort rate, skin temperature, body position and motion) and transfers it via a wireless network to a computer terminal. A special software tool enables displaying and storing the data received.

- ***NeXus-4 system [52]:***

The portable, pocket-size system provides several input channels enabling to measure an individually fixed set of specific vital signs chosen from EEG, EMG, ECG, EOG, BVP, skin

conductance or temperature, respiration and heart rate. Physiological data can be stored either on the built-in flash memory or has to be forwarded to a computer.

- ***Vivoresponder™ system [263]:***

The real-time, remote life-sign monitoring system developed for people working in dangerous environment can be applied in healthcare institutions as well. Sensors measuring respiration, heart rate, skin temperature, activity and body position are included in the lightweight chest strap.

- **Sensium™ system [264]:**

The ultra low power sensor interface and transceiver platform is able to locally process signals. Appropriate external sensors provide monitoring of ECG, temperature, blood glucose, acceleration, heart rate and respiration. Data can be further filtered and processed by the application software installed at the recording PC.

Based on price, functionality and ease of use, the Equivital sensor belt was selected for this module.

Equvital Belt

The EQ-01 Sensor Unit [51] consists of the EQ-01 Monitoring Belt and EQ-01 Sensor Electronics Module (SEM). The Equvital monitoring belt features a system of fabric embedded, intelligent sensors to measure medical grade physiology in real time. The monitoring belt connects with the Sensor Electronics Module. Figures 3.26 and 3.27 illustrate the Equvital sensor unit. It allows for continuous monitoring and storage of information for a period of 24 hours. It has a range limit of 10 meters between the Bluetooth dongle and the monitoring device, using a Class 2 Bluetooth dongle. The SEM weighs just 75g, is robust and water proof. The SEM features embedded algorithms which allow intelligent real time processing of the wearer's vital signs.

Figure 3. 26: EQ-01 Sensor Electronics Module (SEM).

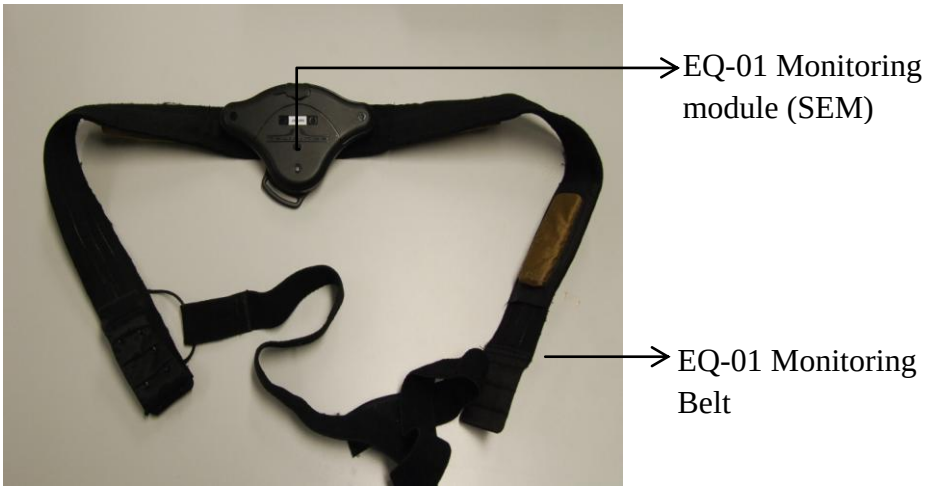


Figure 3. 27: EQ-01 Monitoring Belt and EQ-01 Sensor Electronics Module (SEM).

Key features of this SEM unit are as follows: Patient monitoring suitable for use with high levels of activity and perspiration, multi-parameter sensor, and a tri-axis accelerometer measures body position, impact and activity.

3.4.2 Data collection

The sensor unit is attached to a belt which can be worn by a patient (Figure 3.26). A Class 2 Bluetooth interface (transmitter and receiver) is provided with the sensor unit; the transmitter is linked to the sensor belt, and the receiver is located in the service module box on the robot. Once the robot gets close to the patient sensor data are transmitted to the receiver. Since each belt has its own identification number, the measured data can be related to the patient.

Chapter 4

MODULE SOFTWARE DEVELOPMENT

4.1 Introduction

The developed modules are to operate while attached to a mobile robot platform in a hospital environment, patrolling the hallways and patient wards for movement and report any unusual activity to the central terminal of the SWARM system, using the developed ORCA interface.

The software development and integration is one of the vital parts necessary to achieve the features of the final product of this research. All the code is written in C++ and tested on the Linux (Ubuntu) platform.

4.2 Module Communications

4.2.1 Orca Interface

As described in Chapter 3, in iWARD the physical data connection between a functional module and the robot base is through an Ethernet hot-plug connector. In order for a module to function correctly, each functional module should have the following sequence of operation:

- The module is inserted into an available slot on the robot. By doing this the module is physically connected to +5V and +12V DC power and to the data communication network of the robot.
- The embedded computer (Gumstix) boots up. The Linux operating system is loaded and starts the major components of the module software.
- The module registers in the Hardware Manager of the robot. This enables the update of the shared knowledge of the whole system, so the system is aware of which modules are available on which robots.
- If necessary, the module provides interface for other software components to communicate with the module.
- The module waits till the Mission Controller orders to start the module software.

- The module starts executing its main software. The operation is either terminated by the module itself, or it can be interrupted by the Mission Controller. In either case, the module software enters standby mode and waits for new instructions.

The robot system contains a lot of software components that function in parallel. In many cases, the components need to communicate between themselves, for example to provide data for other components. In iWARD, the principles of communications were laid down in a work package that dealt with the Communication Infrastructure, lead by two project partners (Fatronic and Fraunhofer IAO). It was decided that communication between software components would be through the Orca2 middleware software. In the Orca2 framework two components can only communicate if one of them provides an interface for the other (Figure 4.1). The second component then can use the functions of the provided interface to send/request information to/from the other module.

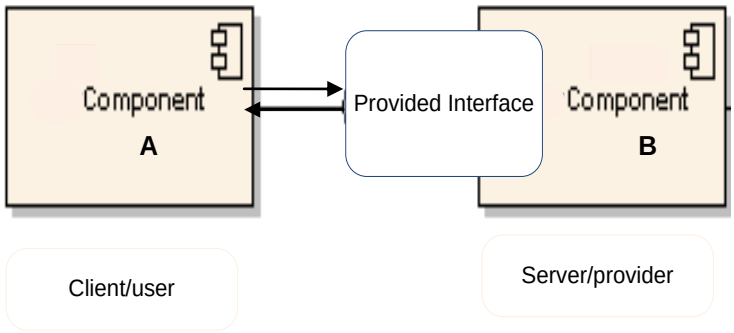


Figure 4. 1: Orca2 interface.

The software component of each functional module also needs to be able to communicate with other components. If they request information from them then they can use the interfaces provided by the external components. For example, the environment monitoring module may request the current position of the robot using the interface provided by the navigation component. However, if an external components needs to access the modules' software then each module component needs to provide the necessary interface for that. The template for such an interface in iWARD was developed by Fatronic, and the interface definitions were developed as part of the modules' software.

The main components that need to communicate with the modules' software (running on the embedded computers of the modules) are the Hardware Manager (HM) and the Mission Controller (MC). The module component uses the interface of the HM (running on the computer of the robot) to register/de-register itself when the module box is inserted into a slot on the robot. Since this is done live (when the robot is already functioning), each module operates in a plug-and-play manner. The MC (running on the computer of the robot) uses the modules' interfaces to order the execution/stop of the module programme and request information from them. Also, any external component that wants to access the module components directly needs to request the address of the module from the HM.

Another way of communication provided by Orca2 and used in the iWARD system is through *events*. The standard events handling capabilities are used to raise/process events. This method is used when urgent action is required by the system. An example is when the environment monitoring module detects smoke. The module software raises an event *Smoke detected* and the robot system reacts to it immediately.
The interface functions for each module are given below.

4.2.2 Module Interface Functions

An interface provided by a functional module is developed in two phases:

- Interface definition: using Fatronic's template, the interface functions and the data structure that they return are described.
- Interface implementation: the interface functions are developed as part of the module software.

The module interface definitions are given in Appendix A.

4.2.2.1 Basic Module

Certain interface functions are common to all functional modules, so they are defined in an abstract module called a *basic module*. It provides functions to operate the modules and defines the structure of events that can be raised by the modules.
Interface functions provided:

- **shutdown**: to shut down the module completely; required before removing the module from the robot.

- **reset**: to re-start a module; used if the module has a failure and the embedded computer needs to be re-booted.

Interface functions used (provided by external components):

- **doRegister** and **doUnregister**: these functions are provided by the interface of the HM and the modules use them to register or un-register themselves. Registration is needed when the module powers up, and un-registration before it is physically removed from the robot.

Event structures provided:

- Error events: they are raised if a module error occurs: hardware or software. The structure provides a detailed description of the error.

4.2.2.2 Environment Monitoring Module

Interface functions provided:

- **activate** and **deactivate**: power/unpower all sensors and start/stop taking measurements.

- **getLastEnvData**: provides the last set of measured environmental data together with the location of the measurement and the normal limits for that location.

Interface functions used (provided by external components):

- **getCurrentPosition**: requests the current position of the robot from the navigation component. It is needed to find out in which room the measurements take place.

4.2.2.3 Situation Recognition Module

Interface functions provided:

- **startPatientRecognition** and **stopPatientRecognition**: start/stop the software for recognising patient on the floor.

- **startIntruderRecognition** and **stopIntruderRecognition**: start/stop the software for recognising intruders in the hospital.

4.2.2.4 Patient Condition Monitoring Module

Interface functions provided:

- **startPatientMonitoring** and **stopPatientMonitoring**: start/stop taking measurements from patients.
- **getLastPatientData**: provides the last set of measured data from a patient together with the location of the measurement and the identificator of the sensor belt.

Interface functions used (provided by external components):

- **getCurrentPosition**: requests the current position of the robot from the navigation component. It is needed for behavioural analysis only, to find out where the patient was located when the measurements took place.

4.3 Environmental condition monitoring module

4.3.1 Driver software

The smoke, temperature and humidity detection sensors of the environmental condition monitoring module give analogue signals (voltage) as their output. These outputs are required to be converted into a digital form in order to be processed by the software. Hence a DAQ is utilised (described earlier) for this purpose. In order to receive data through the DAQ, a device driver had to be developed for it. The general settings for the RSIOADXR DAQ are given below.

Baud Rate DIP switches: DIP1: ON; DIP2: ON; DIP3: OFF => Baud rate 9600
Port A Pull-up/Pull-down jumper: Pull-down => floating inputs are '0'
Channels used: Port A, Channels 0-2: 10-bit analogue inputs for temperature, humidity, and smoke sensors.

According to the instructions of the DAQ manufacturer, no signal should be connected to the input port unless the device is powered. As a safety measure, a relay is connected between the +12V and Ground signals of the DAQ. Once a 12V DC power supply is connected to the

RSIOADXR, the relay's coil is powered and three N/O contacts of the relay connect the live wires of the sensors to the analogue inputs.

Following the above settings and guidelines for the DAQ device, the software driver was developed in the C++ programming language. The developed software is able to communicate through the serial (RS-232) port of the Gumstix computer and measure the sensors' output signals for further analysis. This software driver is compatible with the Linux (Ubuntu) operating system.

4.3.2 Module algorithm

The environmental condition monitoring software receives module activation commands through the above mentioned orca interfaces (section 4.2). To get the location ID (where the sensors' data to be measured), the module software also uses the Orca interface. Then the measured sensor data is compared with pre-defined values for that specific location and are stored for future reference. In case of any event (i.e. high/low temperature, humidity and fire/smoke), the module raises an event alarm which is shared with other SWARM components by publishing it through the orca interface. Figure 4.2 shows a flow diagram of this module's software algorithm.

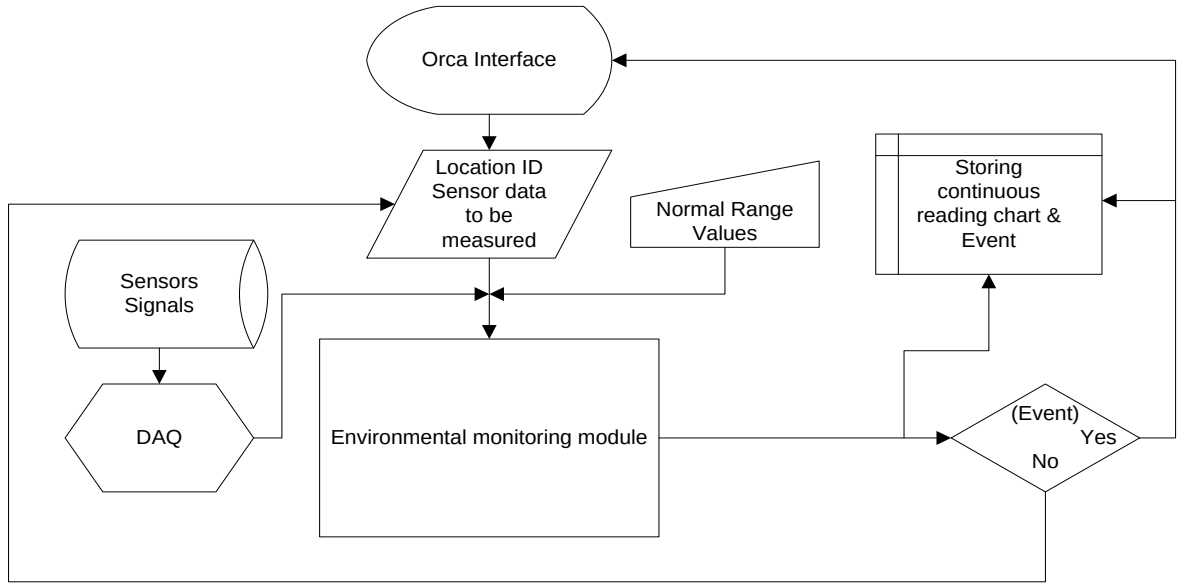


Figure 4. 2: Flow diagram of the environmental condition monitoring module software algorithm.

4.3.3 Module software

The iWARD Environmental Monitoring software actions are as follows:

1. Gets the location where measurements need to be taken.

2. Reads normal sensor values for this location from the data file.

3. Reads and converts each of the three 10-bit (0-1023) analogue signals of Port A (‘Temperature_Channel’, ’Humidity_Channel’ and 'Smoke_Channel’) to a voltage range of (0-5V).

4. Calculates temperature, humidity, smoke values.

5. Compares calculated values with the predefined values. An event is raised if measurements are outside the predefined ranges for the specific location. Then a module status log file is generated. More details are reported in chapter 5.

6. If at any stage the equipment stops working, the status of ‘Module_Active’ changes to ‘NOT_WORKING’ and an event is raised. All operation is stopped and only 'shutdown' or 'reset' commands are accepted from other components.

4.4 Situation recognition module

The situation recognition module consists of a camera and a distance measurement PMD laser sensor which are shared for both scenarios: intruder monitoring and patient on the floor monitoring. The following section includes the details of the driver developed for them. A DAQ device similar to the one used in the environment monitoring module is used in this module to get the PIR sensor signal according to the same principle as reported earlier.

4.4.1 Driver software

4.4.1.1 Camera

Four different video cameras were used in different stages of the module development period (described in Chapter 3) in this work. In order to obtain streamed images and to control the cameras, driver software had to be developed. The driver software details are as follows:

- ***Philips ToUcam Pro II***

Using the OpenCV library for grabbing images from the Philips camera is a straightforward task. Unfortunately OpenCV doesn't support the extended Philips webcam features like controlling shutter speed settings and gain control. Since lighting variations can cause undesired effects during visual tracking or colour segmentation, a Phillips web cam wrapper is combined with OpenCV in order to gain full feature availability.

- ***Canon VC-C4***

The 'PAN/TILT/ZOOM' mechanism of the Canon VC-C4 can be controlled through a serial port and a C++ driver has been developed for the Linux platform. A driver to control the Canon VC-C4 camera through a serial interface was developed together with function definitions for Orca to perform remote controlling of the camera.

- ***Axis 214 PTZ IP camera and Abus Digi-Lan TV7240 IP camera***

Both of these cameras are IP cameras that include a web server. The drivers, written in C++, were developed for the Linux platform using a cURL [265] interface. cURL is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP and so on) which is powered by the libcurl library for all transfer-related features. The driver for the Abus camera was developed jointly with one of the iWARD partners, Fraunhofer IAO.

4.4.1.2 PMD (3D) laser sensor

The O3D distance measuring sensor contains a 64x50 Pixel PMD-Matrix, which provides intensity and distance information of the scene for each pixel. The O3D provides two interfaces, one is an Ethernet connection for parameterising the sensor and obtaining data, the other is a process connection providing digital input/output lines and an analogue output. An Ethernet communication interface is used to get the sensor data. The communication with the O3D is based on two TCP/IP ports. The first port configures the sensor settings based on an XMLRPC server-protocol. The XML-RPC interface library may be obtained Online [266]. The second port transfers image-data based on a TCP/IP byte stream socket. Each picture is queried by sending a single byte to the sensor via this socket.

In order to provide access to the sensor's image data and to configure the sensor, an interface is provided using the XML-RPC protocol. It is necessary to create a XML-RPC client software to talk to the sensor's server software. A free XML-RPC library is available at [267].

4.4.2 Face detection

The situation recognition module (serving two scenarios: intruder monitoring and patient on the floor monitoring) requires a face detection technique. Early face-detection techniques focused on frontal human faces, whereas newer techniques attempt to solve the more common and complex problem of multi-view face detection. There are two common techniques, classifier based and skin colour segmentation-based, that are widely used in face detection [268,269, 270]. This section includes the description of both techniques which are used in the situation recognition module.

When the situation recognition module is activated, the first task for the vision system of the module is to load an image from the camera. Figure 4.3 shows input images of different real time scenes (i.e. varying distance, lighting conditions and orientation of the face relative to the camera) for demonstration purposes which cover both scenarios of the situation recognition module.

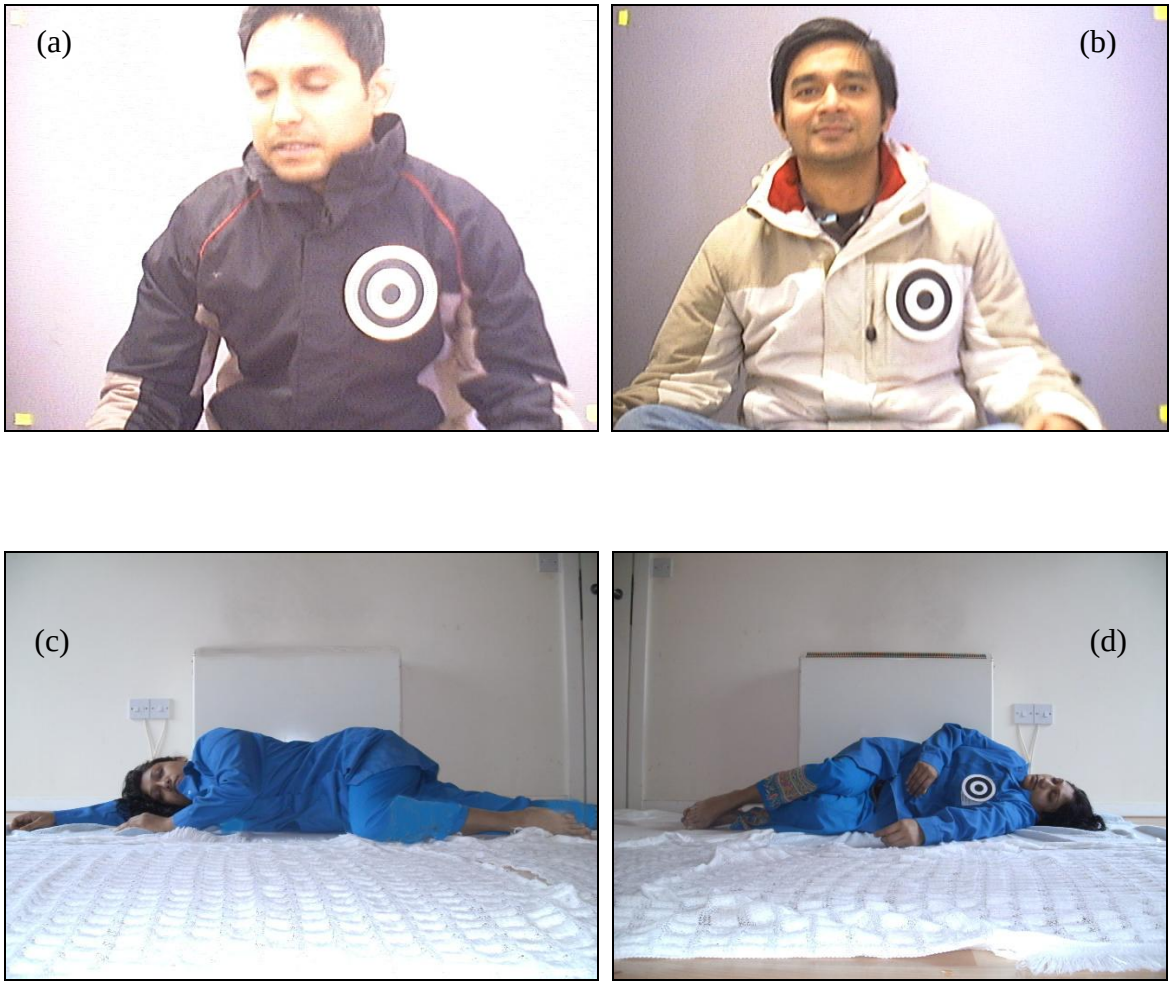


Figure 4. 3: Input Images of different real time scenes (a), (b) and (c) frontal face, (d) profile face.

4.4.2.1 Face Detection Using Haar-cascade and Region of interest (ROI) Body Area Selection

Adaboost [176-180] is a learning algorithm which constructs a strong classifier by attaching weak classifiers. Major properties include the optimisation of speed using cascades, combining the ability with any classifier to find weak rules (e.g. With SVM [64]). Haar [111, 127, 154], is one of the features which is used in the Adaboost learning technique (detailed in Chapter 2).

The computer vision library OpenCV (developed by 'Intel') [254] can be used for face detection. Its own trained facial haar cascades are used as the classifier. The face detection classifier [271] is trained by a large number of human faces to identify a face in a new image.

Haar-like features encode the existence of oriented contrasts between regions in the image (i.e. the class to be detected). A set of these features can be used to encode the contrasts exhibited by a human face and their special relationships.

The object detector of OpenCV has been initially proposed by Paul Viola and was improved by Rainer Lienhart [271]. First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundred of sample views of a particular object (i.e. a face), called positive examples, that are scaled to the same size, and negative examples - arbitrary images of the same size.

After the training of a classifier, it can be applied to a region of interest of an input image. The classifier outputs a "1" if the region is likely to show the object (i.e., face), and "0" otherwise. More details of classifier training are reported later in this chapter where this work presents a classifier for badge detection.

The module software searches for faces using the aforementioned face detection technique in images whose histograms have been equalised. The function will only exit when at least one face has been found. A circle is drawn around the largest face (Figure 4.4). The area and location of the face is used to estimate the region of the person's upper body, which is marked with a rectangle (Figure 4.4). The system needs to ensure that the rectangle does not exceed the size of the frame which is also used for setting a ROI for the badge contours. A scale factor is used to set the upper body size which is 'UBODY_WIDTH 4.0' and 'UBODY_HEIGHT 4.0' multiplied by the face radius.

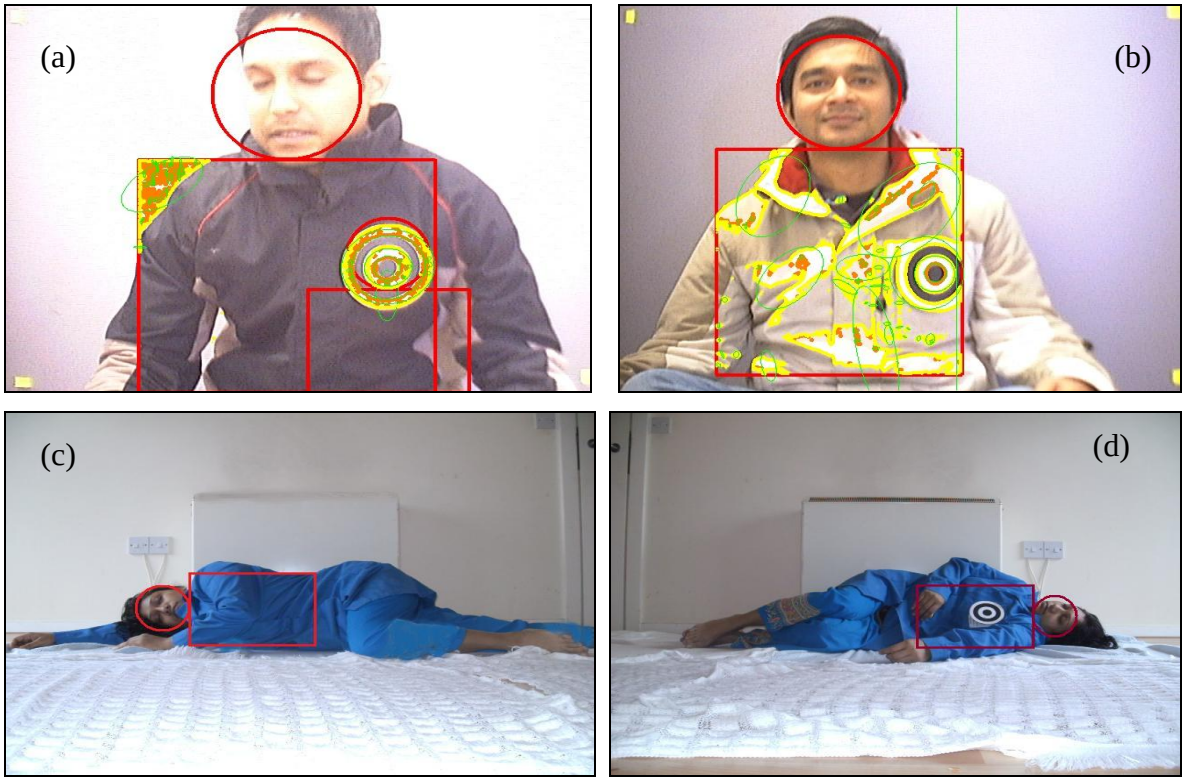


Figure 4. 4: Face detected and ROI selected for the input images shown in Figure 4.3.

Figures 4.4 (a) and (b) show detected faces and ROIs (vertically oriented) where (c) and (d) show detected faces and ROIs (horizontally oriented) relative to the camera. In the cases of the vertically oriented faces the ROI selection is easy because the body position should be below the face area. But in case of the horizontal orientation of a face (i.e. patient lying on the floor) the body area could be right (Figure 4.4- c) or left (Figure 4.4- d) to the face centre. Hence face detection is complemented with dress colour detection. Then the position of the detected dress (i.e. blue) is compared with the face centre and the ROI selection applied to the detected dress area according to the body proportion relative to the face. More details of colour segmentation techniques are reported in the next section.

4.4.2.2 Face Detection (Using Skin Colour Segmentation)

Human skin colour has been used and proven to be an effective feature for face detection [75, 87,129]. Although different ethnicities have different skin colour, various researches have shown that the most important difference lies mostly between their intensity rather than chrominance [272,273]. Different techniques [274,275] have been reported for locating skin

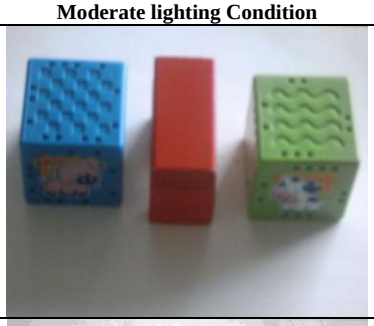
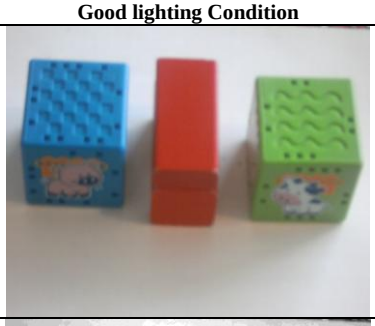
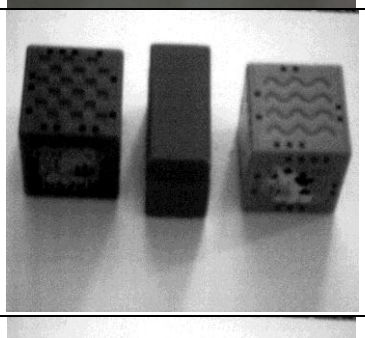
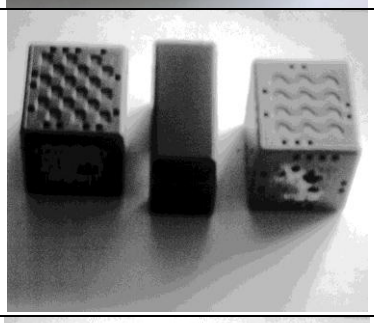
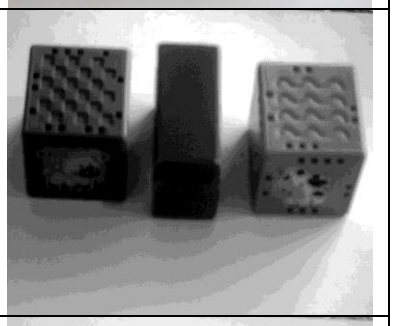
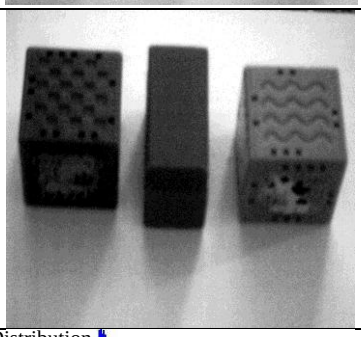
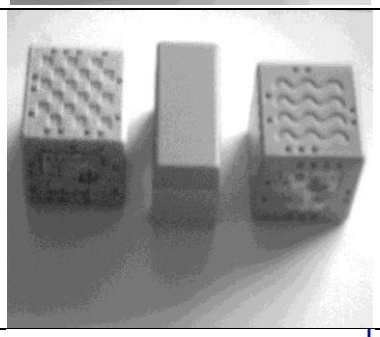
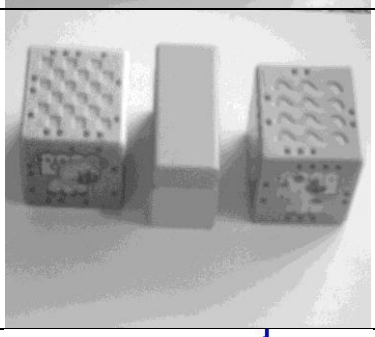
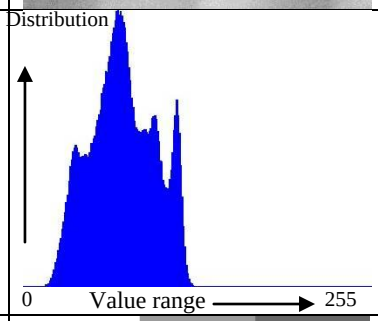
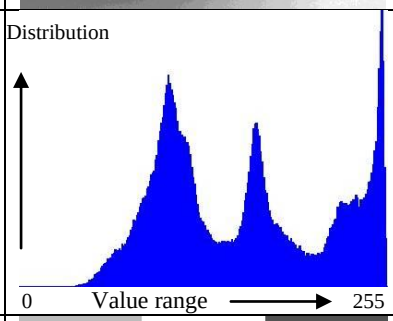
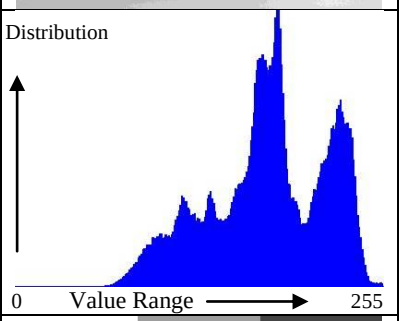
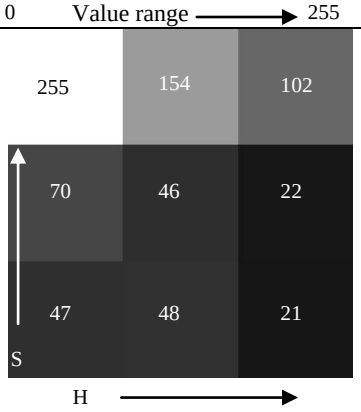
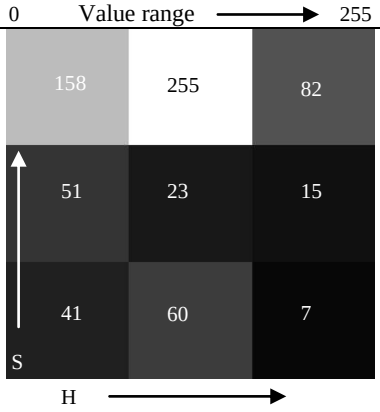
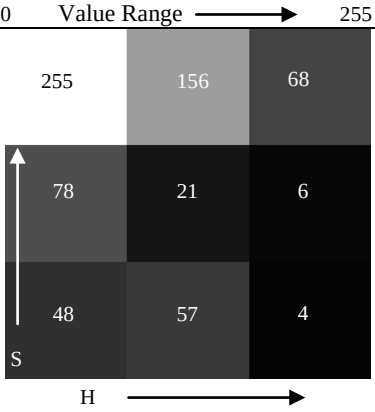
colour regions in the input image. While the input colour image is normally in the *RGB* format, these techniques generally use colour components in other colour spaces (i.e. *HSV* or *YCrCb*). The RGB colour space is highly correlated and the colour information within it varies non-linearly with any changes in illumination. Hue, Saturation and Value (HSV) was favoured here as it is perceptually closer to how humans distinguish colour, and uses better colour gradient information [276].

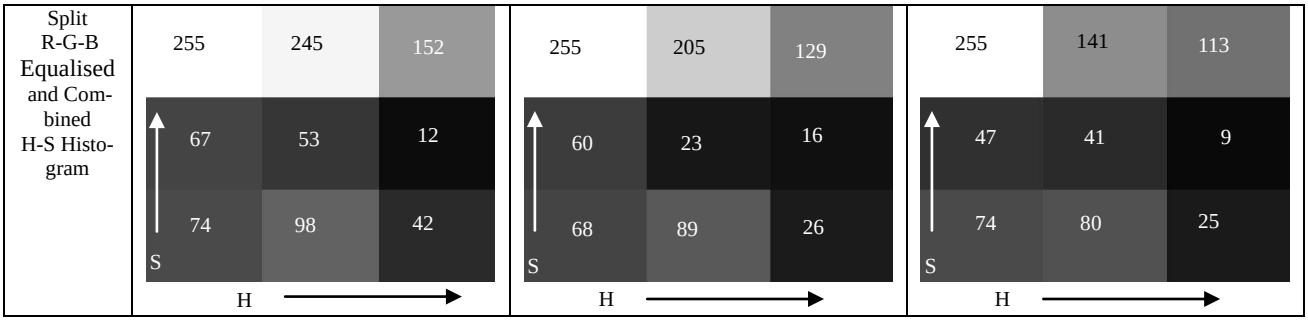
A test of H-S histogram comparison was carried to find the best colour space (RGB vs. HSV) for image processing. The experiment also proved that an equalised image gives much better results in the context of changes in lighting conditions compared to the non-equalised one. Colour histograms taken from a setting of three different colour (RED, GREEN and BLUE) cubes under various lighting conditions are shown in Table 4.1. The 1st row of Table 4.1 shows input images (RGB) in an indoor environment while lighting conditions were varied (low, moderate and good). In the 2nd row are shown the equalised images of the corresponding input images. The 3rd row shows the value (brightness) plane of the input images whereas the 4th row shows the histogram of the value (brightness) plane (the distribution shifts around somewhat as a result of the changing colour of the illuminating light). The 5th and final row show the H-S histograms of the input images non-equalised and equalised respectively where the vertical axis is S (saturation) and the horizontal axis is H (hue).

For histogram matching, this research used the chi-square method [253] which is based on Equation 3. For *chi-square*, a low score represents a better match than a high score. A perfect match is 0 and a total mismatch is unbounded (depending on the size of the histogram).

$$d_{\text{chi-square}}(H_1,\ H_2)=\sum_i \frac{(H_1(i)-H_2(i))^2}{H_1(i)+H_2(i)} \quad \text{... (3)}$$

Table 4. 1: Comparison between raw RGB and Equalised image in various lighting conditions.

Image	Low lighting Condition	Moderate lighting Condition	Good lighting Condition
Input (RGB)			
Equalised (RGB)			
Value Plane of HSV col-our space			
1D Histo-gram of Value Plane			
H-S Histo-gram of Input Image (RGB)			



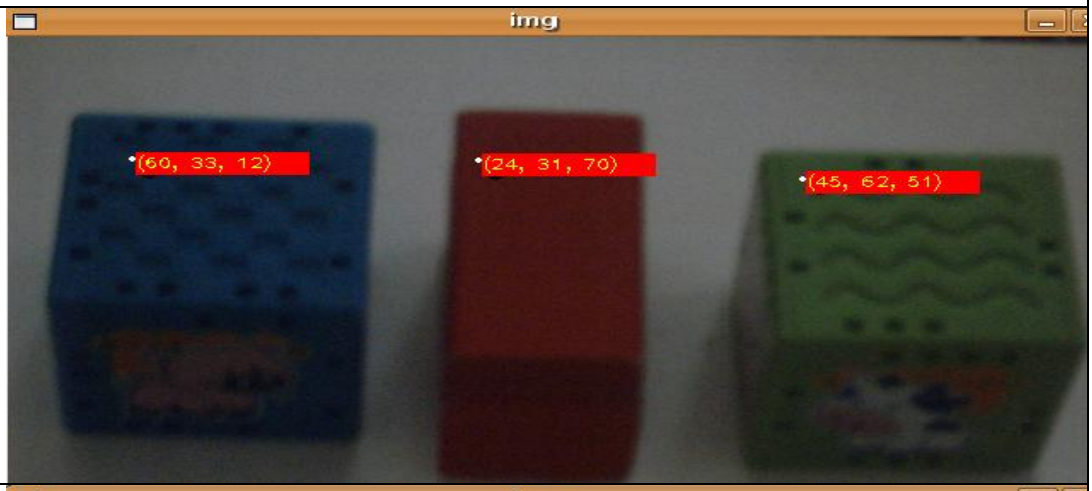
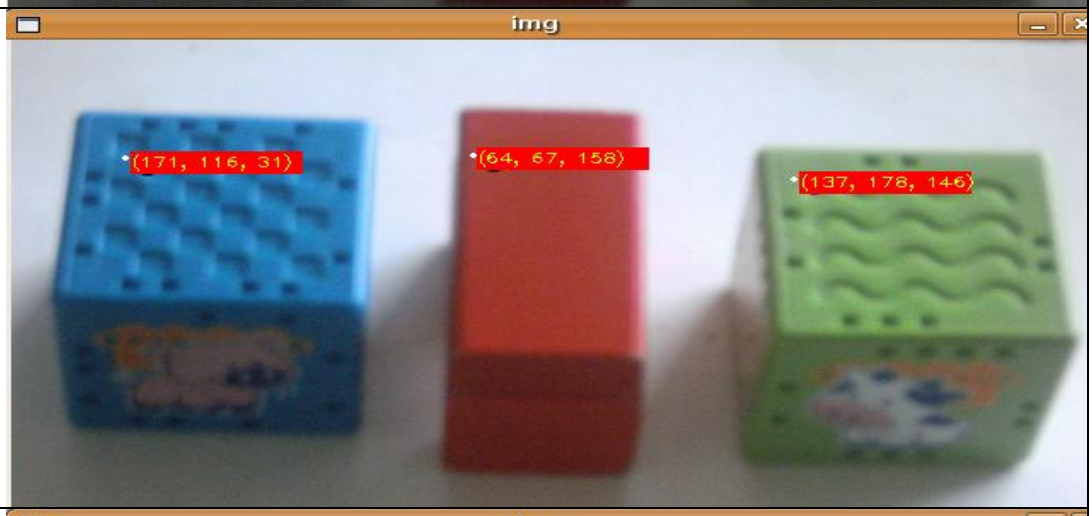
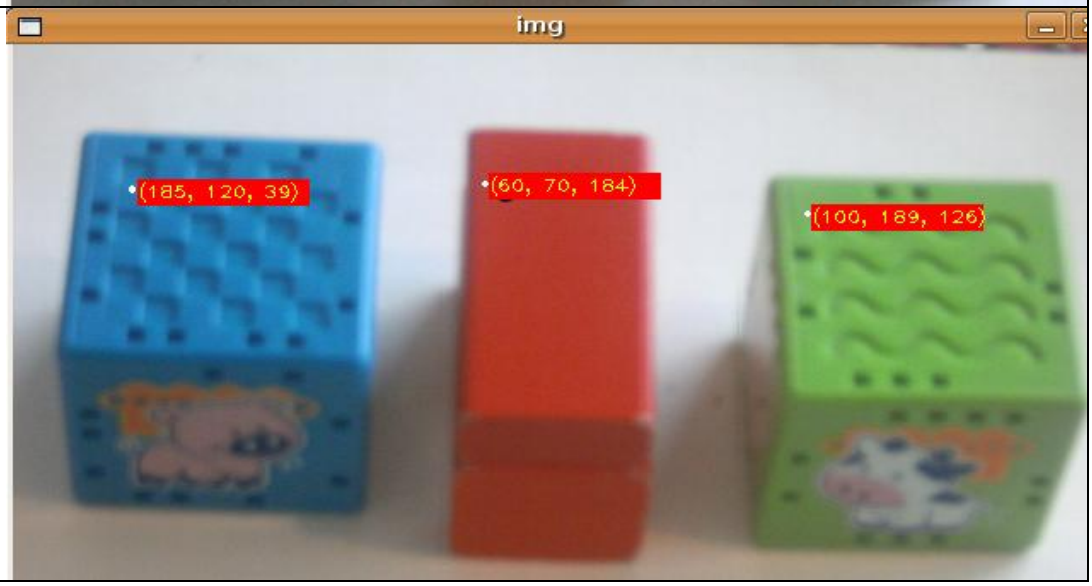
The histogram matching results for our experiment are shown in Table 4.2, which confirms that varying lighting conditions can cause severe mismatches in colour. It also confirms that equalised/normalised RGB (converted to HSV) works better than pure RGB in the context of lighting changes.

Table 4. 2: Histogram matching results (H1 = Histogram of low lighting condition, H2 = Histogram of moderate lighting condition and H3 = Histogram of Good lighting condition).

Comparison	**RGB (Non-equalised)**	**Equalised**
H3 vs. H1	0.062	0.023
H3 vs. H2	0.511	0.021
H2 vs. H1	0.112	0.015

RGB values of the same object change dramatically when lighting conditions are varied. Table 4.3 shows the BGR values comparison in various lighting conditions (white dots are showed as pick points). The first row of the table shows a setting of three different coloured (BLUE, RED and GREEN) objects in low lighting condition, where the following two rows are showing the same objects in moderate and good lighting conditions.

Table 4. 3: BGR values comparison in various lighting condition.

Lighting Condition	BGR (BLUE, RED and GREEN) values
Low	
Moderate	
Good	

To find colour blobs of a face or any other object, the system needs to convert the original colour image from BGR to HSV format so that the colours are easier to separate. An HSV colour conversion tool is used in this work to see which H, S and V values need to be threshold. Figure 4.5 illustrates an example of the Value component of the converted image of the original colour image (Figure 4.3- ‘c’ and ‘d’). Human skin colour diversity is highest in Sub-Saharan African populations [277], with skin reflectance values ranging from 19 to 46 (medium 31) compared with European and East Asian populations (skin reflectance values of 62 to 69 and 50 to 59 respectively) [278]. For experimental purposes, it was assumed that skin has a Hue value within 0 to 18 (out of 180), Saturation above 50, and Brightness above 80. Figure 4.5 shows the value ranges of the input image.

The output image of the skin colour detection function (Figure 4.6) is obtained after eliminating smaller blobs than a fixed predetermined value. This component can use cvInRanges() [253, 254] instead of cvThreshold() to set both a minimum & maximum threshold for each channel instead of just a minimum or maximum. It is possible to get a colour mask image (Figure 4. 7) of the segmented skin blob using cvCopy () [253, 254].

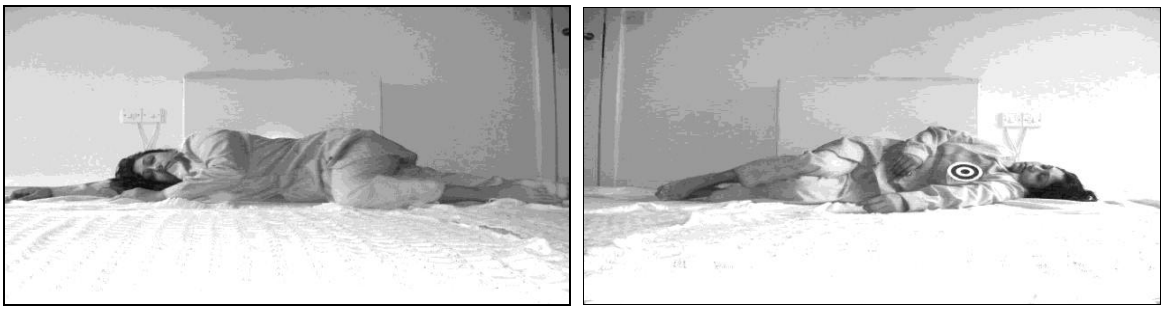


Figure 4. 5: Value components (from the HSV colour space) of the input images from Figure 4.3-‘c’ and ‘d’.

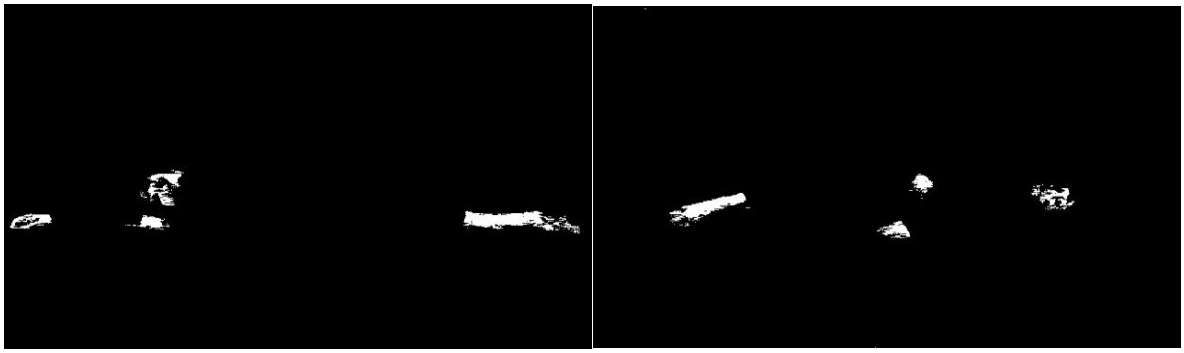


Figure 4. 6: Detected skin blobs (binary images) of the input images (from Figure 4.3-‘c’ and ‘d’).

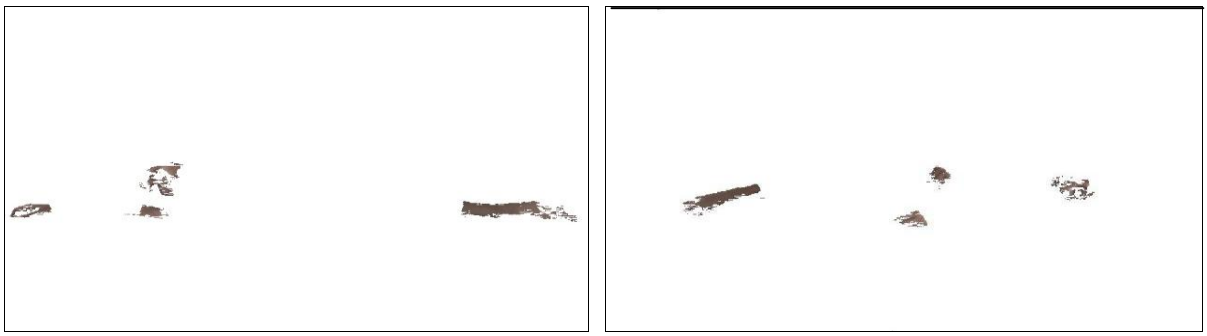
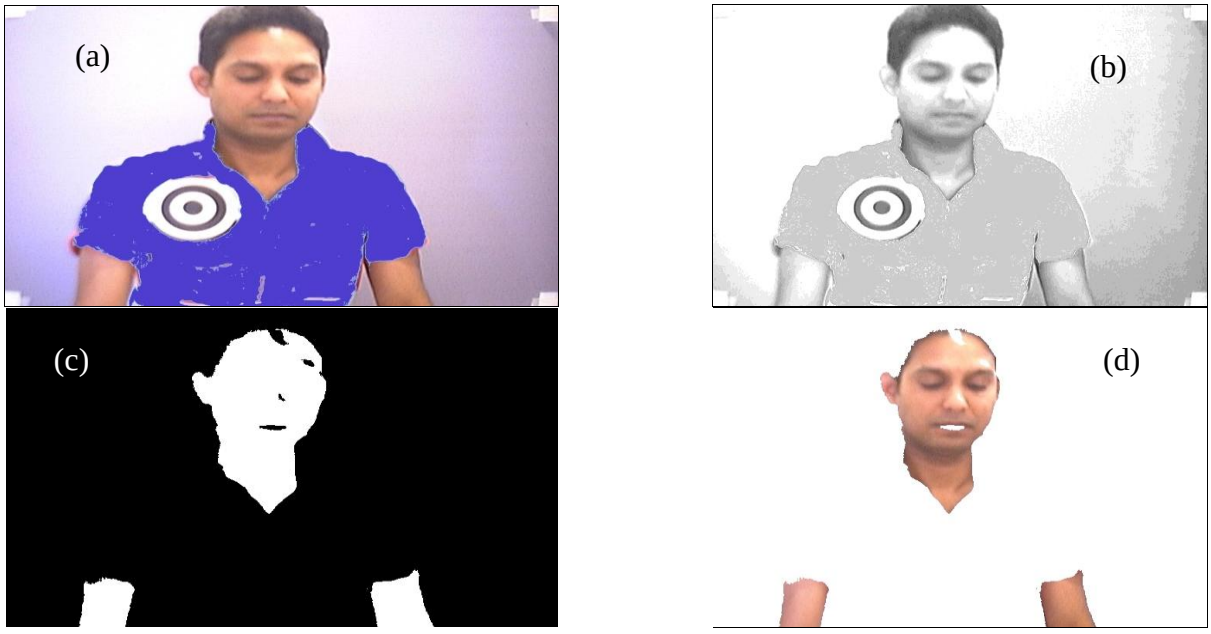


Figure 4. 7: Detected skin colour mask (RGB) of the input images (from Figure 4.3-‘c’ and ‘d’).

Figure 4.8 shows a general overview of the colour detection algorithm (skin and dress colour) steps and results. The dress colour (blue) was manually levelled to experiment with the HSV range of a true colour (i.e. BLUE: RGB = 0, 0, 255 and HSV = 240°, 100.0%, 100.0%). Background subtraction is performed by combining the detection of both skin and dress colour regions (Figure 4.8-g and h) and the resulting image is used by other components of this module (more details are reported later).



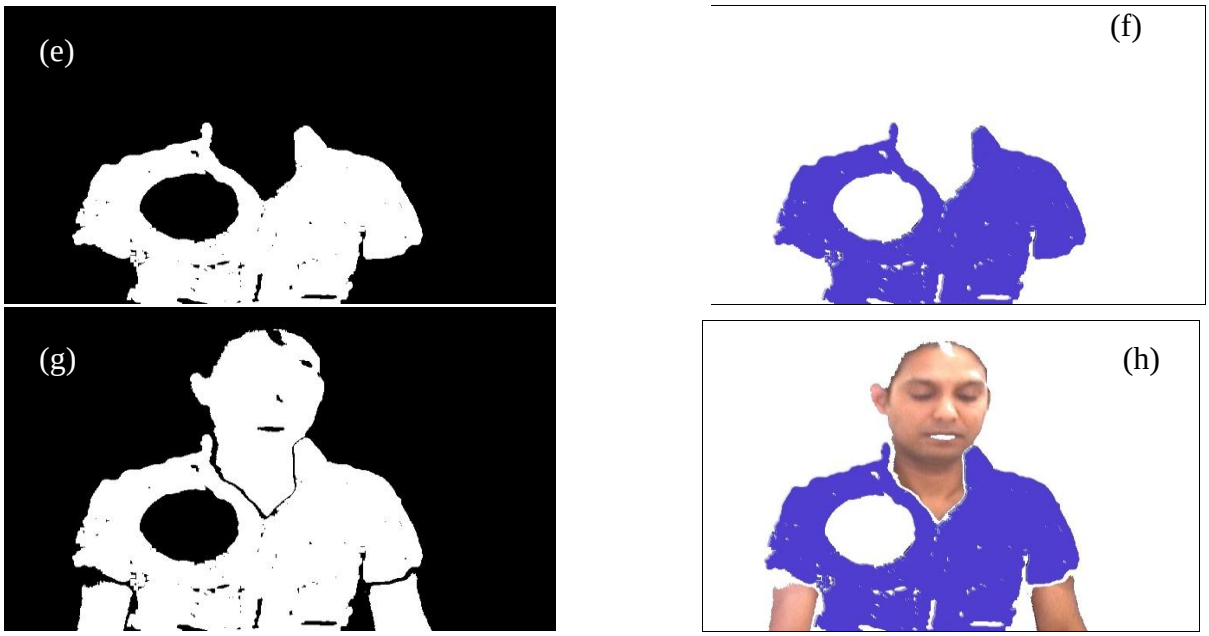


Figure 4. 8: General overview of the colour segmentation method; (a) Input image includes frontal face, (b) value component (HSV colour space) of the input image, (c) binary image of the detected skin blob used as a mask to extract the region from the original input image., (d) skin colour mask, (e) binary image of the detected dress blob, (f) colour dress mask, (g) binary image of the detected combined blob (skin and dress) of target foreground and (h) combined colour mask (skin and dress) of target foreground.

The skin colour segmentation technique only localises regions sharing the same assumed colour tones, so and many false positives can be obtained. Hence a simple algorithm is applied that calculates the largest area of the detected skin blob to be the face.

4.4.3 Badge detection

The situation recognition module also requires a badge detection technique which searches for a predefined badge using image analysis techniques within the input image frame. The badge detection function converts the original image into greyscale which is then equalised (Figure 4.9) in order to be robust to varying lighting conditions. The equalised image is then subjected to binary thresholding (Figure 4.10). In this research, adaptive thresholding is used: the threshold levels depend on the average light intensity of the image.

Figure 4. 9: Grey and equalised histogram-images for the input images.

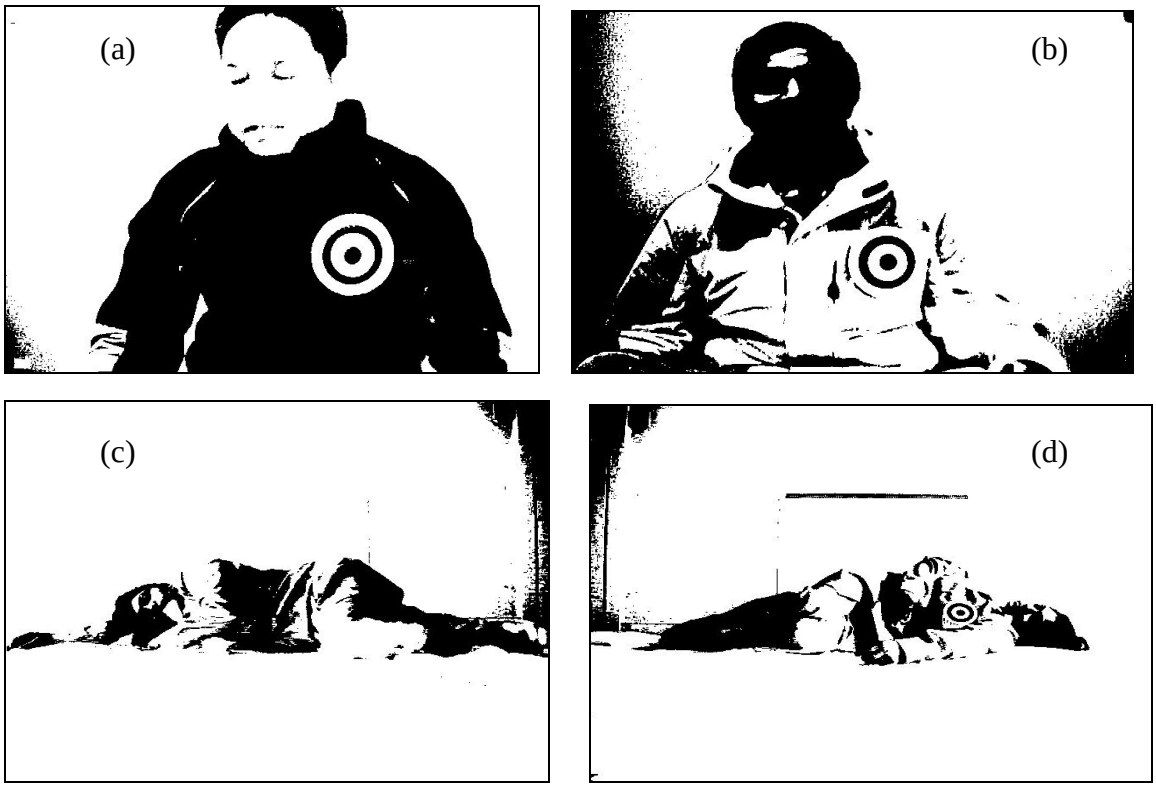


Figure 4. 10: Black and White (binary threshold) image for the input images (from Figure 4.3).

The following two approaches, contour analysis and classifier-based, are applied in this work to detect the badge.

4.4.3.1 Contour analysis approach

It is assumed that all authorised persons in the hospital are wearing identification badges. In this research, a simple identification badge consisting of several concentric circles with alternating black and white fill colour are used. It is especially useful as the centres of the concentric circles remain concentric even when the badge is looked at from an angle (not frontally). The contour-based badge detection function searches the upper body to find the concentric contours of the badge; the number of concentric contours required to identify a badge depends on the configuration of the actual badge. Using a badge with a large (above 4-5) concentric contours is recommended by experiments since it is highly unlikely that any other object can produce such a large number of concentric contours. A contour analysis algorithm is applied to detect the badge.

In this work, in order to find a badge in an image first all contours in the image are collected. Each contour is approximated by an ellipse that is the closest to the points of the contour using the least-square method. The centre of the ellipse is then calculated. Then a distance matrix is created that shows the relative distances of the centre of each contour relative to all other contours. A contour that is concentric with the maximum number of other contours is considered to be part of the badge. Then from the matrix it is possible to find all contours belonging to the badge (they are all concentric with the first contour). Concentricity is assumed if the distance between the centres of two contours is lower than a pre-defined value.

Structure of the contour centre distance matrix:

```
   0 1 2 3 4 5 6 7 8 9 ... <- contour number
0 X o o o o o o o o o ...
1 o X o o X o o o o o ...
2 o o X o o X X o o o ... <- Winner line (3 centres are close enough)
3 o o X o o X X o o o ... <- same as Winner line
4 o o o o o o o o X o o ...
.
.
```

Here ‘X’ means the distance between the centres of two contours is close enough. 'o' means the distance between centres of two contours is NOT close enough. Since each contour is compared to every other contour, there will be as many identical winning lines in the matrix as many concentric contours are found for the badge. Any of the winning lines can be used.

Figure 4. 11: Detected Badge Image for the input images shown in Figure 4.3; (a, b, d) contains detected badge contours (c) no badge found.

The ‘Find Contours’ function is applied to the ROI of the Black and White image (i.e. Figure 4.10). Figure 4.11 displays the resulting image containing the badges.

4.4.3.2 Haar-Classifier approach

As was explained earlier, the haar-classifier is widely used in various applications related to object detection (i.e. face, hand detection) [111, 127, 154, 271]. This work trained a haar-

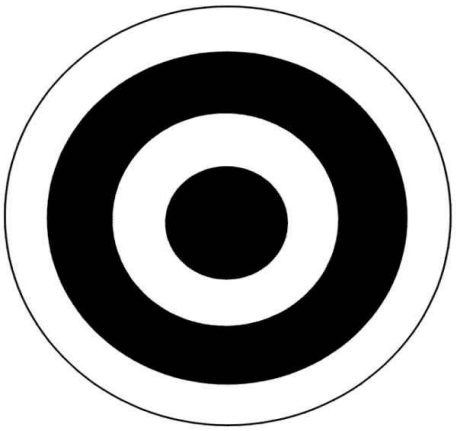
classifier for a predefined badge (i.e. concentric circles) using the OpenCV library [254]. The following section describes the procedure of the development of a ‘haar-classifier’.

(i)Data Preparation

(a) Positive (Badge) and Negative (Background) Images

To develop any haar-classifier, the system needs to collect positive images that contain only objects of interest, e.g., a badge (Figure 4.12-a), and negative images (Figure 4.12-b) that must not contain any objects of interest, e.g., any background image. Normally the ratio of the number of positive to negative images is 5:3 [279]. The accuracy increases the more sample images are used, however the training time period also increases (can take several days in many cases).

(a)



(b)

Figure 4. 12: (a) Badge sample image (b) Background image (contains no badge).

(b) Create training samples from one Positive (badge) Image

Although it is possible to create a large collection of positive images of the badge, this would be extremely time-consuming. Instead, it is possible to automate the process by using one image only and apply distortions to it: rotations in two directions, using various lighting conditions. However, the collection of negative images can not be automated as this would lead to reduced efficiency of the classifier system.

(ii) Training (Haar Training)

After collecting all positive and negative images, the badge detection classifier was trained by OpenCV's haartraining utility [254].

This process to generate a classifier for the target badge took around 67 hours. The classifier was stored in xml format.

(iii) Testing (Performance Evaluation)

The evaluation of the performance of the trained classifier was carried out by using OpenCV's performance utility. Figure 4.13 illustrates an output of the performance utility showing the total number of hits, the number of missed and false detections, and the weak classifiers.

```
+===================================+======+======+======+
|             File Name             | Hits |Missed| False|
+===================================+======+======+======+
|tests/01/img01.bmp/0001_0153_005|      0|      1|      0|
+-----------------------------------+------+------+------+
....
+-----------------------------------+------+------+------+
|                             Total|   874|   554|    72|
+===================================+======+======+======+
Number of stages: 15
Number of weak classifiers: 68
Total time: 115.000000
15
        874     72      0.612045        0.050420
        874     72      0.612045        0.050420
        360     2       0.252101        0.001401
        115     0       0.080532        0.000000
        26      0       0.018207        0.000000
        8       0       0.005602        0.000000
        4       0       0.002801        0.000000
        1       0       0.000700        0.000000
        ....
```

Figure 4. 13: Output of the performance utility of the badge detection classifier.

If a face is detected (either using the classifier or colour segmentation) and the upper body ROI is set (as detailed above), the badge detector looks for a badge in the input image (Figure 4.3). If a badge is found, a circle is drawn around it (Figure 4.14) and the system repeats the

algorithm using the next input frame. If no badge is found in the frame, the badge detector returns and stores the original input image which can be used by other software components, for example to raise an event in the intruder detection scenario.

Figure 4. 14: Badge detection using a classifier; (a, b, d) Badge detected and (c) No badge detected.

To test performance of both detectors (contour analysis and badge-classifier) a total number of 1000 image frames containing humans with badge and without a badge (670 and 330 respectively) were examined. Out of those sample frames 995 were perfectly classified using contour analysis approaches and 977 with the classifier. The classifier-based technique consumes much more time during the training session, and works with slightly less accuracy than the contour analysis approach.

Finding face and badge using any of the two described techniques (contour analysis or classifier) is the confirmation that a legal person is present in the frame, whereas the initial PIR signal only confirms the human occupancy in a certain range surrounding the mobile robot platform. In some scenarios the badge might not visible (i.e. Figure 4.3-c), and so the system would require to use other approaches/techniques to define a class for the detected object.

4.4.4 Sensor fusion

Combining colour cues (i.e. video image) with distance data (i.e. PMD laser sensor data) is a good example of sensor fusion that greatly increases the amount of information gathered from a scene. This sensor fusion approach is utilised here in order to increase the robustness of the system by complementing the visual sensing side of the project. Both data (colour and distance) are fused using a perspective image transformation method. The perspective transformation projects points onto the image plane along lines that emanate from a single point, called the centre of projection [253]. The OpenCV function cvGetPerspectiveTransform() [254] calculates a matrix perspective transformation such that:

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \texttt{mapMatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \qquad ...4$$

Where

$$dst(i) = (x_i', y_i'), src(i) = (x_i, y_i), i = 0, 1, 2, 3.$$

(dst = destination; src = source)

The PMD 3D Sensor (Figure 3.20) provides distance values for each pixel (in a 64 x 48 matrix) by measuring the time of flight of sent and reflected light. The nearest object is identified as the foreground using a background subtraction method and is overlapped with the 2D image (colour segmented image) to confirm that the subtracted object is a human or not. The fused image provides a 3D scene of the environment in real time and allows the

analysis of 3D shape and size to contribute to the object detection algorithm. By applying a threshold distance (robot to human minimum distance, say 0.5 m) a robot safety system can also be specified. If a human with no identification badge is too close (compared with the threshold distance value) to the robot then the system raises a self protection warning alarm.

The data file “Laser_Distance.dat” contains the 3D matrix information of the scene (i.e. x, y, z where x & y are the pixel co-ordinates and z is the distance value). This data file is generated with the developed PMD driver software discussed in section 4.4.1.2. When the situation recognition module is activated the PMD driver programme is called which retrieves the sensor data and stores it in a file for each frame requested by the situation recognition module. The depth information combined with the colour of each pixel makes image recognition much more efficient.

Figure 4.15 (a) shows the RGB input image for a particular instance. Figure 4.15 (b, c & d) are the images of intensity, distance and perspective view resulting from the PMD 3D sensor. The intensity image (Figure 4.15-b) visualises the distance measured by the sensor as a greyscale image. This mode is suitable for the following activities:

● Setting of the lens focus (by means of the setting screw on the back of the unit)

● Adjusting the position of the active image section.

The colour images in Figure 4.15-‘c’ and ‘d’ represents the distance value of each pixel in the frame.

For visualisation, the individual pixels are standardised across the entire image zone. Highly intense pixels are displayed as bright, whereas low intensity pixels are displayed darkly. If the dynamics of the sensor image (i .e. the difference between the "strongest" and the "weakest" pixels) is very high, the little reflected zones are displayed as very dark. Hence contours at a larger distance can often no longer be recognised. In such case the band width of intensities can be displayed as a logarithm which is almost like a reduction in contrast [261].

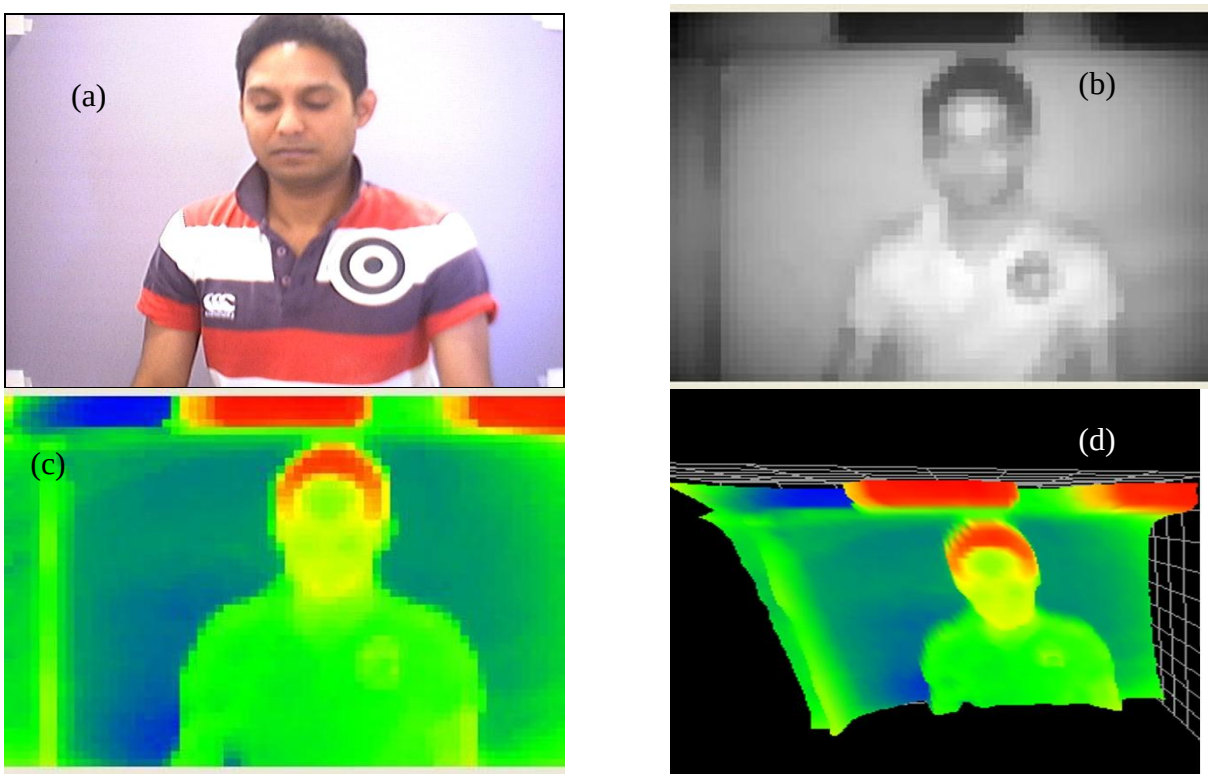


Figure 4. 15: (a) Video (RGB) input image, (b) distance value converted to Intensity image (PMD), (c) distance value displayed with a colour map & (d) perspective view of the 3D scenes.

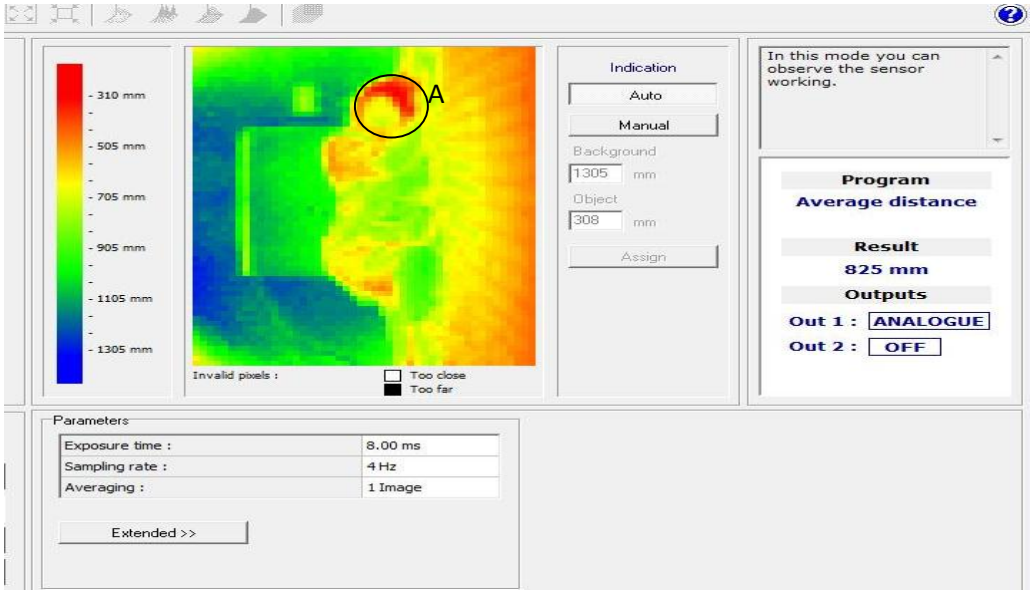


Figure 4. 16: Distance value displayed with colour map and index. A patient is lying on the floor and his image is vertically projecting the real time horizontal scene. 'A' marks the black hair of the human model.

4.4.5 Intruder monitoring

Intruder monitoring is one of the scenarios of the situation recognition module. The following section reports details of the situation recognition module algorithm involved with the Image processing which is able to detect an intruder in the hospital environment, mainly at night time.

4.4.5.1 Module algorithm

The intruder monitoring scenario is a combination of several components: face detection, badge detection, image fusion, and intruder detection). Figure 4.17 shows the flow diagram of the software algorithm.

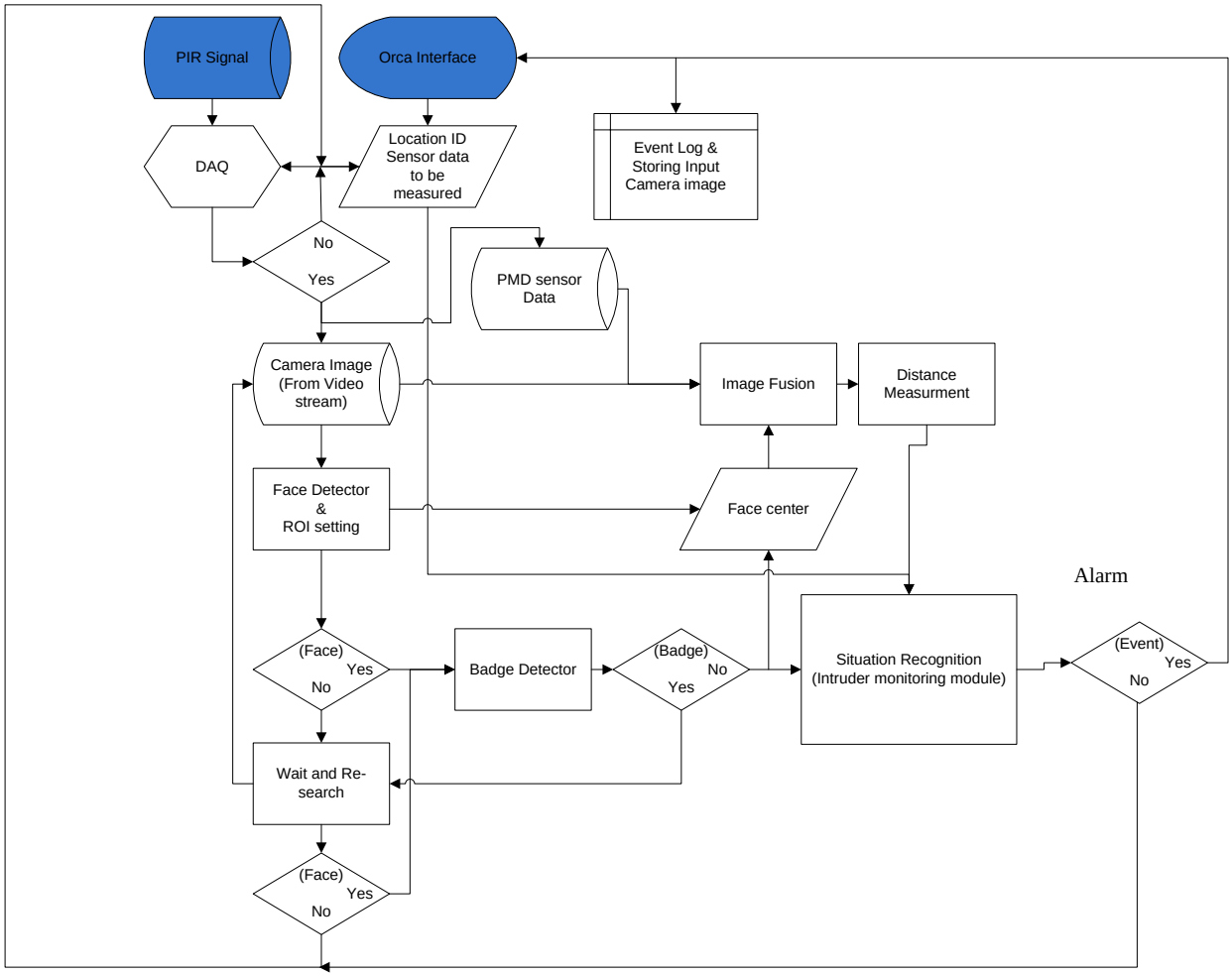


Figure 4. 17: Flow diagram (starts from blue blocks) of the Intruder monitoring module software algorithm.

4.4.5.2 Module software

The steps of the Intruder Monitoring scenario are as follows:

1. The mission scheduler component starts the Intruder Monitoring component. When started, the camera and laser sensor are not working yet. Most of the module hardware is in standby mode in order to save battery life.

2. The PIR sensor detects motion in the vicinity of the robot. This triggers the activation of all sensors (including the video camera and laser sensor).

3. The module obtains the location ID through the Orca interface.

4. An image is grabbed from the IP camera, and the face detector searches for a positive match. If no face is found the Pan/Tilt/Zoom mechanism of the camera is used to get a new image frame in the vicinity of the robot platform by changing the camera orientation and field of view. If a face is found, its size is measured. An upper body area is defined proportioned relative to the size of the face to select a region of interest.

5. At the same time the PMD 3D laser sensor is activated to provide distance information for each pixel of the colour image.

6. It is assumed that all authorised persons in the hospital are wearing identification badges. It consists at least 5 black and white concentric circles. The programme searches the upper body to find this badge.

7. If the badge is found the system waits till the identified person is able to leave the vicinity of the robot (so that the does not identify the same person again), and then the whole procedure is repeated again for the next image frame as long as the module operation is active.

8. If the badge is not found, an event 'Intruder Detected' is raised. The image of the intruder is saved locally. The event message includes the filename along with the received location ID through the developed Orca interface.

9. The distance in the middle of the detected face is calculated. This will be regarded as the distance between the potential intruder and the robot platform. If the intruder is closer

than the minimum allowable distance, the module raises a protection alarm. The module then returns to STANDBY mode until getting further positive signals from the PIR sensor.

Face detection is performed by the classifier-based approach (detailed in section 4.4.2.1) due to its reliability [271]. Both approaches of badge detection (contour-based and classifier-based, detailed in section 4.4.3) can be used in this module to detect black and white concentric badges in the input image. Combining the results of both approaches has increased the confidence level of decision making.

4.4.6 Monitoring patients on the floor

The situation recognition module is also responsible for detecting any patient who has fallen and is lying on the floor. It consists of the following components: Face detector, Badge detector, Colour segmentation, Image fusion, Background segmentation, Blob creator, and Decision making based on Artificial Neural Networks (ANN). It is worth mentioning that although within this work a haar-like classifier was developed to detect human bodies lying on the floor, the classification results have not been satisfactory. The drawbacks are high computational cost and very low detection rate. Instead of using a classifier, other methods have proved to be more robust.

4.4.6.1 Module algorithm

Figure 4.18 shows the flow diagram of the detection of a patient lying on the floor scenario module software algorithm.

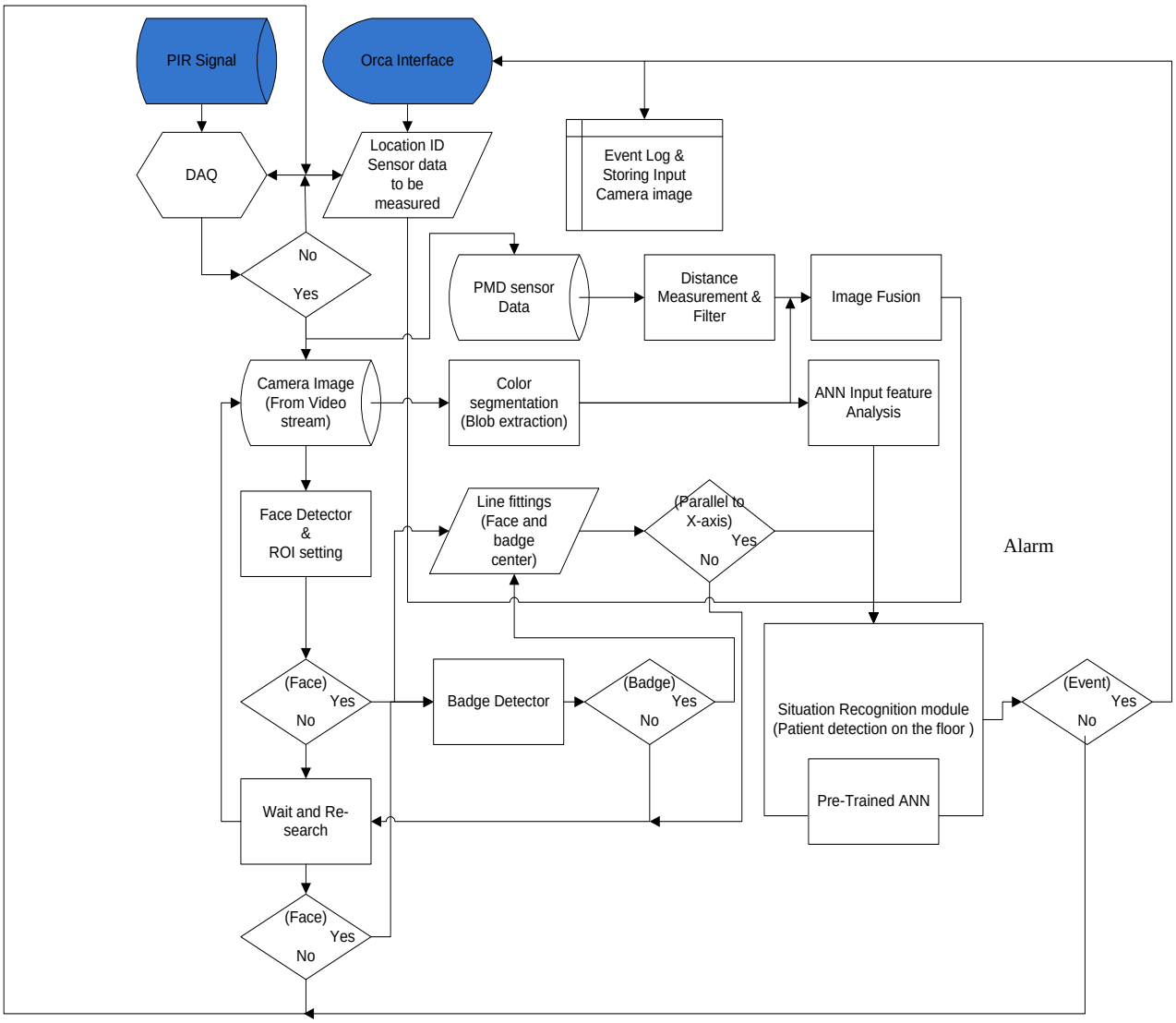


Figure 4. 18: Flow diagram (starts from blue blocks) of the detection of patient lying on the module software algorithm.

4.4.6.2 Module software

The PTZ video camera and the PMD 3D laser sensor are hardware shared by both scenarios (intruder monitoring and patient on the floor detection).

The steps of detecting a patient lying on the floor are as follows:

The first six steps are similar to those for intruder monitoring, all sensors are switched on immediately after the component is activated by the mission scheduler.

7. If a face is found in the current frame, an upper body area is defined proportioned relative to the size and position of the face. A region of interest (ROI) is marked on the upper body according to the algorithm mentioned earlier in this chapter. Then the badge detector searches for a badge within the ROI.

8. If the badge is in the ROI and the ROI is either on the left of right of the face then it is one of the indications to assume that a patient is lying on the floor. This step also draws a line between the face centre and badge centre (if any) to confirm the orientation of the patient body. The angle of the line θ defines the orientation of the body relative to the vertical direction. A near-90^0 angle (within a certain tolerance range of ±α is an indication that the patient is lying horizontally (Figure 4.19). The position of the line relative to ground level (D) confirms if the patient is lying on the floor. 'D' is calculated based on the current zoom level of the camera and the size of the face in the current frame.

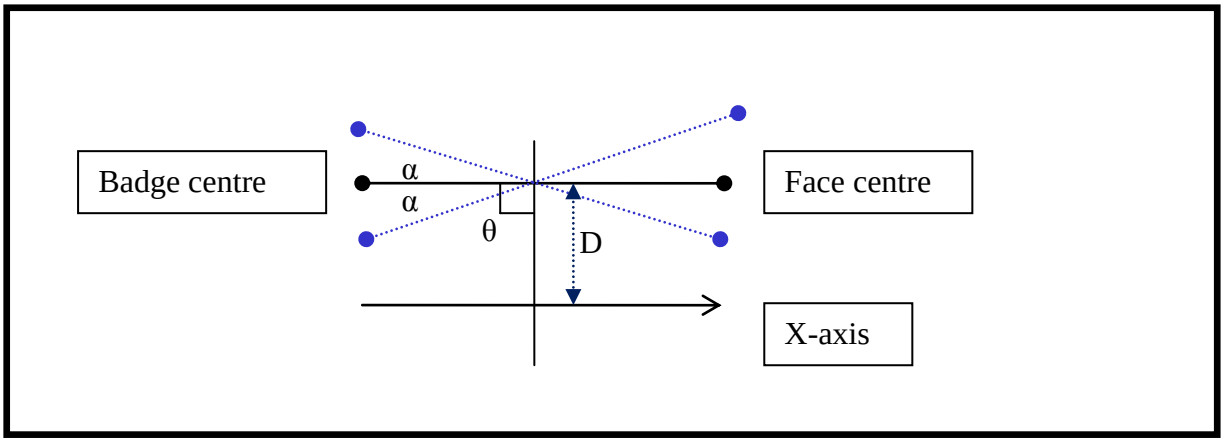


Figure 4. 19: Face and badge centre line orientation relative to the x-axis of the image frame.

9. If only either a face or a badge is found in the frame, its relative position to ground level is again an indication that a patient may be lying on the floor.

10. In some cases a face and/or badge might be occluded from the view of the camera, and as such only a segmented dress colour blob will be combined with the PMD sensor data. If neither a face nor a badge is found in the frame, other imaging techniques are required. For this, the software attempts to find colour blobs representing human skin and clothing in the frame. The colour segmentation component (using the HSV colour space) defines an eventual human body based on a patient's unique dress colour (i.e. blue) and skin colour. Once all skin

and clothing blobs are found, using the fused sensor data (colour and distance) the average distance of each blob is defined, and blobs with similar distances are grouped together. A combined blob with correct size (depending on the current zoom value and the size of the blob in the frame) is considered to be a possible image of a person in the frame.

11. To eliminate potential errors, the fused sensor data is used to separate background pixels that are clearly not part of the foreground object. An envelope is drawn around the foreground object (its size corresponds to the size of an average human), and all pixels that are outside this envelope are removed from the image. However, the truncated image still contains pixels that cannot be part of the foreground object. This is because the laser distance sensor cannot distinguish pixels that belong to the floor and pixels of objects that lie directly on the floor. By repeating the blob search technique on the truncated image, those pixels that are too far away in the HSV colour scheme from the skin and clothing colours are eliminated.

12. The resulting combined image is then analysed to retrieve the parameters (i.e. Freeman chain code, Differential Freeman chain code, Convexity defects and Hue-Saturation histogram) required for the artificial neural network (ANN) to obtain a classification of the detected object.

13. If a patient is found lying on the floor, the input image is locally saved and an event 'Patient lying on the Floor' is raised through the Orca interface to be shared with the other SWARM components. The event message includes the filename and location ID of the event. Then the Module returns to STANDBY mode.

4.4.6.3 Face detection and Colour segmentation

This work proposed two approaches for the automatic detection of human faces in colour images: skin segmentation to identify probable regions corresponding to human faces, and face detection classifiers. Human skin segmentation employs a model based approach to represent and differentiate the background colours and skin colours. Combining both approaches make this system more robust even in the presence of unique facial adornments like beard, spectacles and so on. Colour segmentation was briefly described earlier. This section describes the process of extracting a binary large object (blob) from an image.

1. Convert a colour image from BGR to HSV using.
2. Split the HSV colour image into its separate H, S and V components.
3. Threshold the image to look for pixels those are in the correct range of Hue, Saturation and Value (brightness) for the skin and unique dress colour separately.
4. Mask the image to create the target object blob.

5. Find the blobs in the thresholded image to get their sizes and positions. An artificial neural network will later be used to classify the object that corresponds to this blob.

To filter out unwanted noise from the output of this component, morphological operations are performed, such as a combination of smoothing, eroding, and dilating the image.

It is assumed that in each hospital, a unique colour dress is provided to the admitted patient and hospital staff by the hospital authority. For example, for the purposes of this research blue colour was selected. The only difference is that the HSV range is different for the skin colour and the blue range dress colour. The user can define their own dress colour pre setting the values of the HSV during the thresholding process. But it is recommended to avoid a colour (i.e. pink, brown) which might fall within the skin colour range. In this research, the blue range HSV values are as follows: lower range HSV(85, 100, 100) and upper range HSV(145, 255, 255). Figure 4.20 shows the binary image of a segmented dress colour and Figure 4.21 shows the colour mask image of the segmented blob of the input image in Figure 4.3- ‘c’ and ‘d’ (a model of a patient lying on the floor in different orientations relative to the camera position).

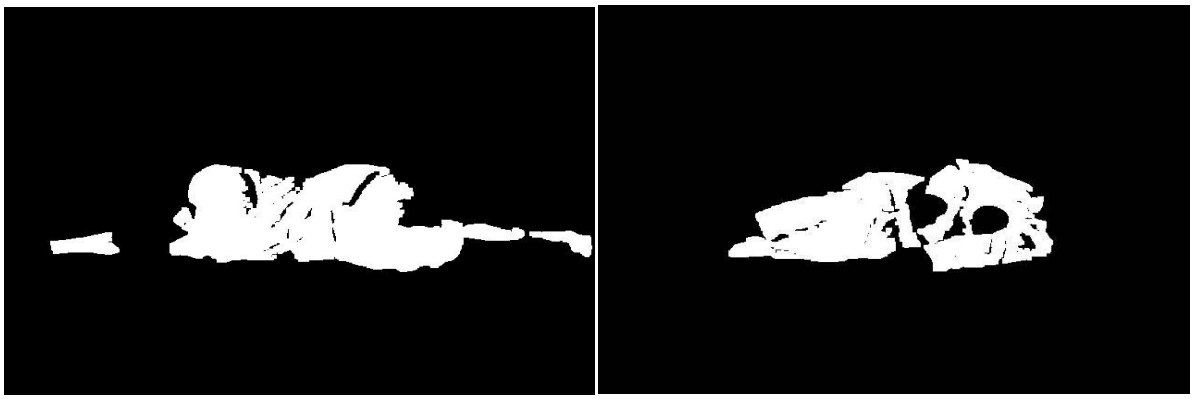


Figure 4. 20: Binary image of the segmented dress colour of the input image in Figure 4.3- ‘c’ and ‘d’.

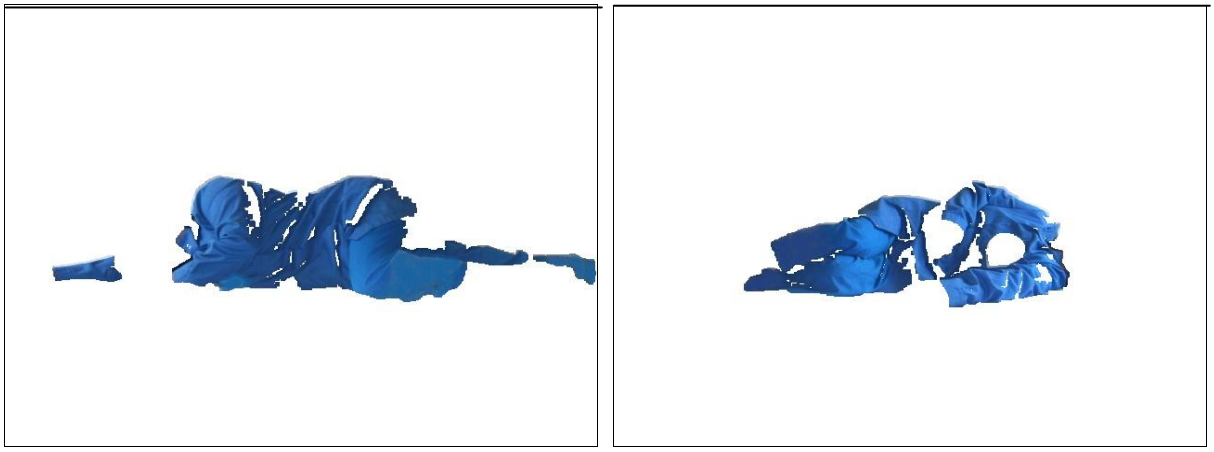


Figure 4. 21: Colour mask image of the segmented blob of the input image in Figure 4.3-‘c’ and ‘d’.

To get the blob of the full body, the blobs of the two colours (skin and dress) are combined together (Figure 4.22 and 4.23). This colour segmented blob will be used later by the two other components (Input image of the HS histogram and fused with PMD 3D matrix) of this module’s software, which is detailed later.

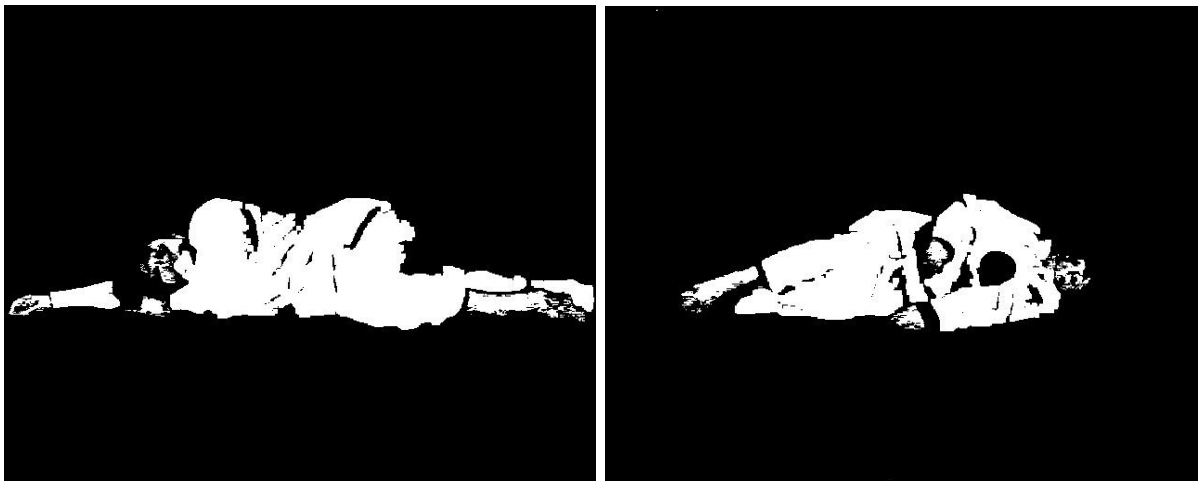


Figure 4. 22: Segmented binary blob of the full body (target object), where input images are from Figure 4.3-‘c’ and ‘d’.

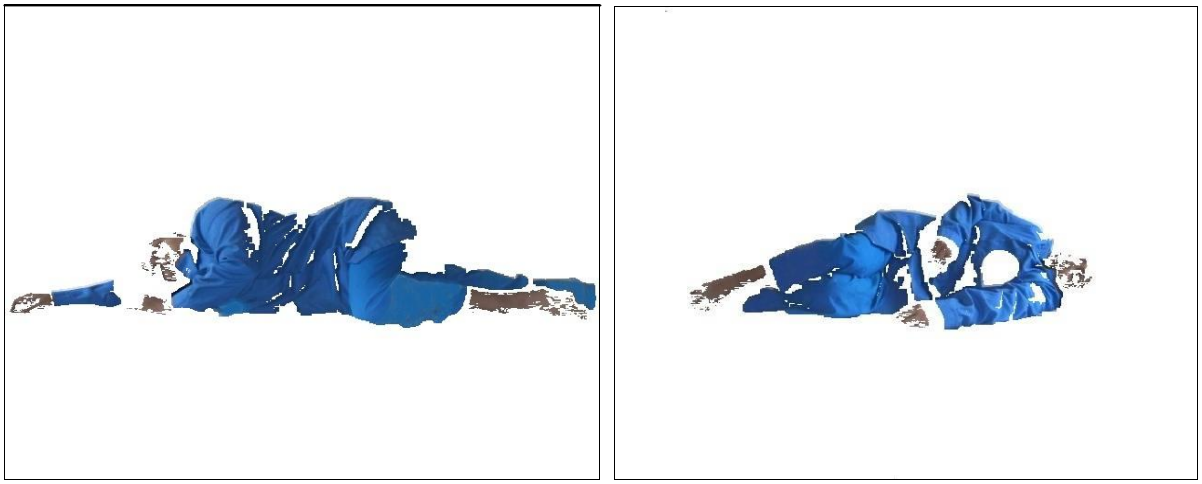


Figure 4. 23: Segmented colour blob of the full body (target object), where input images are from Figure 4.3-‘c’ and ‘d’.

4.4.6.4 Badge detection

Similarly to the intruder monitoring scenario, both methods (contour-based and classifier-based) for detecting a badge are used for the ‘patient lying on the floor’ scenario. This increases the robustness of the system.

4.4. 6.5 Image Fusion and Background subtraction (Blob extraction)

The PMD 3D laser sensor (Figure 3.20) provides distance values for each pixel; for sensor fusion, it is overlapped with the video camera (colour and 2D) image using the perspective image transformation method. A brief description of the procedure which adds depth (distance) information to a camera image using a laser scanner is as follows:

First, data from the PMD3D laser scanner is converted into a greyscale image. The distance values in the data file are converted to intensity values in the image. The data file contains 64 rows and 50 columns. It is assumed that the laser distances are measured in the Z direction. Since the laser sensor measures distances in polar coordinates, they need to be translated to Z-distances while creating the laser distance matrix.

The maximum ('maxDistance') and minimum ('minDistance') distances in the file are calculated, and are scaled to intensity values. The distance range 'minDistance' &

'maxDistance' is scaled into the range 1..0 (the closer the point the higher its intensity).

Since the laser image only has 64x50 pixels, for visualisation purposes this image is magnified 10 times.

Since the viewing angles of the camera and the laser sensor are different, and their positions relative to the viewed object are never identical, the images obtained from the two sensors are not identical. To fuse the distance sensor data with the camera image data, a perspective warp transformation is needed. This means that four corresponding points in both the camera and laser images need to be identified, and the camera image needs ot be stretched so that the corresponding four points coincide with the same four points in the laser image. Once the camera image is deformed to match the laser image, for each pixel in the camera image a distance value can be obtained from the laser distance matrix. Figure 4.24 illustrates the principle of the image fusion (2D and 3D) technique. In the figure, the four reflective yellow markers (in the four corners) are used to stretch the camera image on order to match the laser image.

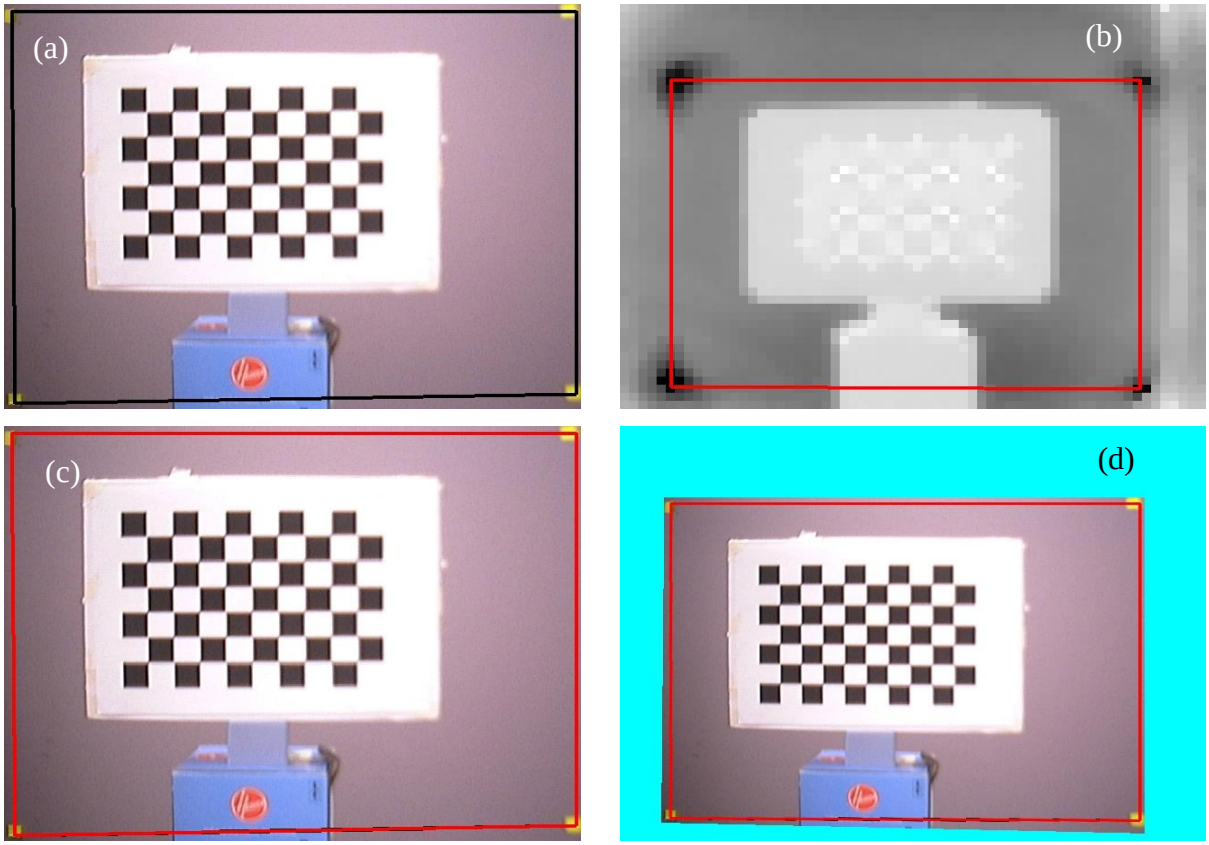


Figure 4. 24: Image fusion: (a) Original camera image, (b) Original laser (PMD 3D0 image, (c) Stretched camera image and (d) Perspective warp image.

The distance values are converted into an intensity image (Figure 4.25) which is magnified ten times to make its size similar to that of the camera image. Figure 4.26 shows a resized (64x50) camera image of the segmented full body blob. There are several interpolation methods included in OpenCV's in Resize() function. This work used bilinear interpolation. This resized image is used to filter out the background data of the laser image by applying thresholds to pixels. This approach (fusion of a 2D and a 3D image information) increased the robustness of the segmentation process. This process also includes distance threshold (described in section 4.4.6.2) to get the target object by filtering out the closest and farthest objects. Figure 4.27 shows a segmented laser image (resized to 640x500 pixels). Finally, Figure 4.28 shows the projected fused image which creates a real time 3D scene of the 2D RGB image. This study reports three possible points that might significantly improve the quality of the projected/fused image as follows: (i) minimise the reflectivity error (i.e. caused by PMD), (ii) use of another type of interpolation method during the resizing of the images and (iii) improve the point correspondence matching for the estimation of the warping transformation.

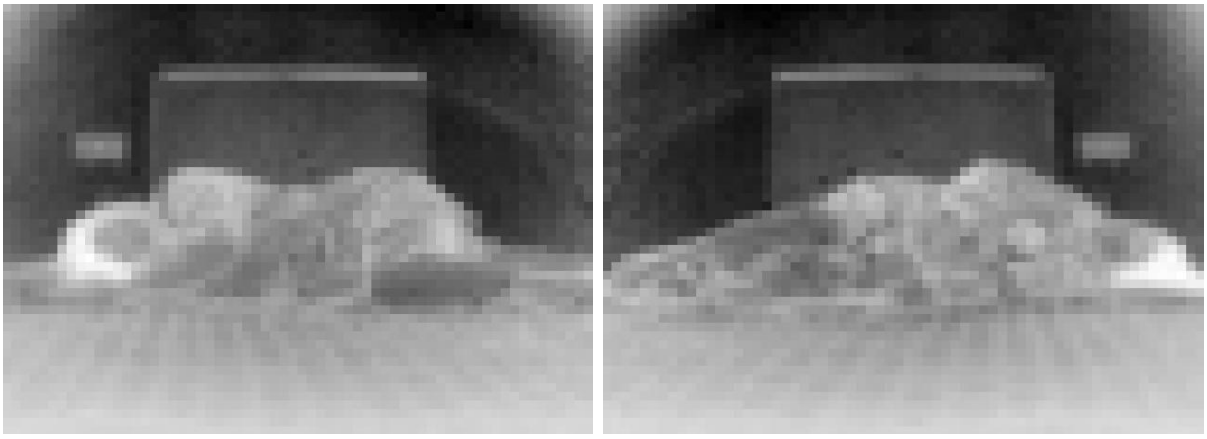


Figure 4. 25: Resized laser intensity image for two different orientations of the lying person, presented in figure 4.3-'c' and 'd'.

Figure 4. 26: Resized camera image of binary blob to match the original size (64x50)of the PMD image for two different orientations of the lying person, presented in figure 4.3-'c' and 'd'.

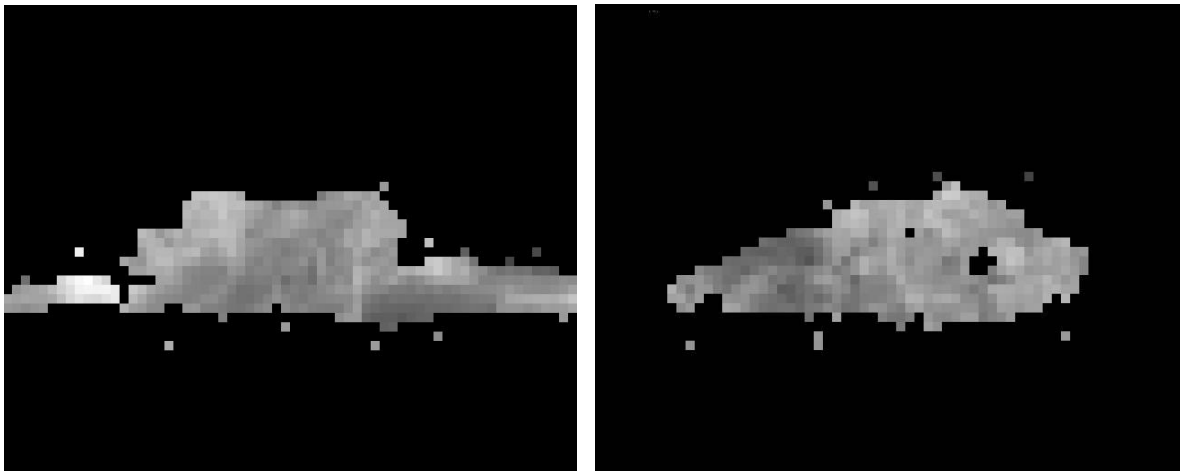


Figure 4. 27: Resized (640x480) laser image for two different orientations of the lying person, presented in figure 4.3-‘c’and ‘d’ after applying distance threshold.

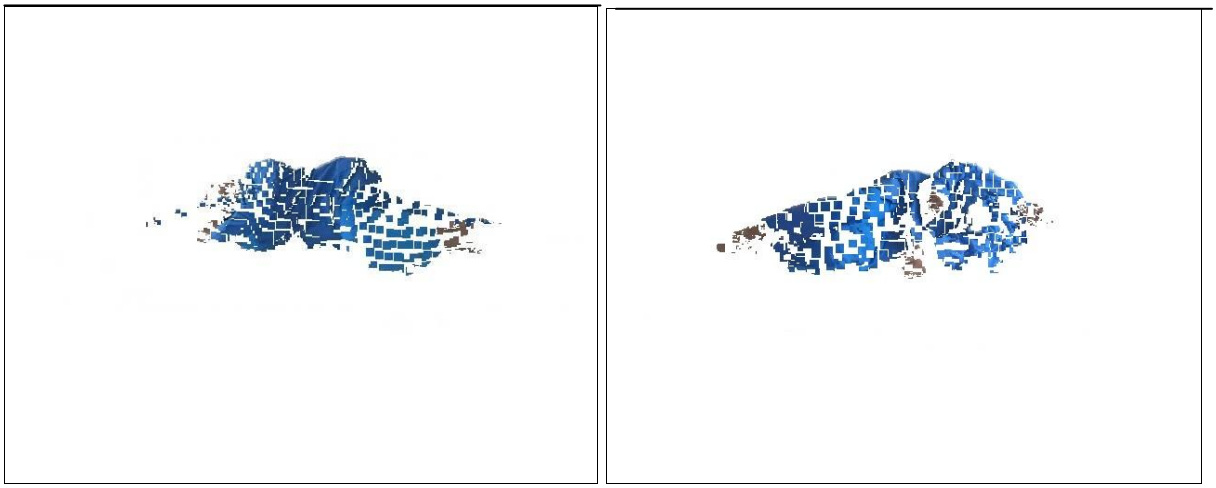


Figure 4. 28: Projected image (fusion of colour and distance) for two different orientations of the lying person, presented in Figure 4.2-‘c’ and ‘d’.

The background subtraction process creates blobs of the target object. The target object is labelled as the foreground. The created blob is then passed through the Blob Image Analysis component (outlined next).

4.4.6.6 Blob Image Analysis

This component combines the following image analysis operations: binarisation, Freeman Chain Code (FCC), Differential Freeman Chain Code (DFCC) Histogram, Convexity Defects

Histogram, Hue-Saturation (HS) Histogram. Each pixel of the blob image (scaled to 32 x 24 pixels to reduce the computing cost for the pixel base analysis) is converted to a binary image (i.e. white is 255 for integer pixels or 1 for floating point pixels, and black is 0) using the aforementioned average threshold values for different intensity levels. These binary values were used for as inputs to an artificial neuron network (ANN) for each sample. The final version of this software didn't use pixel-level data as input parameters due to its high computational cost and low performance. Each blob image (original size: 640x480 pixels) is also passed to a contour analysis algorithm to obtain the FCC, DFCC, Convexity Defects and HS histogram characteristics. Then all these characteristics are stored in a text file. They are accessible by the ANN. This ANN is pre-trained with an image database (manually labelled) which can successfully classify people (i.e. patients) lying (or not lying) on the floor. Results and analysis are reported later in chapter 5. Two types of samples (positive and negative) are essentials to train the ANN. More details about ANN are reported in the next section.

4.4.6.7. Positive Sample Data preparation for ANN

All positives sample need to contain a human body (patient for this current study) lying on the floor. Characteristics of each sample are obtained as follows:

(a) Contour Analysis

Image segmentation techniques can be divided as region-based and contour-based approaches. Contour-based approaches usually begin with edge detection (methods like canny, sobel and so on), followed by a linking process that seeks to develop curvilinear continuity [280].

Dynamic programming [281], relaxation approaches [282], saliency networks [283], stochastic completion [284] are the notable applications of the contour based approaches. Their drawback is that the decision of labelling an edge is taken prematurely [280]. To detect extended contours of very low contrast, a very low threshold has to be set for the edge detector. Hence this study filtered out the noisy contours using a smoothing operation and used distance thresholding to avoid unnecessary contours in the foreground image.

After thresholding, the final contours (see Figure 4.29) are obtained by passing the binarised image through the contour finding function.

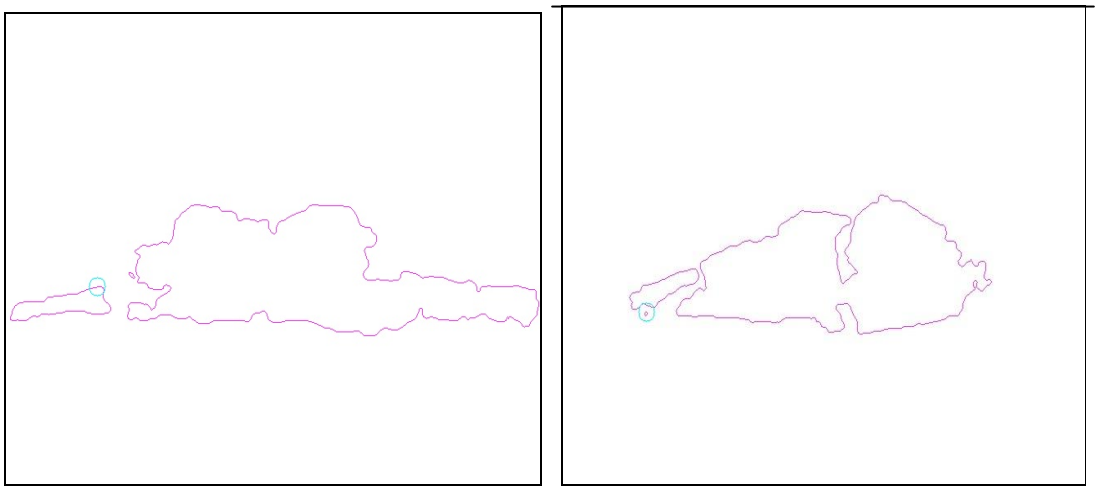


Figure 4. 29: Filtered blob contours of the input images presented in Figure 4.3-‘c’ and ‘d’.

After extracting the contours of the segmented blob this work applies the following analysis of the contours: Freeman chain code (FCC), Differential freeman chain code (DFCC), Convexity defects. For each case a histogram is generated to represents their values. The following section describes each of them individually.

(i) Chain code

Chain codes are one of the shape representations which are used to represent a boundary by a connected sequence of straight line segments of specified length and direction. This representation is based on 4-connectivity or 8- connectivity (Figure 4.30-a) of the segments [285]and this research used the latter one due to its robustness while being able to handle complex shapes. Chain codes have been claimed as one of the techniques that are able to recognize complex shapes of different objects (i.e. characters and digits) successfully [286]due to a number of advantages possessed by this technique as listed by [287].A chain code is a complete representation of an object which means that a system can compute any shape feature from the chain codes. According to [288], chain codes are a lossless compression of monochromatic images which preserves all topological and morphological information. The technique provides another benefit in terms of speed and effectiveness for the analysis of contours. Freeman chain code [289] and Directional Freeman Chain Code of Eight Connectivity (DFCCE) [290] are two different popular methods among others such as Vertex Chain Code (VCC) [291], Three Orthogonal symbol chain code (3OT) [292].

With a Freeman chain [289] (Figure 4.30), a polygon is represented as a sequence of steps in one of eight directions; each step is designated by an integer from 0 to 7 (0 – Up, 1 - Up-Right, 2 – Right, 3 - Down-Right, 4 – Down, 5 - Down-Left, 6 – Left, 7 - Up-Left). Freeman chains have useful applications in recognition and other contexts.

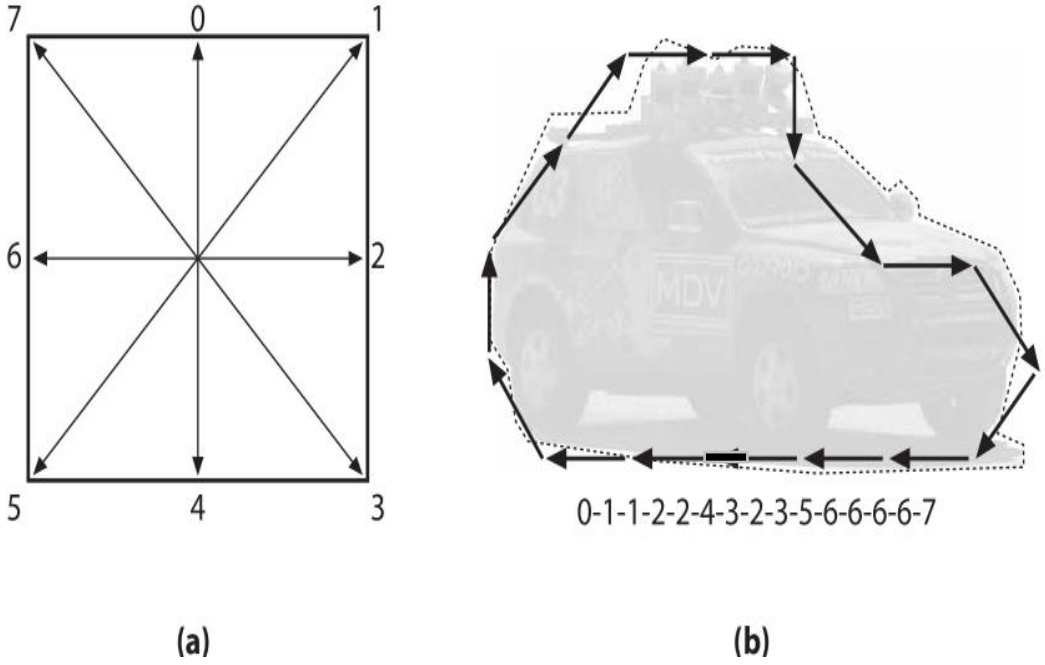


Figure 4. 30: (a) Freeman chain moves are numbered 0–7; (b) contour converted to a Freeman chain-code representation starting from the back bumper [253].

Comparing Freeman chain codes directly is a very time-consuming process and is error-prone. Instead, a histogram (the movement-code distribution) can be used. However, although a Freeman chain code gives a good representation for the general shape of an object, its histogram fails to distinguish variations of it. For example, the three shapes in Figure 4.31 have different chain codes, but their histograms are identical (the total number of elementary movements in each direction is the same across the three shapes).

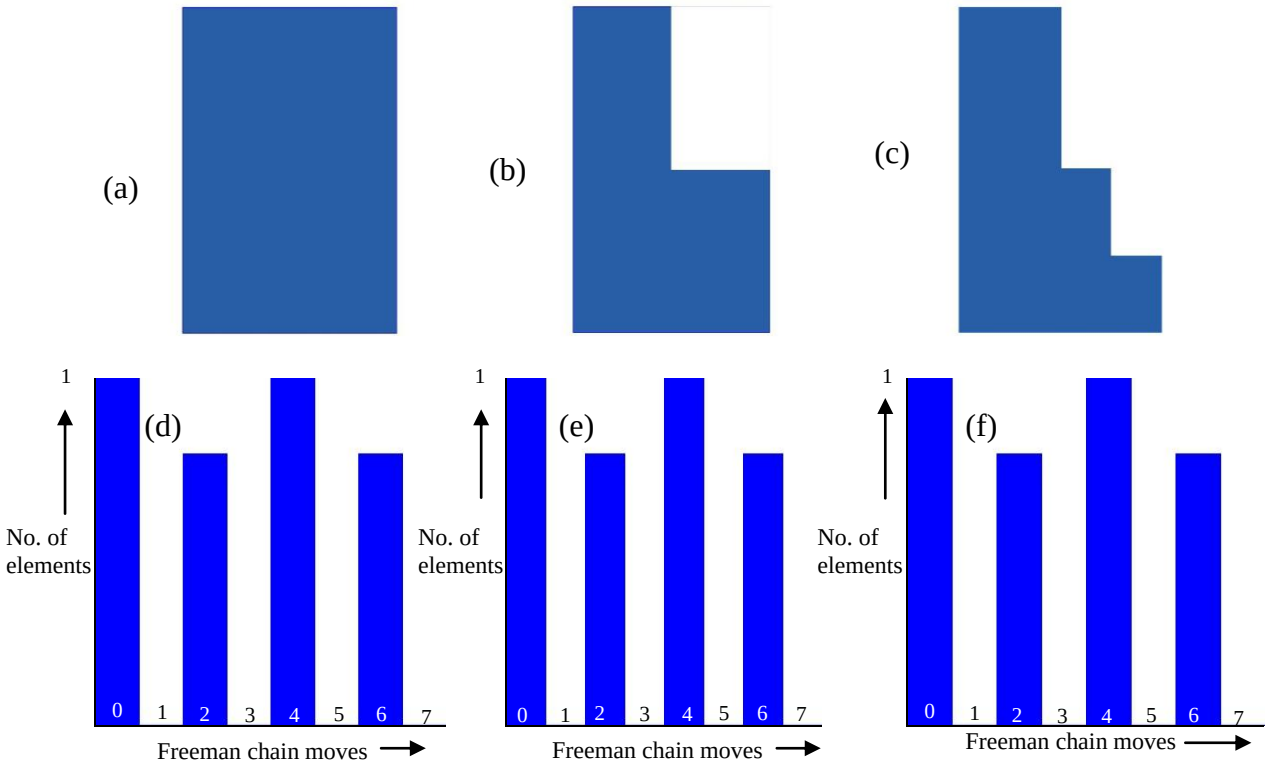


Figure 4. 31: Three differently shaped rectangular objects (a, b and c) with identical Freeman chain code histograms (d, e and f).

In order to eliminate this problem, a new approach is suggested in this research. Apart from the original chain code histogram, a Differential Freeman Chain Code (DFCC) is also applied. The elements of the differential chain code are not directional vectors but directional changes. Each elementary movement is compared with its predecessor. The directional changes can be one of the following: same direction, left turn by 45^0, left turn by 90^0, left turn by 135^0, right turn by 45^0, right turn by 90^0, and right turn by 135^0. Although this code can work with contours with sharp, distinguishable edges, code generation may be affected by the smoothing process or by other factors. A modified version applies a weighting factor to each directional change: the larger the turn, the higher it weighting factor. A 45^0 turn count as 1, a 90^0 turn count as 2, and a 135^0 turn count as 3. In the modified differential chain code there are only three possible values: same direction, left turn, right turn. An additional value can sum all motions that have a changing direction. Figures 4.32, 4.33 and 4.34-‘b’ include the 'same direction' and 'different direction' values. However, if a shape has long straight edges the 'same direction' bin contains a much larger number and the other bins are hardly visible in the histogram (Figure 4.32, 4.33 and 4.34-‘c’), which may make the contents of those bins

unimportant for the neural network. This would be a substantial error as the directional changes would not take part in the training process. Hence this work introduced another type of histogram (Figure 4.32, 4.33 and 4.34-d), which only counts the changes of direction but not the ‘same direction’ values. Figures 4.31-4.34 present the difference between the traditional FCC and the proposed DFCC for the above mentioned three different input shapes (from Figure 4.31).

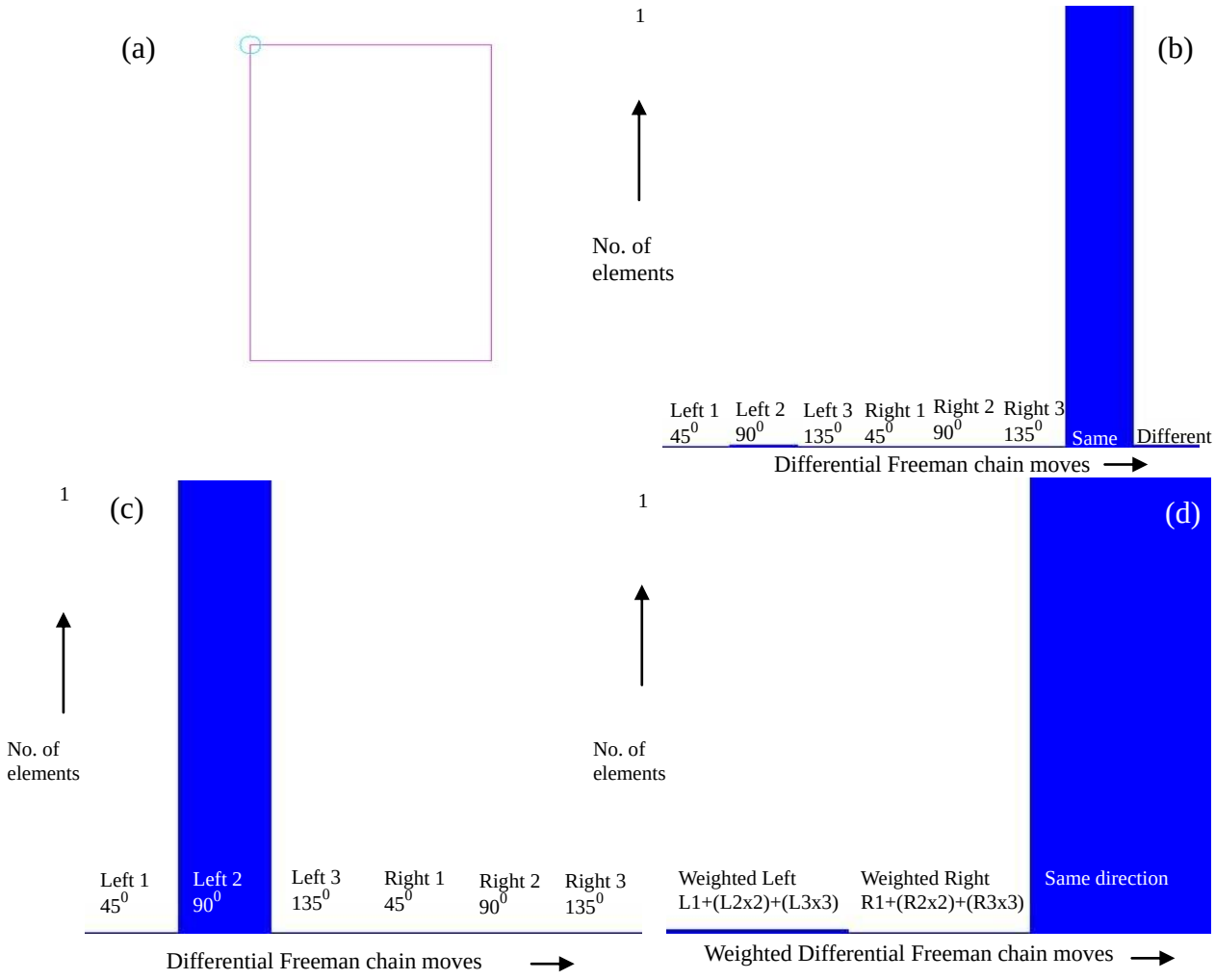


Figure 4. 32: (a) Contour of the rectangular shape object, (b) DFCC (including 'same directions' and 'different' directions), (c) DFCC (excluding changing in the same direction) and (d) DFCC (sums of all left and right turns in a weighted manner).

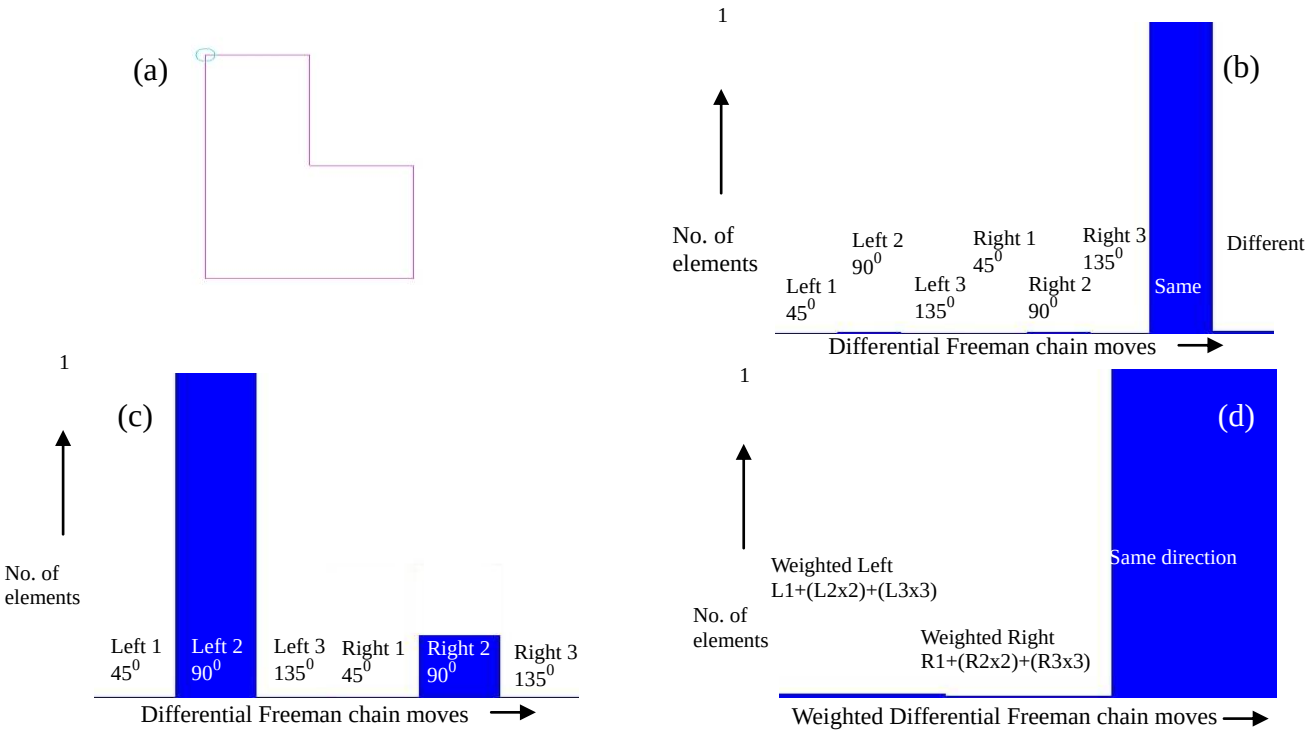


Figure 4. 33: (a) Contour of the 'L' shape object, (b) DFCC (with 'same' and 'different' directions), (c) DFCC (excluded changing in same direction) and (d) DFCC (sums of all left and all right turns in a weighted manner).

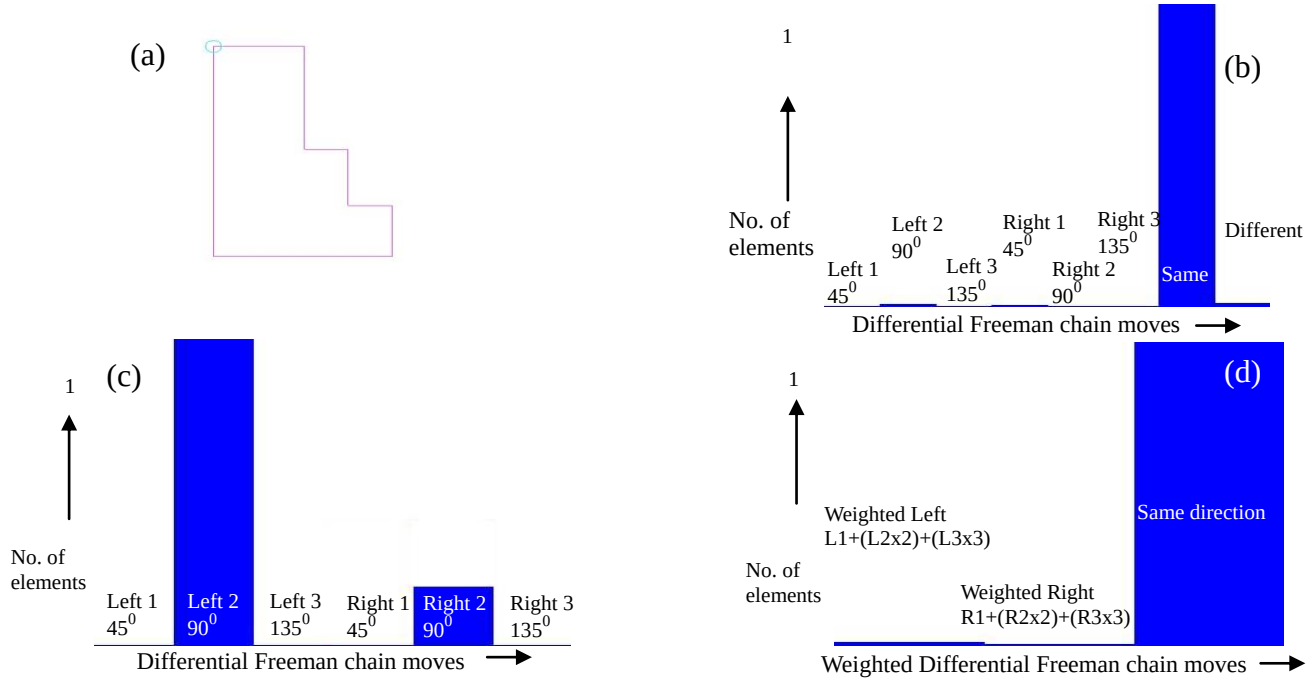


Figure 4. 34: (a) Contour of the 'Stairs' shape object, (b) DFCC (with 'same' and 'different' directions), (c) DFCC (excluded changing in same direction) and (d) DFCC (sums of all left and all right turns in a weighted manner).

The following section describes the numerical analysis of the histogram (i.e. FCC and DFCC) characteristics for the aforementioned sample shapes presented in Figure 4.31. It is clearly visible that each case contains the same values (scaled 0-1) for each of the FCC bins (marked green, see Table 4.4 and Figure 4.31-'d', 'e' and 'f'). FCC only provides the total number of direction changes which are identical for all three cases (1540) (marked yellow, see Table 4.4). But the remaining DFCC characteristics values are different for each shape (see also Figure 4.32-4.34-'b', 'c' and 'd').

Table 4. 4**:** FCC and DFCC characteristic values for Figure 4.30.

Data for Figure 4.30 (a): Rectangle shape object

0.280864 0.000000 0.219136 0.000000 0.280864 0.000000 0.219136 0.000000
0.000000 0.003527 0.000000 0.000000 0.000000 0.000000 1.000000 0.003527
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.007055 0.000000
1.000000
Sum (Total): 1540 Same: 1536.00 Diff: 4.00 L1: 0.00 L2: 4.00 L3: 0.00
R1: 10.00 R2: 0.00 R3: 0.00

Data for Figure 4.30 (b): 'L' shape object

0.280864 0.000000 0.219136 0.000000 0.280864 0.000000 0.219136 0.000000
0.000000 0.004570 0.000000 0.000000 0.001828 0.000000 1.000000 0.006399
0.000000 1.000000 0.000000 0.000000 0.400000 0.000000 0.009141 0.001828
1.000000

Sum (Total): 1540 Same: 1534.00 Diff: 6.00 L1: 0.00 L2: 5.00 L3: 0.00
R1: 0.00 R2: 1.00 R3: 0.00

Data for Figure 4.30 (c): 'Stairs' shape object

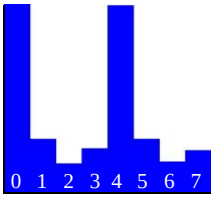
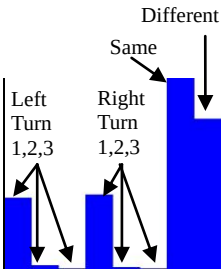
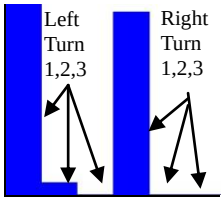
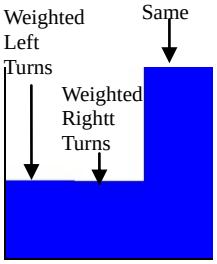
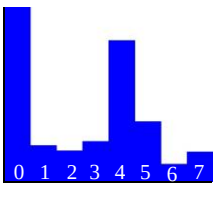
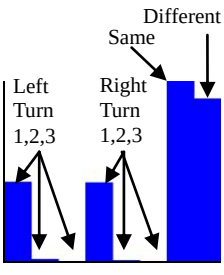
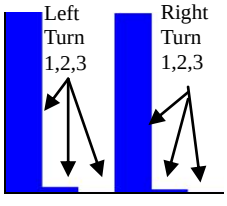
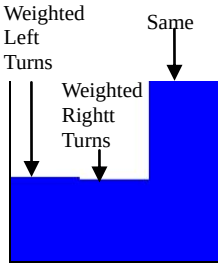
0.280864 0.000000 0.219136 0.000000 0.280864 0.000000 0.219136 0.000000
0.000000 0.005445 0.000000 0.000000 0.003630 0.000000 1.000000 0.009074
0.000000 1.000000 0.000000 0.000000 0.666667 0.000000 0.010889 0.003630
1.000000

Sum (Total): 1540 Same: 1532.00 Diff: 8.00 L1: 0.00 L2: 6.00 L3: 0.00
R1: 0.00 R2: 2.00 R3: 0.00

Combination of DFCC and FCC characteristics are robust than using only FCC to define any shape. More details are reported in chapter 5.

Considering the above description; this work applied a combined approach that includes both FCC and DFCC characteristics of positive and negative samples as an input neuron value of the neural network. Table 4.5 presents the Freeman Chain Code histogram, the differential freeman chain code histogram image including the 'same direction' and 'different direction' values, a histogram avoiding the 'same direction' and 'different direction' values and a histogram that sums all left and all right turns in a weighted manner drawn from the contour input image presented in Figure 4.29.

Table 4. 5**:** Different chain codes histogram for different positive samples.

Input shape contours of positive samples (redrawn Figure 4.29)	FCC Histograms	**DFCC Histograms**		
		Includes 'same direction' and 'different direction'	Excludes 'same direction' and 'different direction'	Sums of all left and all right turns in a weighted manner
	No. of elements, 1, Freeman chain moves	No. of elements, 1, Differential chain moves (1=45^0, 2= 90^0, 3=135^0)		
(a) 				
(b) 				

Each histogram contains unique values for each contour which provides a more unique set of data to the ANN and allows the classification system to be more robustness.

(ii) Convexity Defects

Interpreting human actions is a complex problem for computer vision. Feature extraction approaches using convexity defects are widely used in classification systems [293,294].This algorithmic approach offers a unique method of comprehending the shape of an object or contour by computing a convex hull for the object and then computing its convexity defects [295]. The shapes of many complex objects are well characterised by such defects.

From the blob contour and the blob's convex hull, it is possible to calculate a sequence of contour points between two consecutive convex hull vertices. This sequence forms the convexity defect and it is possible to compute the depth of the i^{th}-convexity defect, d_i . From these depths some useful characteristics for the blob shape can be derived like the depth ratio histogram which is unique to each contour shape.

This software module creates a hull around the segmented blob of interest in which the regions that exist in the hull are recognised. A feature database is created through a dataset of features for multiple individuals lying on the floor. These feature points are normalised and registered for analysis and training of the ANN component. The ANN classifies a contour as a person lying on the floor or another object. Adding this approach with other features: FCC, DFCC (reported in the previous section) and the Hue-Saturation histogram (detailed in the next section) offers a more robust and accurate method to classifying the objects encountered by the robot.

The software that deals with convexity defects returns an image containing the convexity defects, convex hull and extracted top contour (Figure 4.35-a and b). A convexity defect is represented by the start and end points of the defect and by its depth. A scale factor (0-1) is used which determines the ratio of a convexity defect to the maximum dimension of the contour. The defects and plotted into three histogram bins. This work recommends three bins, because more number of bins increases the number of input neurons of the ANN, which leads to high computational cost during the ANN training. On the other hand, three bins are sufficient to contain all the major defects proportions (small, medium and large). Figures 4.35-c and d show the convexity defect histogram of the input contour image.

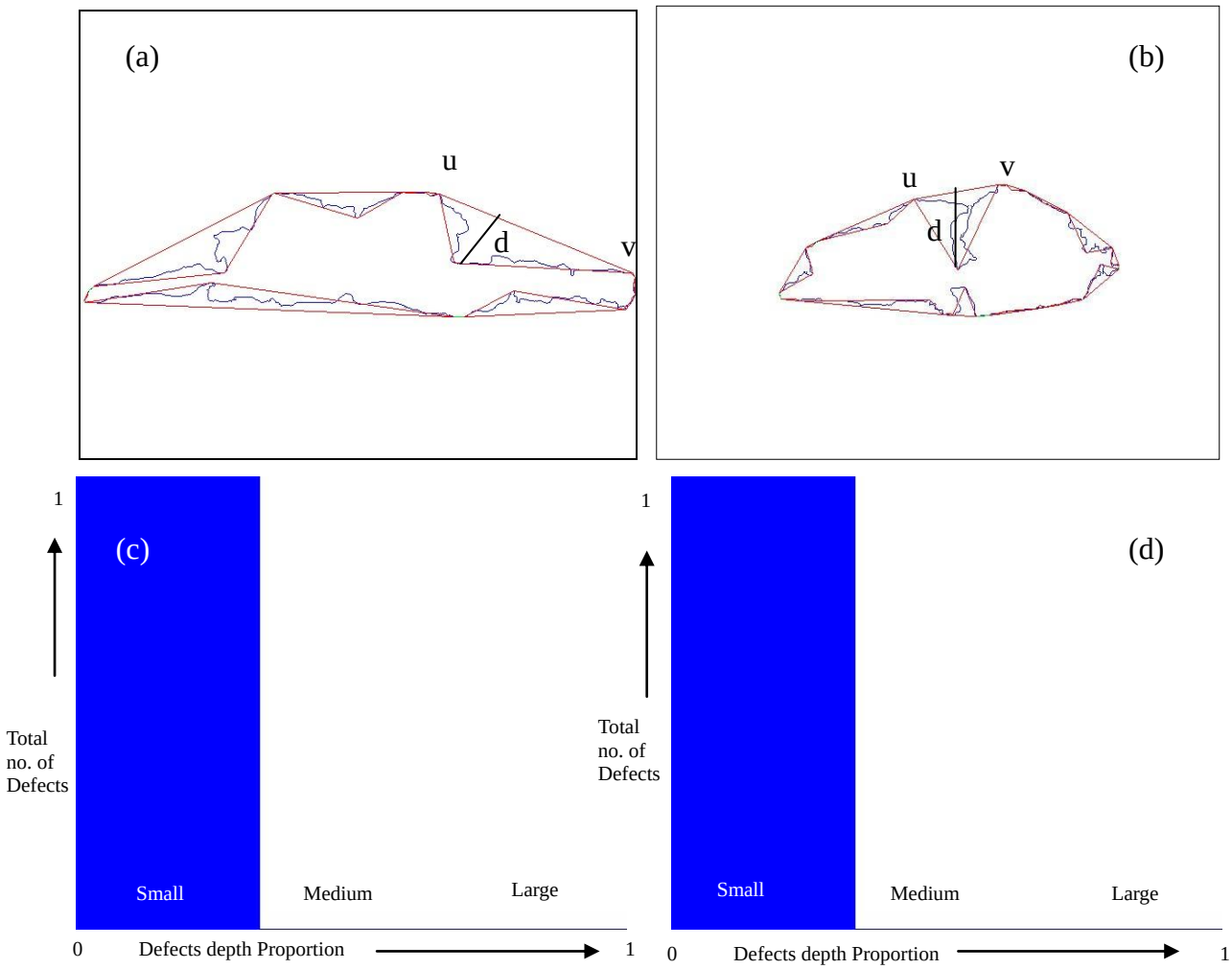


Figure 4. 35: (a) and (b) extracted features for the patient lying on the floor detection process, (c) and (d) shows the convexity defects histograms for the images (a) and (b). In image (a), u and v indicate the start and the end points of the i^{th}-convexity defect; the depth d is the distance from the farthermost point of the convexity defect to the convex hull segment.

(b) Colour based Analysis

Region-based techniques lend themselves more readily to defining a global objective function (for example, Markov random fields [296] or variational formulations [297]).The advantage of having a global objective function is that decisions are made only when information from the whole image is taken into account at the same time. Pixel based analysis is not suitable for complex shape definition due to its low resolution and high computing cost. But Hue-Saturation histogram analysis is another useful approach for object recognition [298,299].The

RGB input needs to be converted into the HSV colour space before obtaining the HS histogram. Figure 4.36 shows the hue and saturation values of the previously segmented colour mask of the full body (see Figure 4.23). It is clearly visible that both histograms are identical, because both input images contain the same colour scheme (i.e. skin and dress) although they are oriented differently (i.e. Figure 4.23).

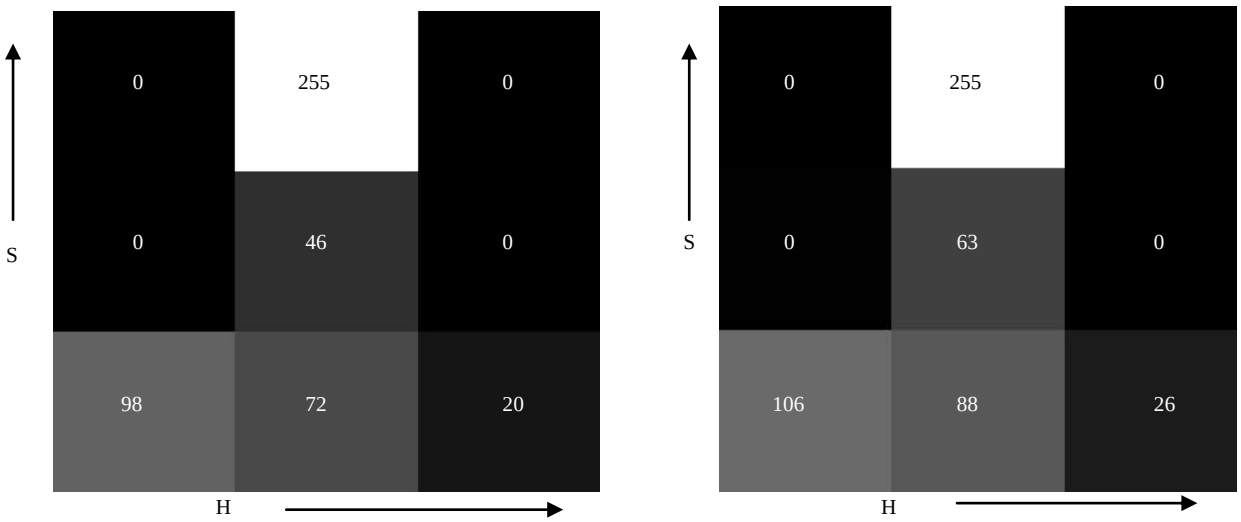


Figure 4. 36: HS Histogram for input image (from Figure 4.23). Both histograms are almost similar due to the same skin and dress colour of the two input images.

This section outlines the histogram extraction procedure. All retrieved descriptors (i.e. FCC, DFCC, Convexity-defects and HS histograms) are written to a data file which is used during the training session and during the real time application of the ANN. The following section describes two training sets for the used input image (i.e. Figure 4.23).

Each training set contains 37 different values (Table 4.6) for the input neuron of the ANN. In Table 4.6, training set '1' is labelled with different colours; the 1st eight values are for FCC histogram (yellow), the following 17 values are for DFCC (green); the following three are the Convexity defect histogram values (red), then next nine are HS (pink) and finally the value 0/1 means the solution of the classification, that is whether the contour is a human lying on the floor. Since the solution (the output of the neural network) is given together with the input values, the training process is called 'supervised'. Each set of parameter values normalised in the range 0-1. More details about ANN are reported later in this chapter.

Table 4. 6: Different ANN descriptor values for training samples (patient lying on the floor).

Training set 1:

0.300597 0.084937 0.045786 0.070338 0.297279 0.084937 0.049104 0.*067021* *0.369464*
0.017483 0.000000 0.388112 0.005828 0.000000 1.000000 0.782051 0.951952 0.045045
0.000000 1.000000 0.015015 0.000000 0.404429 0.399767 1.000000 *1.000000 0.000000*
0.000000 0.199014 0.000132 0.000000 0.147566 0.094683 0.516633 0.041970 0.000000
0.000000 *1*

Training set 2:

0.280864 0.000000 0.219136 0.000000 0.280864 0.000000 0.219136 0.000000 0.441233
0.013487 0.000000 0.437380 0.007707 0.000000 1.000000 0.899807 1.000000 0.030568
0.000000 0.991266 0.017467 0.000000 0.468208 0.452794 1.000000 .700000 0.30000
0.000000 0.196090 0.001367 0.000000 0.163360 0.118120 0.471473 0.049566 0.000022
0.000000 *1*

4.4.6.8 Negative sample Data for ANN

To train any detection system (such as an ANN), negative samples are also essential along with positive samples. Without the description of both types of training samples, such system is unable to make any decision. Therefore the ANN training set includes both positive (patient lying on the floor) and negative (other objects) samples as detailed earlier in section 4.4.6.7. More images of non-target objects are reported in chapter 5. Figure 4.37 presents two raw input images of two different scenarios showing on the left: a person (blue dress) in standing position and on the right: two boxes (blue and yellow colour), where none of the frames contains any patients lying on the floor. This means they are both negative images despite the fact that the one on the left does contain a person, possibly a patient. This section will also show that the developed software is able to detect objects of user defined colours. Figures 4.38 (a) and (b) show the binary segmented mask of the viewed object. Figures 4.39 (c) and (d) show the extracted colour segmented blob of the binary segment object from which an HS histogram is obtained along with fusing the image with the PMD 3D distance matrix. Figures 4.38 (e) and (f) show their respective extracted contour. The FCC histogram is illustrated in Figures 4.39(a) and (b). Figures 4.39(c) and (d) show the DFCC histogram

excluding the 'same direction' and 'different direction' values. The DFCC histogram including sums of all left and all right turns in a weighted manner is shown in Figures 4.39 (e) and (f).

Figure 4. 37: Raw (RGB) input images (Left: Human sample) and (Right: Non-Human sample).

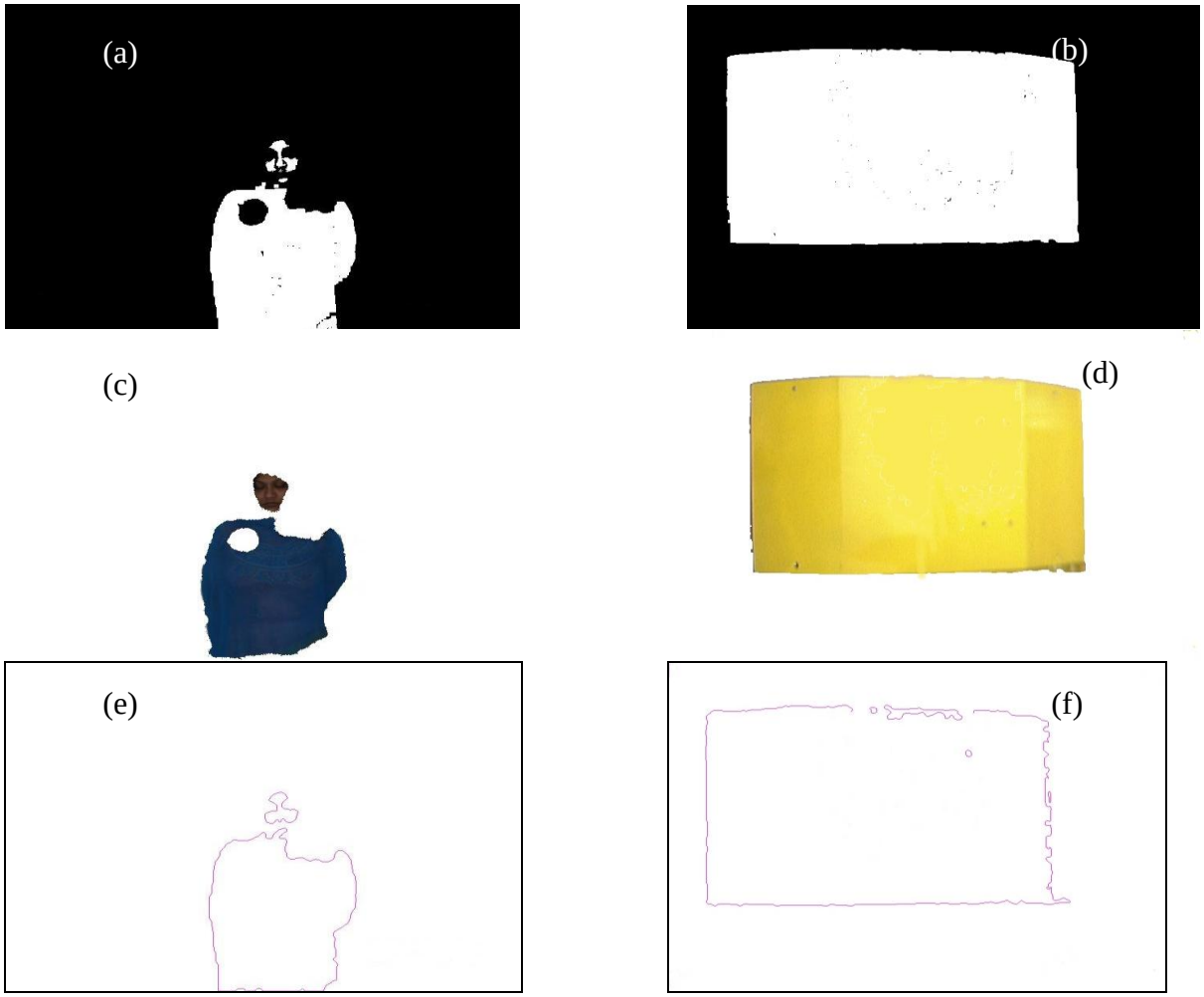


Figure 4. 38: (a) and (b) B&W blob image segment, (c) and (d) Colour blob segment, (e) and (f) the extracted contour sequence of the input images in Figure 4.37.

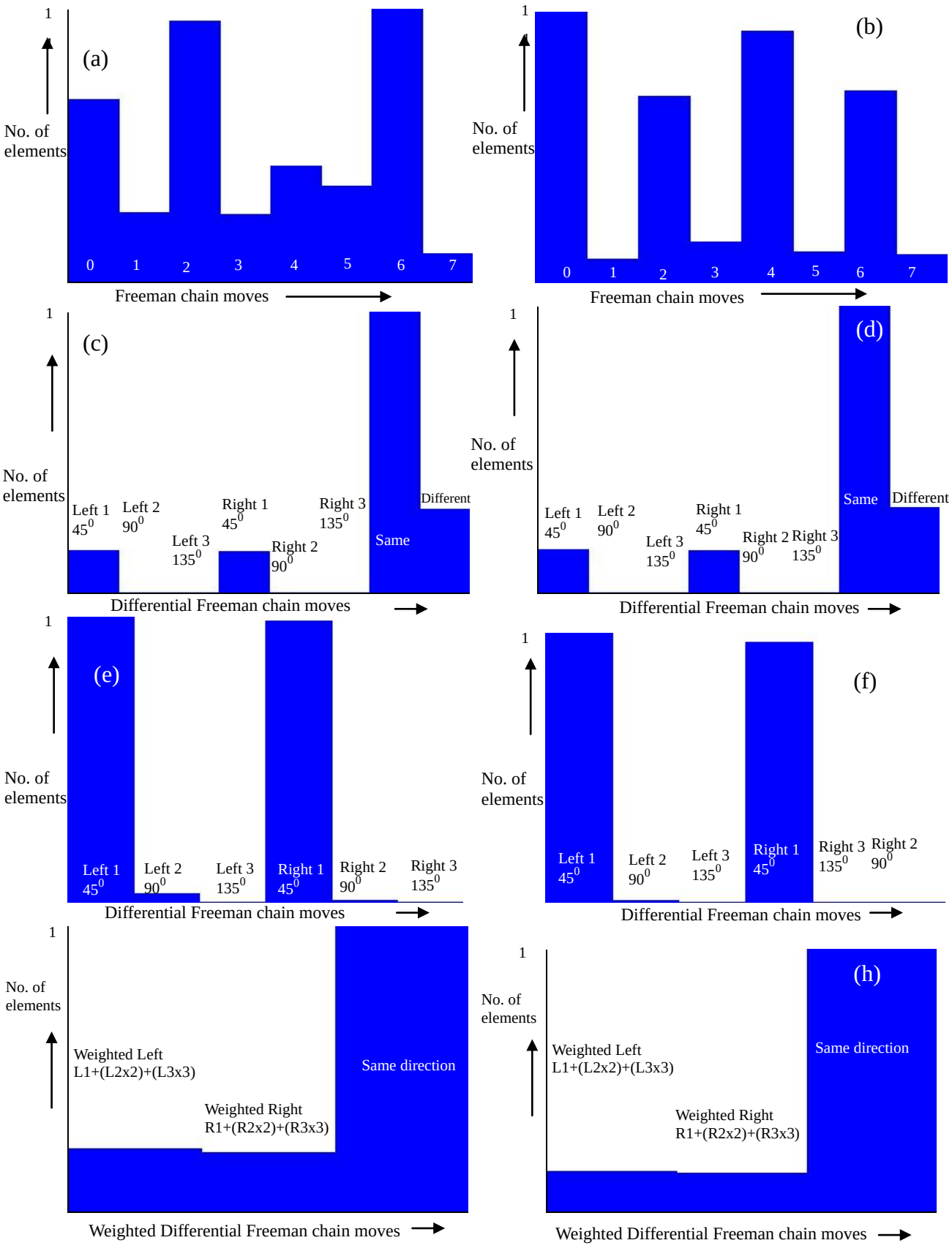


Figure 4. 39: Contour-related (FCC and DFCC) Histograms.

Figures 4.40 (a) and (b) show the convexity defects, convex hull and extracted top contours of the input contour images. Figures 4.40 (c) and (d) show the convexity defect histogram of the input contour images. The hue and saturation values of the obtained colour mask segment (Figure 4.38-c and d) generates the HS histogram presented in Figure 4.41.

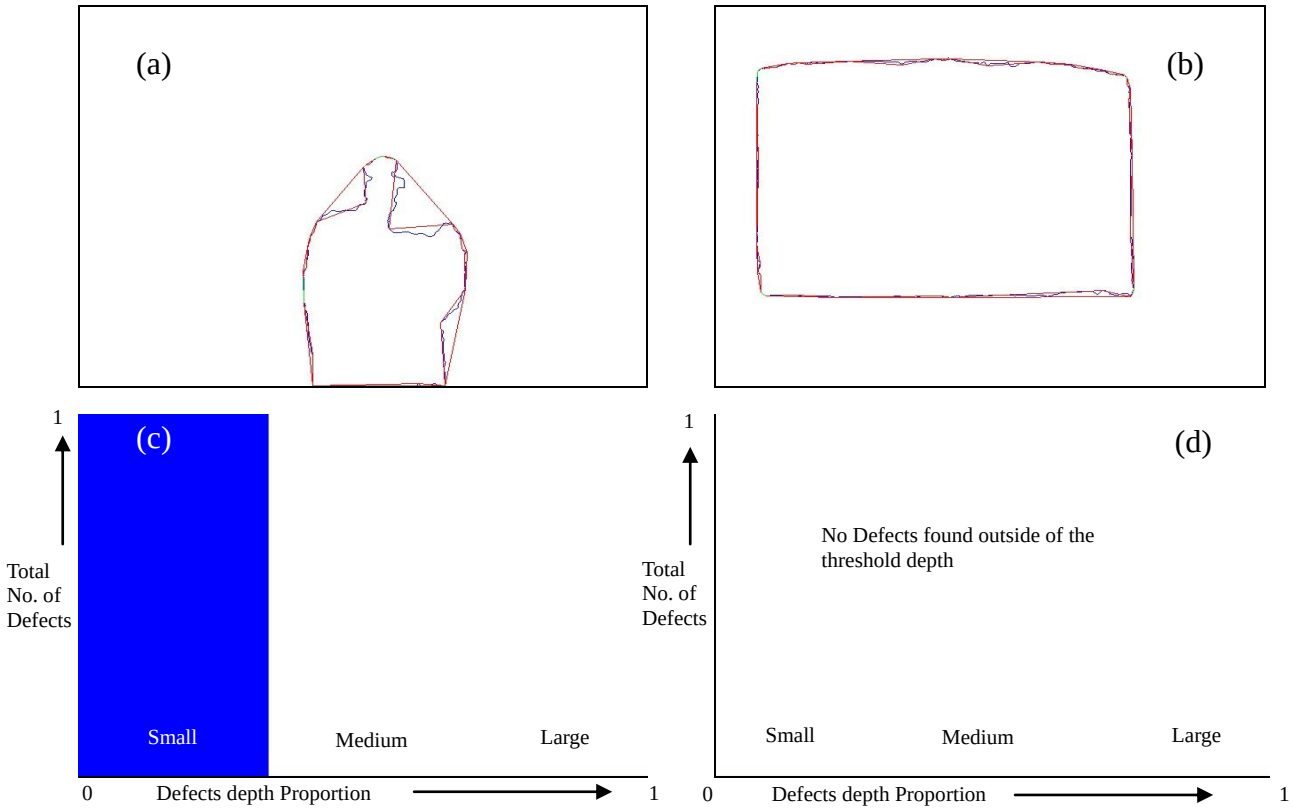


Figure 4. 40: (a) and (b) Convexity defects of the input image blob contours (Figure 4.38), (c) and (d) convexity defects histogram of the respective images (a) and (b).

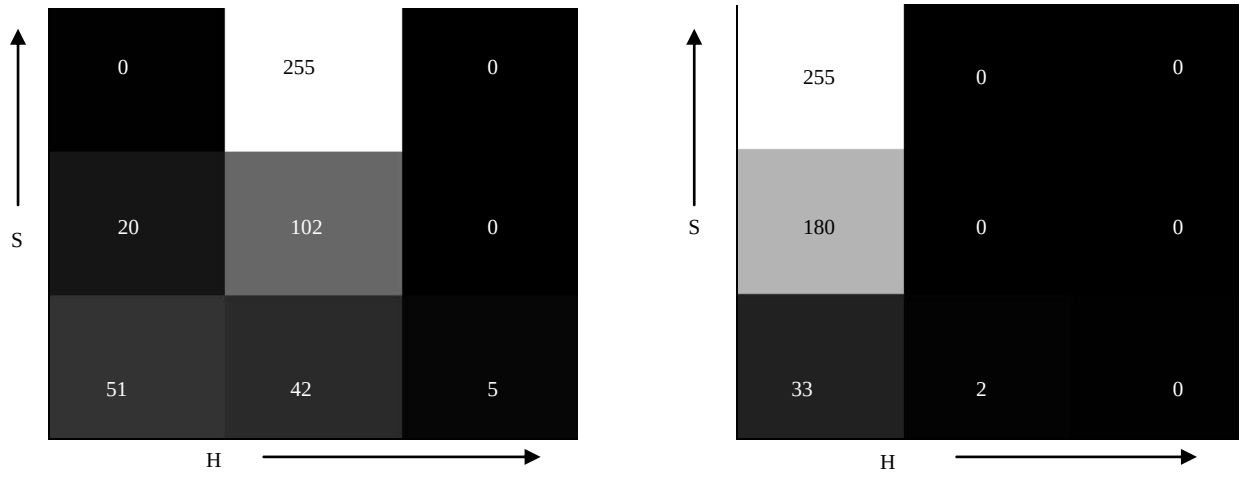


Figure 4. 41: The hue and saturation histogram of the obtained colour mask segment.

Table 4.7 presents the negative ANN training Data samples for the above examples where the last value '0' means the target object is not a patient lying on the floor.

Table 4. 7: Different ANN descriptor values for target example (no patient lying on the floor) samples.

Training set 3:
0.166667 0.063063 0.238739 0.061937 0.105856 0.087838 0.250000 0.025901 0.208267 0.006359 0.000000 0.205087 0.001590 0.000000 1.000000 0.421304 1.000000 0.030534 0.000000 0.984733 0.007634 0.000000 0.220986 0.208267 1.000000 1.000000 0.000000 0.000000 0.107447 0.043570 0.001916 0.088470 0.214575 0.532152 0.011871 0.000000 0.000000 0
Training set 4:
0.264993 0.023013 0.182008 0.039749 0.245467 0.029986 0.187587 0.027197 0.150858 0.000903 0.000000 0.145438 0.000000 0.000000 1.000000 0.297200 1.000000 0.005988 0.000000 0.964072 0.000000 0.000000 0.152665 0.145438 1.000000 0.000000 0.000000 0.000000 0.070206 0.382139 0.540134 0.006308 0.000000 0.000000 0.001212 0.000000 0.000000 0

4.4.6.9 Artificial Neural Network (ANN)

The shape of the lying patient can vary in size, shape and orientation relative to the camera. Hence this work has implemented a multi layered algorithm that can take into account these variations. This multi-layered ANN is fed with the FCC, DFCC, convexity defects and HS histograms.

A neural network performs well under non-linear conditions [300]. This research [300] has proved that the multi-layered neural network can accurately approximate any linear or non-linear function. Even by learning an incomplete dataset, the neural network can achieve an accurate prediction of the entire data with its generalisation capability. It is, therefore, possible to utilise the neural network to detect any person lying on the floor.

The system uses a feed-forward neural network trained with the back propagation algorithm in order to classify objects. The inputs of this ANN component are the features computed by the previous components (i.e. FCC, DFCC, Convexity defects and HS). A general representative diagram of neural network neurons can be seen in Figure 4.42. The neural network is composed of three layers: (i) the input layer with input neurons (i.e. FCC, DFCC, Convexity defects and HS histogram value) (ii) the hidden layer with hidden neurons, and (iii) the output layer with one neuron, where the likelihood of being a human lying on the floor or not is stored. All weight values between neurons are scaled between 0 and 1.

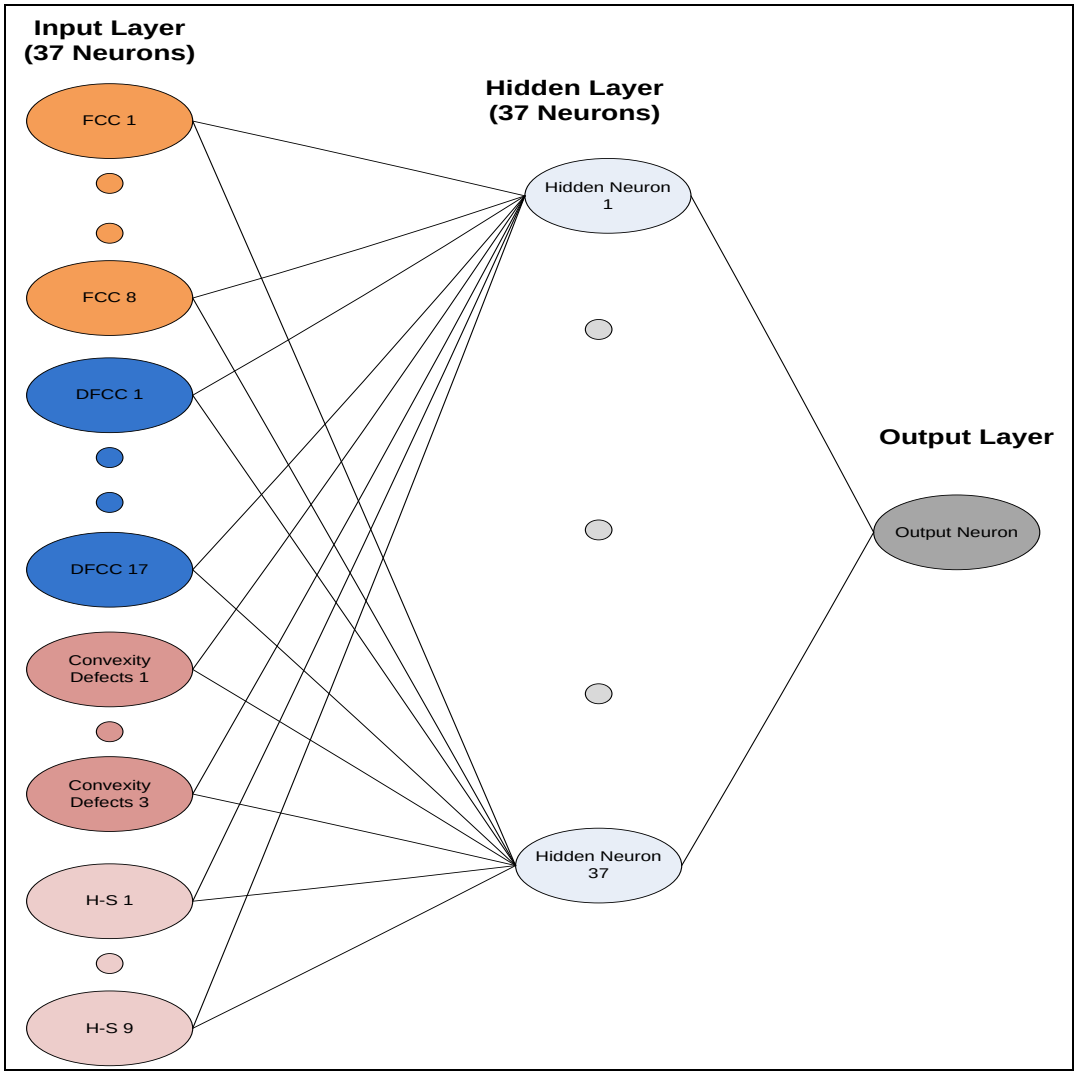


Figure 4. 42: General diagram of the Neural Network model. The system uses a feed forward neural network with three layers. The input layer is composed by one set of input neurons for each used descriptor (FCC, DFCC etc.).

The summary of this section is that ANN uses descriptors (FCC, DFCC, Convexity defects and HS) as the feature space for building a classifier. It uses the fact that the shape of an object can be well represented by a distribution of local intensity gradients or edge directions. This study experimented with several orientation and spatial binning resolutions and normalization schemes to obtain the maximum performance (more detail of these experiments is reported in chapter 5). For classification, a positive dataset containing patients lying on the floor and a negative dataset containing other persons and non-human objects were used, and a linear classifier is trained using ANN on the gradient histogram features from the two classes. This classifier can then be applied to a new input image (that did not take part in the training process) for detecting humans/patients lying on the floor.

The ANN needs to be trained before the real time application. During the training session, the ANN calls a function to get descriptor parameters from a text file called PARAMETERS.TXT (see Appendix –A) generated before the training session of the ANN for each training sample which detailed earlier. This makes the process flexible by avoiding re-compiling the software when parameters are altered. This work used 2000 positive and 1000 negative samples during the training session. More training samples will make the system more robust by minimising the number of false alarms, although it will result in high computation cost of the training session. Each of the samples contains 37 input neuron values and one output value. In real time application the input parameter values are fed to the ANN to get the decision from it in the form of either 0/1 (i.e.'1' for a patient lying on the floor and '0' for any other solution). Based on this decision the Situation reorganisation module (i.e. patient detection on the floor scenario) stores the input image frame for further access and raises an alarm message to the swarm system. This study tested the ANN component with another 500 test samples (including positive and negative samples which didn't take part in the training session) and found 94% accuracy of the system (ANN) detection rate.

4.5 Patient condition monitoring module

This module is responsible to monitor patient psychical condition (i.e. vital signs: Body temperature, ECG, Heart rate, Respiration rate and so on) remotely using the Equvital [51] sensor belt (sensor details reported in chapter 3).

4.5.1 Driver software

The Equvital sensor is shipped with a software which is compatible only with the Windows operating system. Hence for this study it was necessary to develop a new software driver which is able to communicate with the Equvital System in the Linux environment through a serial port and also via a Bluetooth connection. The following section outlines a more detailed overview related to this software development.

In the interface specification document [51], it is stated that the sensor information should contain a letter followed by two characters as shown in Table 4.8. Each letter refers to a different output, for example ‘e’ (hex-65) refers to the output reading from ECG 1 and ‘c’ (hex-63) refers to the output of ECG 2, etc.

Table 4. 8: Interface application Layer

Message Type	**Data 1**	**Data 1**
65	46	28

In this study, the Equvital sensor data (initially received as a Hexadecimal format) is converted according to the ASCII specification (American Standard Code for Information Interchange) as reported in Table 4.9. After receiving the hexadecimal value from the sensor through the commutation port (either RS232/Bluetooth) the software converts these data to the ASCII format so that different parameter readings can be created for each output. This software is written in C++. More details of conversion methods are reported later in this chapter.

4.5.2 Module algorithm

The patient physical condition monitoring software receives module activation commands through the Orca interfaces (described in section 4.2). To get the location ID (where the sensors’ data to be measured), the module software also uses the Orca interface. Then the measured sensor data is compared with pre-defined values for the normal physical condition

and are stored for future reference. In case of any event (i.e. high/low body temperature, heart rate and respiration rate), the module raises an event alarm which is shared with other SWARM components by publishing it through the Orca interface. Figure 4.43 shows a flow diagram of this module's software algorithm.

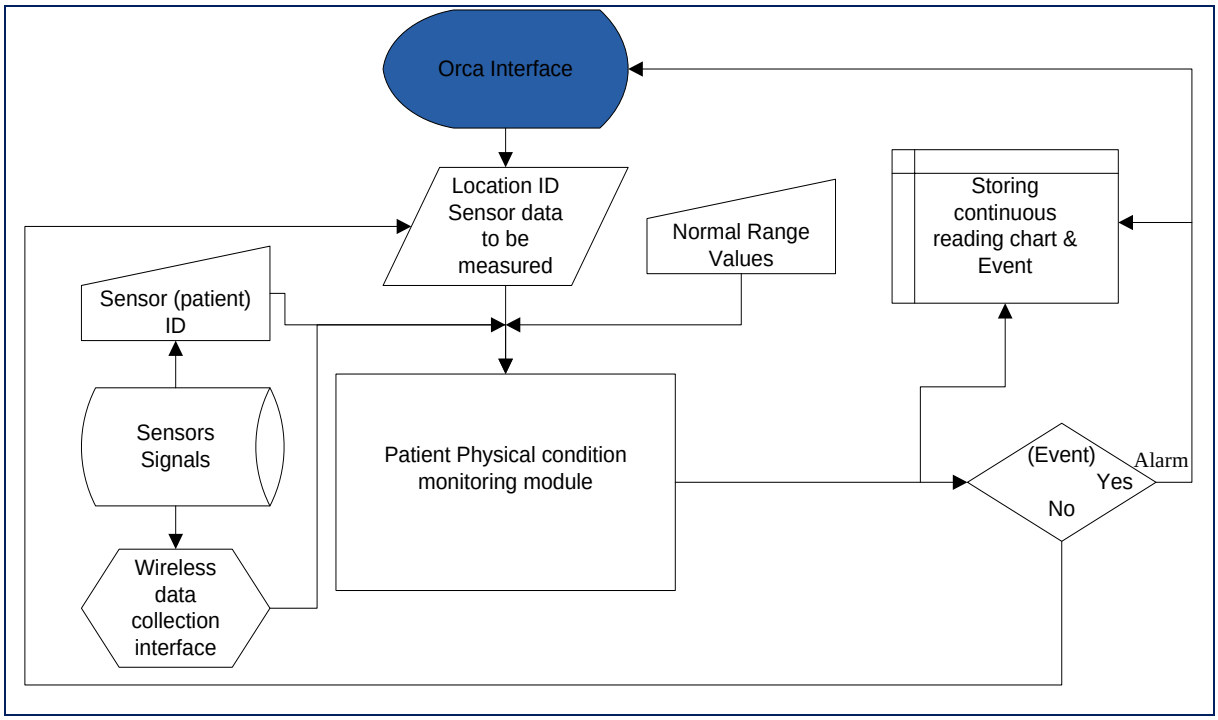


Figure 4. 43: Flow chart of the patient condition monitoring module.

The Patient Condition Monitoring software steps are as follows:

1. Gets location/patient ID where measurements to be taken.

2. Reads normal and previous physical condition values for this ID/patient from the data file.

3. Reads Bluetooth/serial port channel 'Patient Data', converts it according to the technical specification.

4. Calculates body temperature, ECG curve, heart rate, respiration and so on.

5. Compares actual and normal sensor values, raises event if measurements are outside range.

6. Compares actual and previous sensor values (if any), raises event of notification if measurements are increased or decreased.

7. If at any stage the equipment stops working, then the status would be Module_Active = NOT_WORKING and an event is raised. All operation is stopped and only 'shutdown' or 'reset' commands are accepted from other components.

4.5.3 Module software

Present study initially tested the sensor device for experimental purposes and familiarise with the data format and communication. Figure 4.44 and Figure 4.45 represent two snap shots of using readings from the Equvital sensor belt. Detailed analysis can be found below and any supplementary information can be found online [51].

Figure 4.43 demonstrates the key areas of concern in relation to the vital sign display system (VSDS Viewer):

1. Sensor Identity – For each patient monitored a unique identity number is used to distinguish the information between them. The identity number on this purchased device is 0607007; when the communication is established, this information is automatically assigned to a particular patient (it can be used as patient ID). This is a big advantage as the same software can allow for different patients to be monitored at any.
2. Heart Rate –The heart rate is a wave of blood created by contraction of the left ventricle of the heart and is measured in beats per minute (bpm). The normal heart rate for an adult is between 60 – 100 bpm so the average heart rate is approximately 80 bpm, A heart rate which is excessively fast is known as tachycardia and this would be greater than 100 bpm and one less than 60 bpm is referred to as bradycardia. This may also vary depending on age, sex, exercise, fever, medications, haemorrhage, sleep, stress and even position changes [49,301].
3. Respiration Rate – Respiration is referred to as the act of breathing. Inspiration is the air taken into the lungs and expiration is the breathing of air from the lungs. The normal breath rates for an adult are 12 – 20 breaths per minute so the average respiratory rate is approximately 16 breaths per minute [49].
4. Skin Temperature (measured on the chest) – This temperature reading gives the surface body temperature for the patient. In this example the patient has just worn the device so the reading had not reached a steady value. The skin temperature is measured using thermistors to detect a reading and this may take a while before an accurate reading is achieved. For simplicity, the normal range for body temperature according to [301] can be taken between 36°C to 38°C.

5. PWI Indicator (Physiological Welfare Indicator) – This indicates whether the readings for all the physiological parameters are either above or below what's considered normal. Different colours indicate different levels of risk associated with the subject.
6. Sensor Status – This indicates if the battery is low or OK. It does not give different levels for the power of the battery. The sensor status also indicates whether the lead connections, i.e., the ECG sensors are positioned correctly, and whether the belt is connected properly.
7. Ambulatory Status – The ambulatory status indicates whether the subject is moving or stationary. It also indicates if the person is prone, upright, lying, inverted or lying on their side.
8. Summary Update – This updates every 15 seconds and the data is recorded into excel files that can be read by other software components for viewing later.

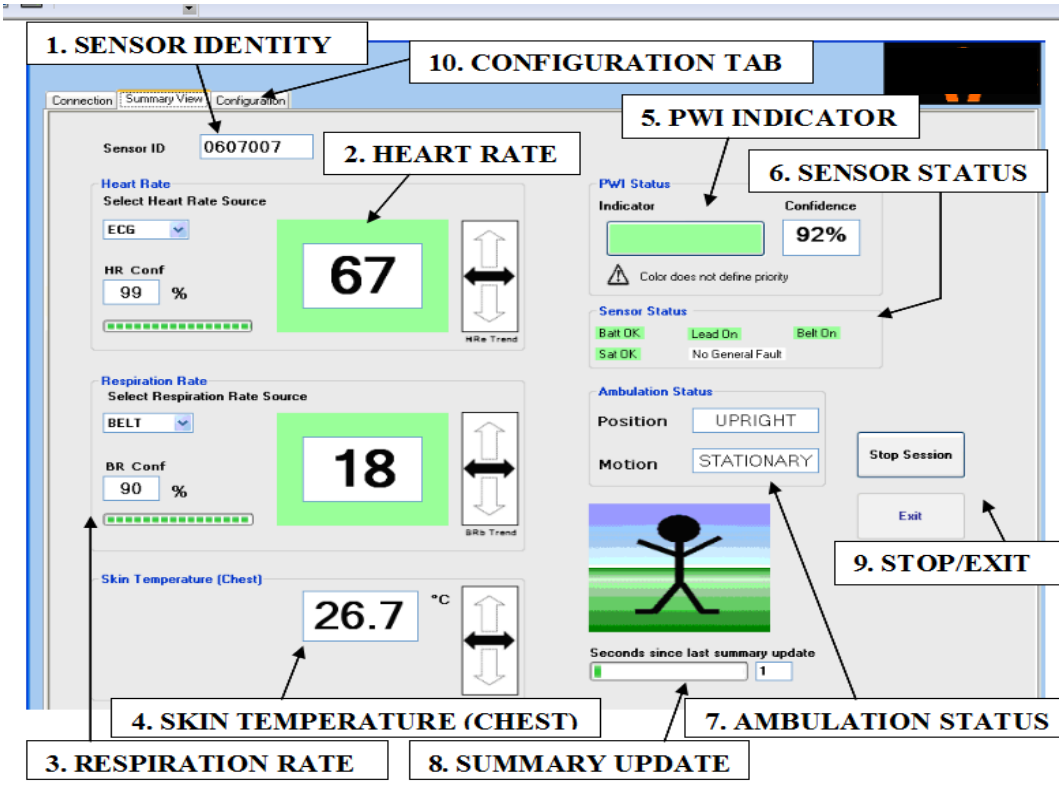


Figure 4. 44: VSDS Viewer

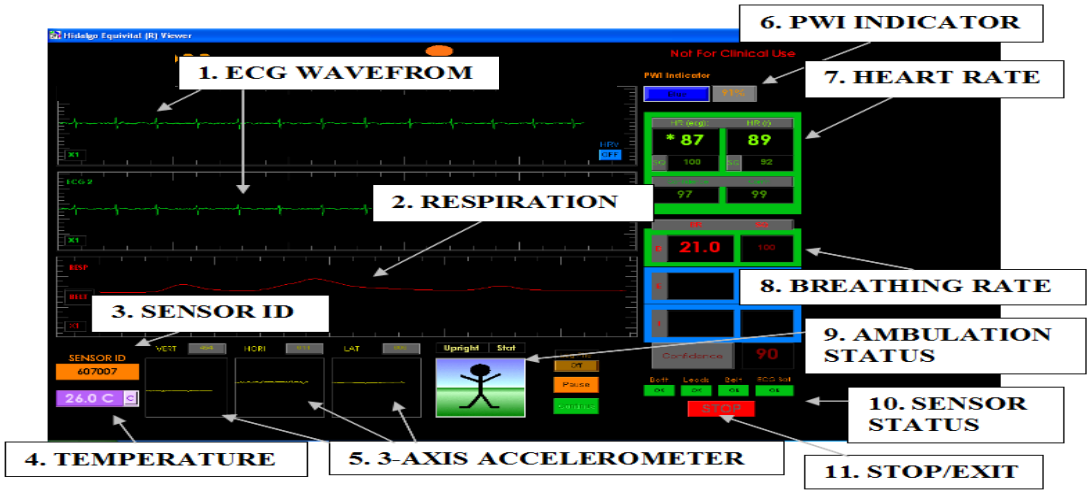


Figure 4. 45: Sensor electronic module Viewer

4.5.3.1 Equvital Sensor Data conversion

When the module is activated then the connection between the sensor and the computing device (Gumstix) is established and the sensor data is transmitted in a pattern presented in Table 4.9.

Table 4. 9: Sample hexadecimal data converted into ASCII.

0C28	**6C4B**	**2976**	**292A**	**7759**	**2665**	**4628**	**6344**	**3865**	**4538**	**6341**
(np) (	l k	) v	) *	w Y	& e	F (	c D	8 e	E 8	c A
3865	**3D48**	**633E**	**3869**	**2428**	**653C**	**5863**	**3E38**	**6620**	**2065**	**3F28**
8 e	= H	c >	8 i	$ (	e <	X c	> 8	F (sp)	(sp) e	? (
6342	**4865**	**4638**	**6345**	**4865**	**4548**	**6341**	**4865**	**3E58**	**633E**	**4865**
c B	H e	F 8	c E	H e	E H	c A	H e	> X	c >	H e
3C28	**633E**	**5865**	**3E38**	**6341**	**586C**	**4F29**	**7626**	**2A77**	**5826**	**6545**
< (	c >	X e	> 8	c A	X l	O)	v &	* w	X &	e E
4863	**4458**	**6545**	**5863**	**4258**	**653F**	**2863**	**3E28**	**6923**	**2865**	**3C38**
H c	D X	e E	X c	B X	e ?	(c	> (	i #	(e	< 8
633E	**2866**	**2020**	**653C**	**4863**	**4028**	**6544**	**5863**	**4328**	**6547**	**2863**
c >	(f	(sp) (sp)	e <	H c	@ (	e D	X c	C (	e G	(c
4338	**6541**	**3863**	**3F38**	**653C**	**4863**	**3E38**	**6E25**	**206E**	**2121**	**6E34**
C 8	e A	8 c	? 8	e <	H c	> 8	n %	(sp) n	! !	n 4
206E	**3020**	**6E38**	**2071**	**2040**	**6F4C**	**2065**	**3D58**	**6340**	**386C**	**5229**
(sp) n	0 (sp)	n 8	(sp) q	(sp) @	o L	(sp) e	= X	c @	8 l	R)
7622	**2A77**	**5C26**	**7B3F**	**2C7D**	**3422**	**7459**	**2261**	**4421**	**7944**	**216B**
v “	* w	\ &	{ ?	, }	4 “	t Y	“ a	D !	y D	! k
4421	**6E45**	**207E**	**4421**	**6820**	**2062**	**2020**	**6044**	**216E**	**2720**	**6E5D**
D !	n E	(sp) ~	D !	h (sp)	(sp) b	(sp) (sp)	‘ D	! n	‘ (sp)	n]
206E	**4020**	**7520**	**206E**	**4120**	**6744**	**2170**	**3A21**	**6E3D**	**207C**	**3C21**

Where:

(np) = new page

(sp) = space

The developed software reads and deciphers the sensor data from the data stream.

Most of the measured sensor values are represented as single values for each measurement. These include body temperature, heart rate, respiration rate, body acceleration, etc. However, an electrocardiogram (ECG) is a waveform that is represented by a sequence of data points. Therefore, it takes a number of measurement cycles to obtain one waveform of ECG. This electrical activity generated by the heart can be displayed graphically as shown in Figure 4.46. If a person is taking rest, each heartbeat takes approximately 0.85 seconds [302].An R-wave interval is represented by the difference between the R peaks per minute.

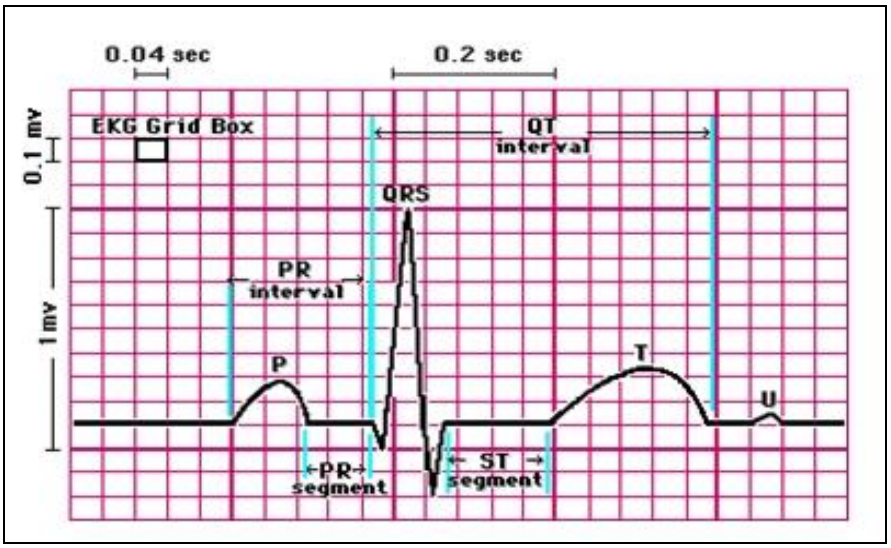


Figure 4. 46: A normal ECG waveform [303]

In order to distinguish between the different messages types, as all the data is sent together, it is necessary to isolate and create an array. This allows the different letters, which represent different information to be cluster together so that a different reading can be created for each output. The results of this module are reported in the next chapter.

Chapter 5

TEST RESULTS AND ANALYSIS

5.1 Introduction

During the existence of the iWARD project, the robot system (as a whole, and its components separately) was tested several times. These include the Annual Review Meetings and the final evaluation tests that were performed in hospitals.

After the iWARD project finished, research on the service modules continued, and the improved modules were tested in a laboratory environment. Since the robot bases remained with their developers, for these final module tests a client software was developed that simulated the actions of the robot base (hardware management, event registration and processing, …).

5.2 iWARD review meeting tests

During the three-year existence of the iWARD project (January 2007 – March 2010), three Annual Project Review Meetings were organised at partners' locations (in Newcastle and San Sebastian). Apart from representatives from all project partners, two EU project reviewers and a project officer took part in these meetings. Their task was to assess and critically evaluate the robot system at each stage of its development. The final review meeting (February 2010, San Sebastian) concluded with the Evaluation Report of the project reviewers, which acknowledged the project as a success. During this final meeting all aspects of the robot system (including the functionality of the service modules) were demonstrated and tested. Figure 5.1 shows the testing of the iWARD robots during the Final Project Review meeting (San Sebastian).

Figure 5. 1: Testing of the iWARD robots during the Final review meeting (February 2010, San Sebastian).

5.3 iWARD hospital tests

The test of the robots during the review meetings were mainly concerned with the technical aspects of the developments: hardware and software functionality. In order to understand the usability of the system by healthcare professionals, two additional evaluation tests were organised in a teaching hospital in Newcastle (October 2009) [240] and in the Matia hospital in San Sebastian (November 2009). Some thirty healthcare professionals (nurses, nursing lecturers and healthcare assistances) attended the two events. Each participant evaluated each scenario of the robot system, including the ones that are the objective of this research thesis. The general consensus of the evaluation was that a robot system consisting of small, self-navigating robots with interchangeable service modules is a good concept for making the

work of healthcare professionals more efficient. Figure 5.2 shows the testing of the iWARD robots during the first and second Evaluation meetings.

Figure 5. 2: Testing of the iWARD robots during the first and second Evaluation meetings (October 2009, Newcastle, and November 2009, San Sebastian).

5.4 Post-iWARD module tests

5.4.1 Environmental Condition Monitoring Module

When activated, the Environmental Condition monitoring module loads the data file 'Environment_Normal_Values.txt' (see Table 5.1) which contains the limits of the normal environmental conditions for each location in the hospital. After a measurement is taken in a room, the module reports and updates the module status to a log file (detailed in Table 5.2). If an event occurs the system also raises an alarm through the Orca component to notify the Swarm system. Figure 5.3 shows the sensor data readings of this module. Figures 5.4 and 5.5

represent sample graphs of temperature and humidity sensor readings with their respective time stamps.

Table 5. 1: Data file 'Environment_Normal_Values.txt'.

Normal Data structures:
Default_Room Temp_Min Temp_Max Hum_Min Hum_Max
Predefined Data:
Room1 Temp_Min 20.2 Temp_Max 28.1 Hum_Min 35.6 Hum_Max 40.9
Room4 Temp_Min 18.3 Temp_Max 20.5 Hum_Min 70.45 Hum_Max 93.6
Pharmacy Temp_Min 19 Temp_Max 23.1 Hum_Min 74.1 Hum_Max 91.5
Corridoor1 Temp_Min 16.4 Temp_Max 19 Hum_Min 65 Hum_Max 70.8

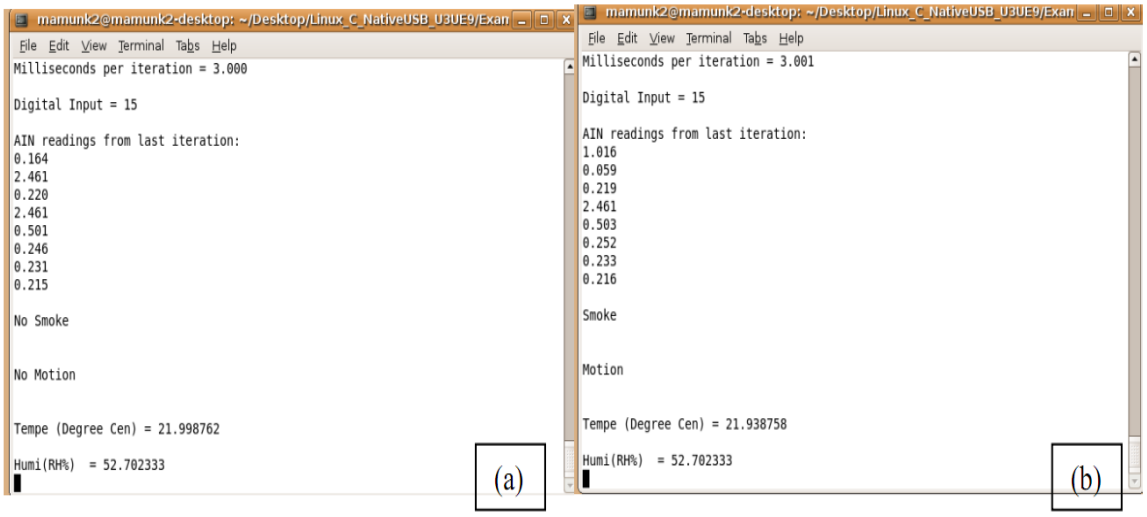


Figure 5. 3: GUI screen shot of the Environmental monitoring module testing; (a) Results show temperature readings with ‘no detection’ of smoke (b) temperature readings with detection of smoke (smoke created under a controlled environment).

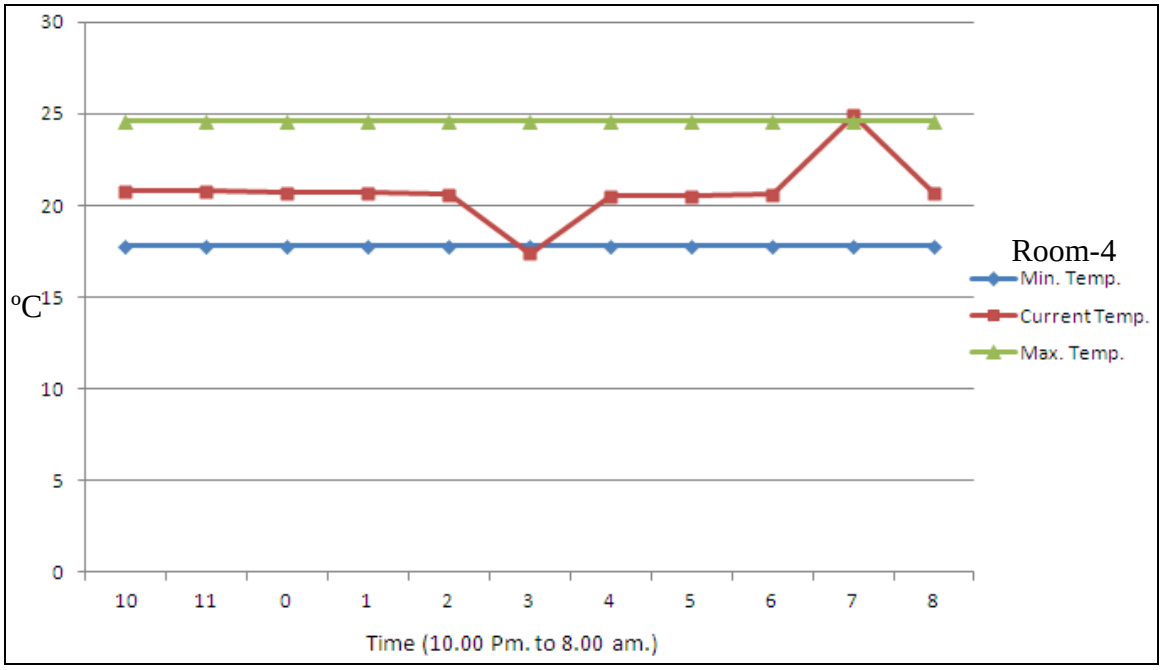


Figure 5. 4: Sample graph of temperature sensor readings with their respective time stamps.

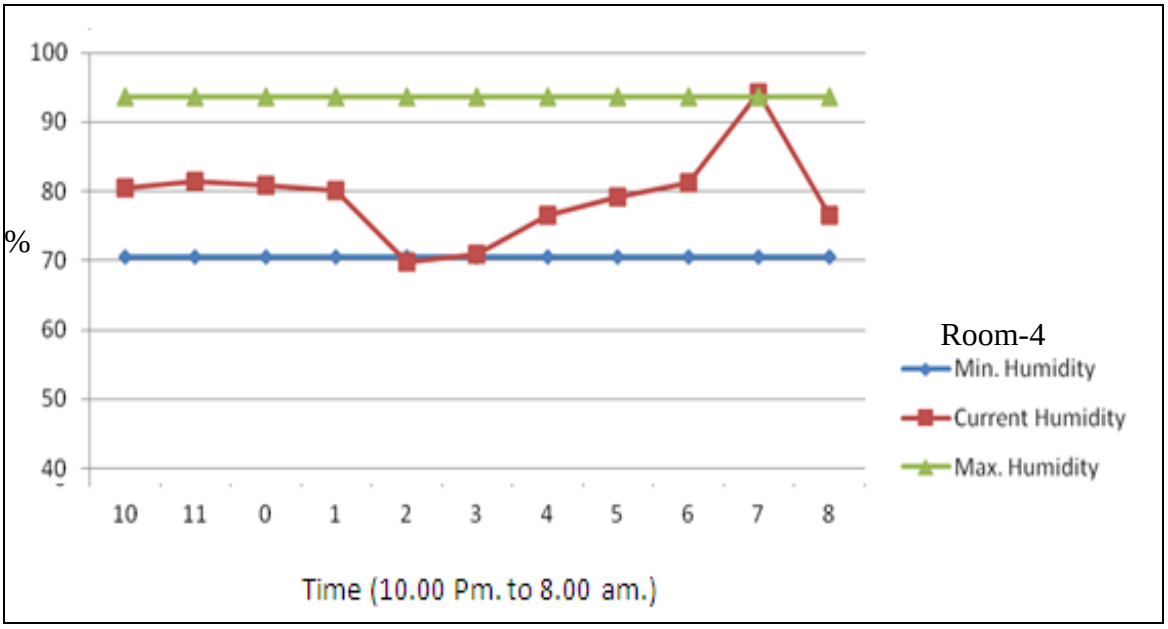


Figure 5. 5: Sample graph of humidity sensor readings with their respective time stamps.

Table 5.2 presents a log file of the Environmental condition monitoring module which is continuously generated during the operation and reports the status and activities of the module.

Table 5. 2: Example Log File.

Serial Port Name: /dev/ttyS1

Environment Monitoring Log File: EnvMon_Logfile.txt

Logfile Max Length: 40000.000000 characters

Environment Monitoring Data File: Environment_Normal_Values.txt

Operation Mode: 0 (0:LIMITED; 1:FULL_ORCA)

Print Mode: 1 (1:print to display; 0:do NOT print)

Error Log Mode: 1 (1:log errors to local file; 0:do NOT log)

Processes Log Mode: 1 (1:log processes to local file; 0:do NOT log)

Event Log Mode: 1 (1:log events to local file; 0:do NOT log)

Max number of cycles to get response from DAQ: 500000.000000

Number of repeated measurements: 3

Time between measurements: 5 sec

"Serial port is open" Time: Sun Jun 7 13:31:02 2009

"Equipment is connected" Time: Sun Jun 7 13:31:02 2009

EVENT:

HIGH TEMPERATURE in Room4

LOW HUMIDITY in Room4

Room4: N.Temp:18.3C-20.5C (22.6)

N.Hum:70.4%-93.6% (36.1)

Smoke:0

Time: Sun Jun 7 13:31:34 2009

"Serial port is Closed" Time: Sun Jun 7 13:31:34 2009

"Equipment is disconnected" Time: Sun Jun 7 13:31:34 2009

5.4.2 Situation recognition module

5.4.2.1 Intruder monitoring

This work used a graphical user interface (GUI) to test and demonstrate the test results of the intruder monitoring scenario. Figures 5.6 and 5.7 show the system's GUI overview of the intruder detection system.

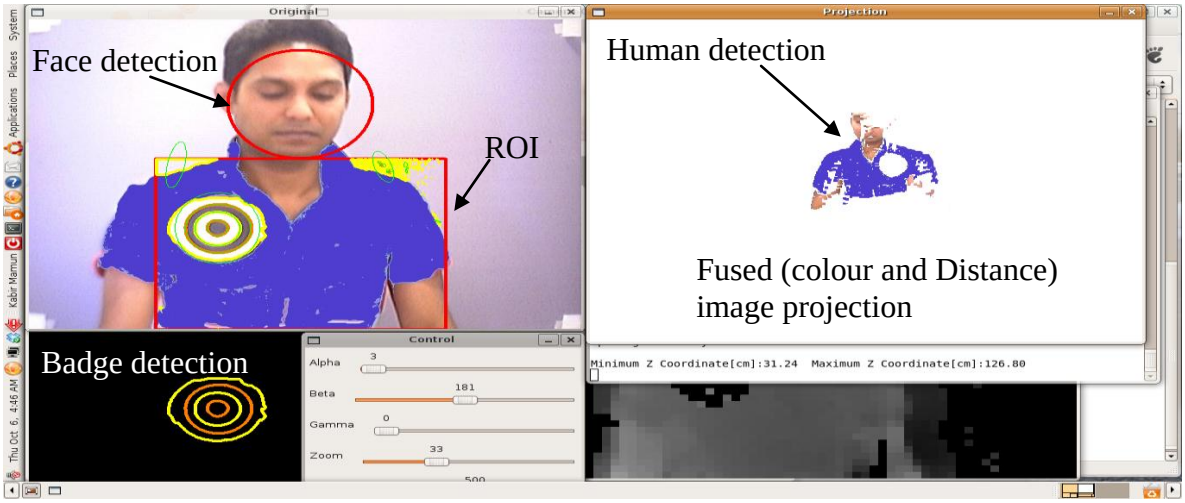


Figure 5. 6: GUI of the Intruder detection scenario.

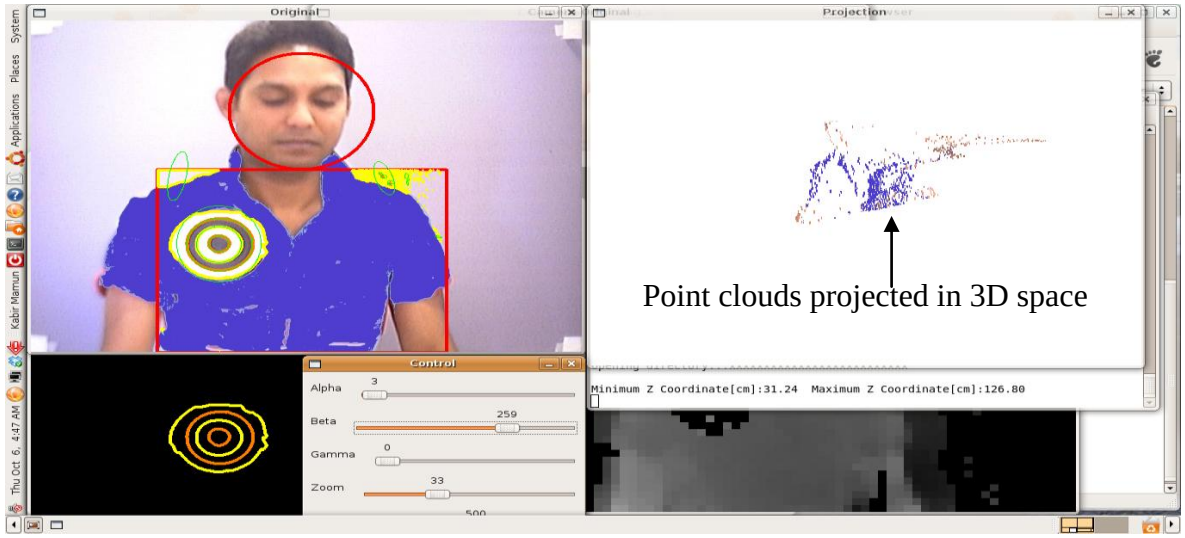


Figure 5. 7: GUI of the Intruder detection scenario shows point clouds of the target object projected in 3D space.

The intruder detection component of the situation recognition module is a fully automated,

unsupervised system which has been tested under varying conditions (duration of operation, lighting conditions, multiple human faces, distances and viewpoint angles). The following section reports more testing results.

A set of test samples containing faces and badges were selected to verify the detection distance capabilities of the system. An object's area in the image varies depending on the camera position and settings (camera height relative to ground, object distance relative to the camera, zoom factor and so on). The camera height from the ground was set to 1.1 meters based on the usual position of the module box on the iWARD robots. Studies show that the average height of a human being varies depending on ethnicities. 1.7 meters are the average height of the people involved in this work. Based on the maximum viewing angle of the Axis camera, in order to get the face and badge area visible in the image frame, the minimum distance of the human relative to the camera must be 1.5 meters. The following results show distance intervals from 1.0 to 7 meters.

The present study observed that maintaining the camera height and field of view (Camera FOV, 36º-H and 27º-V) within a 1.0 meter range is not adequate to fully view the face of the average person (in Figure 5.8 the face detection algorithm cannot work). The minimum viewing distance can significantly be reduced (down to about 0.5 meters) when using the tilt mechanism of the camera. In this case, the main limitation is that both the face and the badge need to be within the frame. Figure 5.9 shows a false detection of a face using the haar classifier [254]. At a distance of 1.5 meters the face is fully visible (Figure 5.8-a) and the face and both of the badge detectors (contour and classifier based) perform successfully (see Figure 5.10). The haar face detection classifier, however, at times generates false positives resulting in multiple ROIs being labelled (see Figure 5.10-b). On the other hand, the developed classifier and contour analysis algorithm are able to detect the badge within the right ROI (see Figure 5.10-c and d).

Figure 5. 8: Target frame containing a human with a badge (camera to badge distance is 1.0 meter).

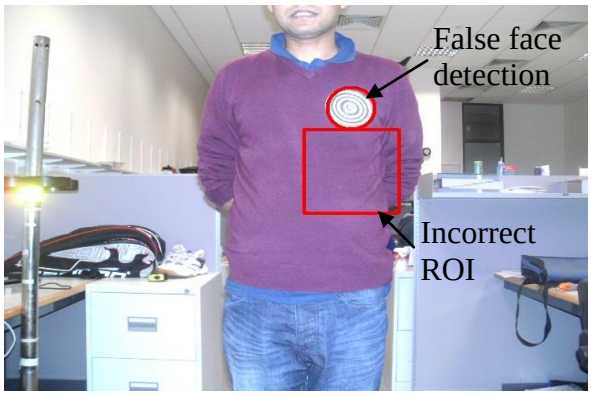


Figure 5. 9: False detection of a face and incorrect ROI labelling using the haar face detection classifier.

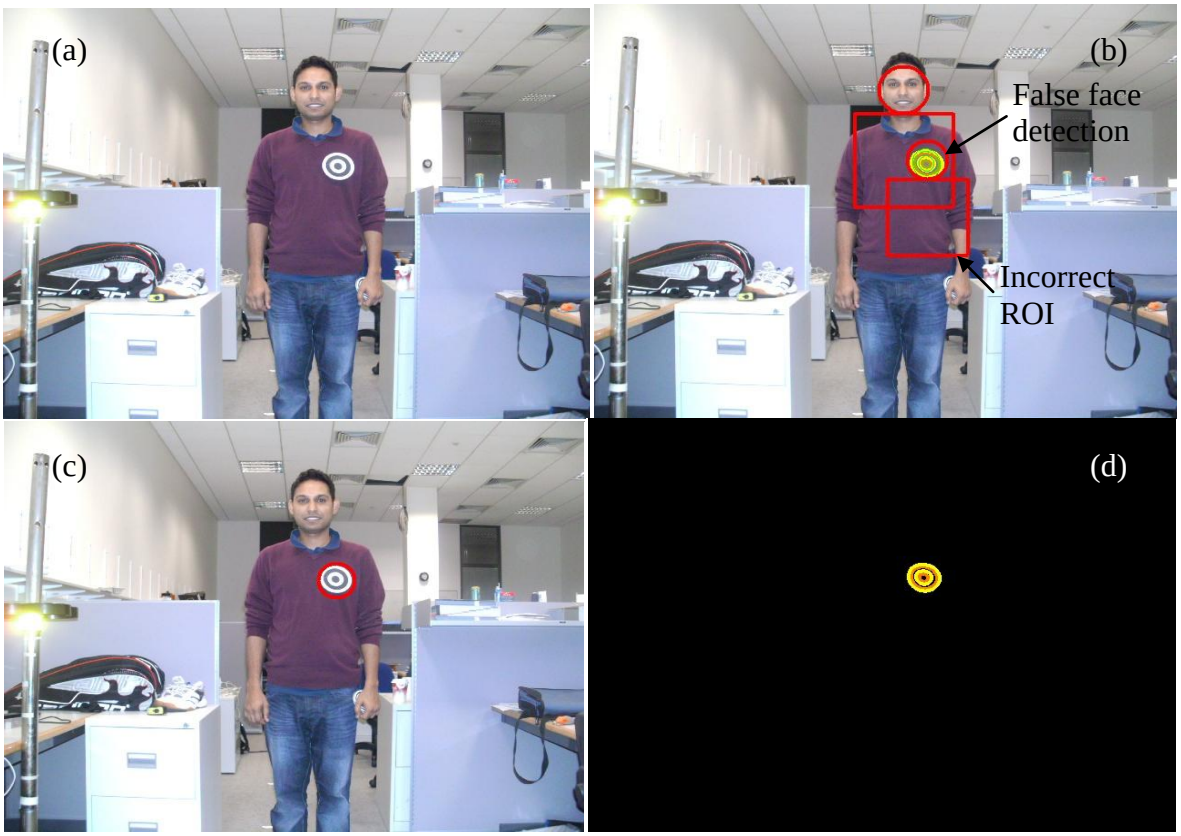


Figure 5. 10: (a) Target frame containing a human and a badge (camera to badge distance is 1.5 m, good lighting condition), (b) face detection and ROI labelling, (c) Detected badge using the developed badge detection classifier and (d) Detected badge using contour analysis.

The present study found that the 1.5 to 4.0 meters range is optimal for badge detection using both techniques (classifier and contour analysis). Haar based detection can work for distances

from 1.5 to 4 meters whereas the contour based detector performs from 1.5-5 meters (see Figures 4.10 and 4.11). Less than 1.5 meters and beyond 5.0 meters range none of the badge detection techniques work unless using the PTZ mechanism.

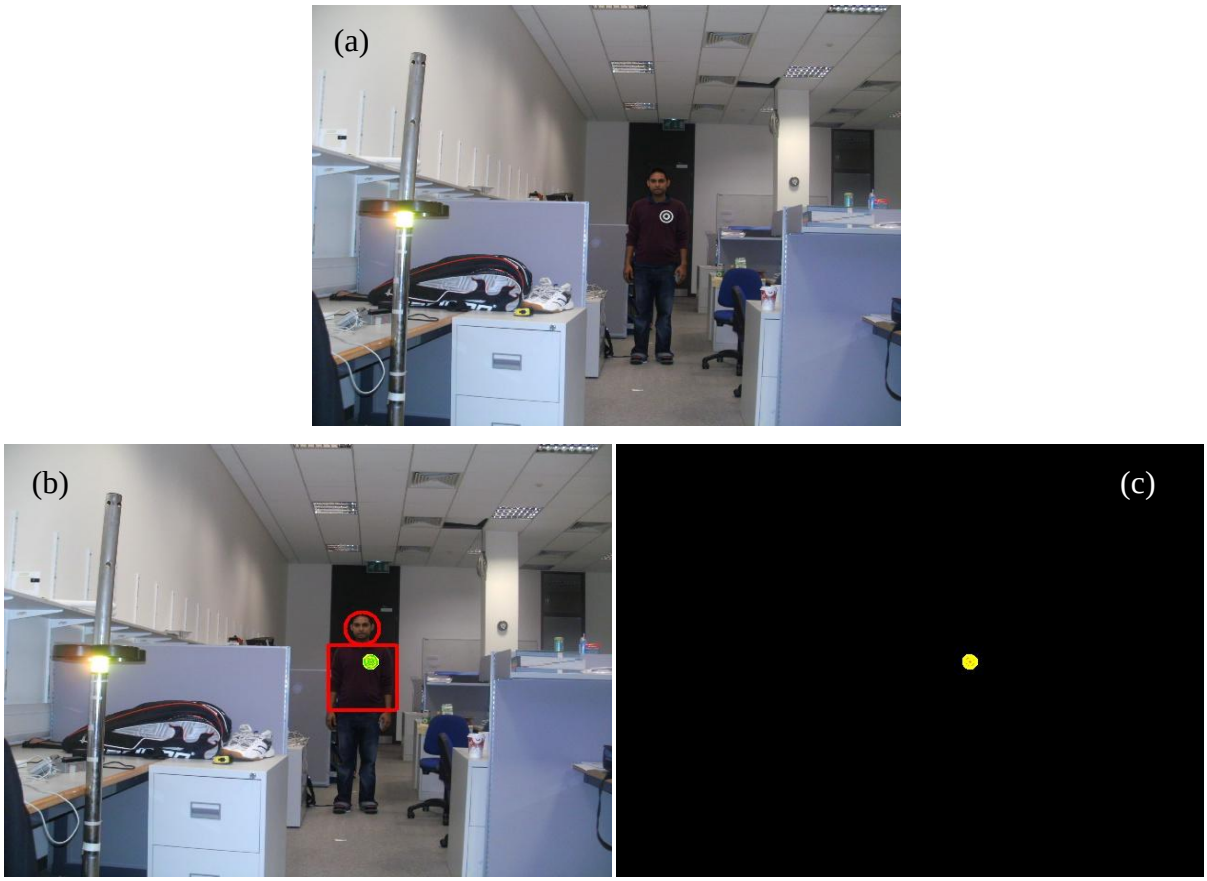


Figure 5. 11: (a) Target frame containing a human and a badge (camera to badge distance is 5 m, low lighting condition), (b) face detection and ROI labelling and (c) Detected badge using contour analysis.

Another experiment was carried out to determine the robustness of the badge detector to variances in rotation around the vertical axis. It was found that the badge can be detected within a ±50º range. Figure 5.12 shows two input images and the detected badge where the badges are rotated at the maximum angles of ±50º. Furthermore it is also tested that the badge detection is possible within the extreme left and right of the FOV (Figure: 5.13, 5.14, and 5.15) which also proved that camera calibration fixed any radial distortion on the periphery of the image frame that can affect the shape of the badge in the frame.

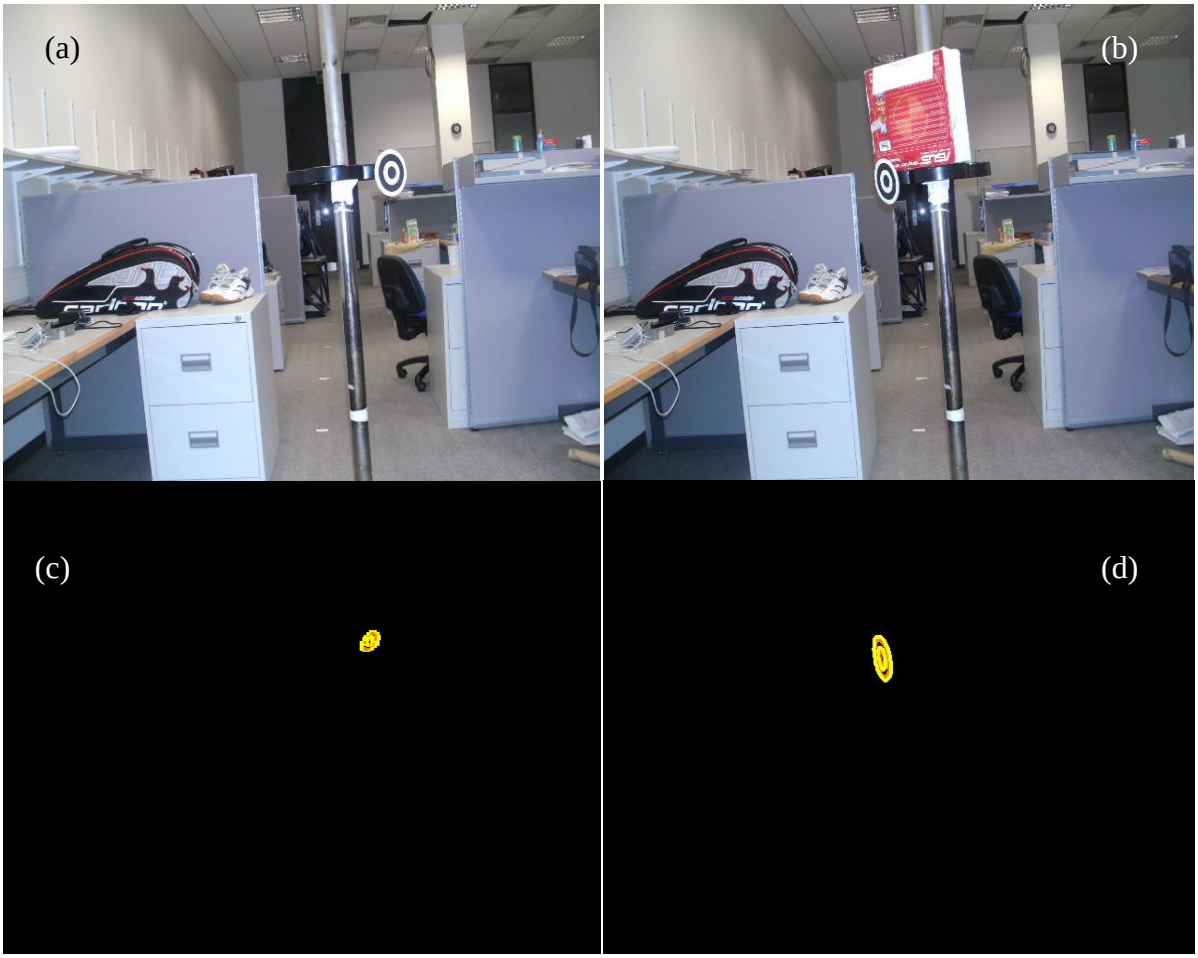


Figure 5. 12: (a), (b) Target frame containing a badge rotated at ±50º relative to the horizontal axis (camera to badge distance is 1.5 meters, moderate lighting condition) and (c), (d) Detected badge using contour analysis.

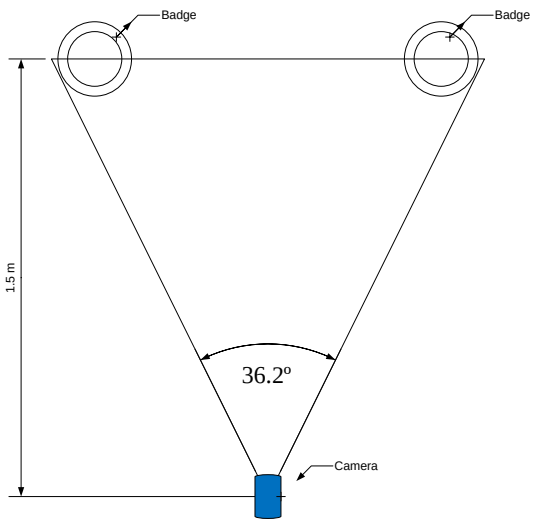


Figure 5. 13: Schematic diagram for extreme badge positioning relative to the camera.

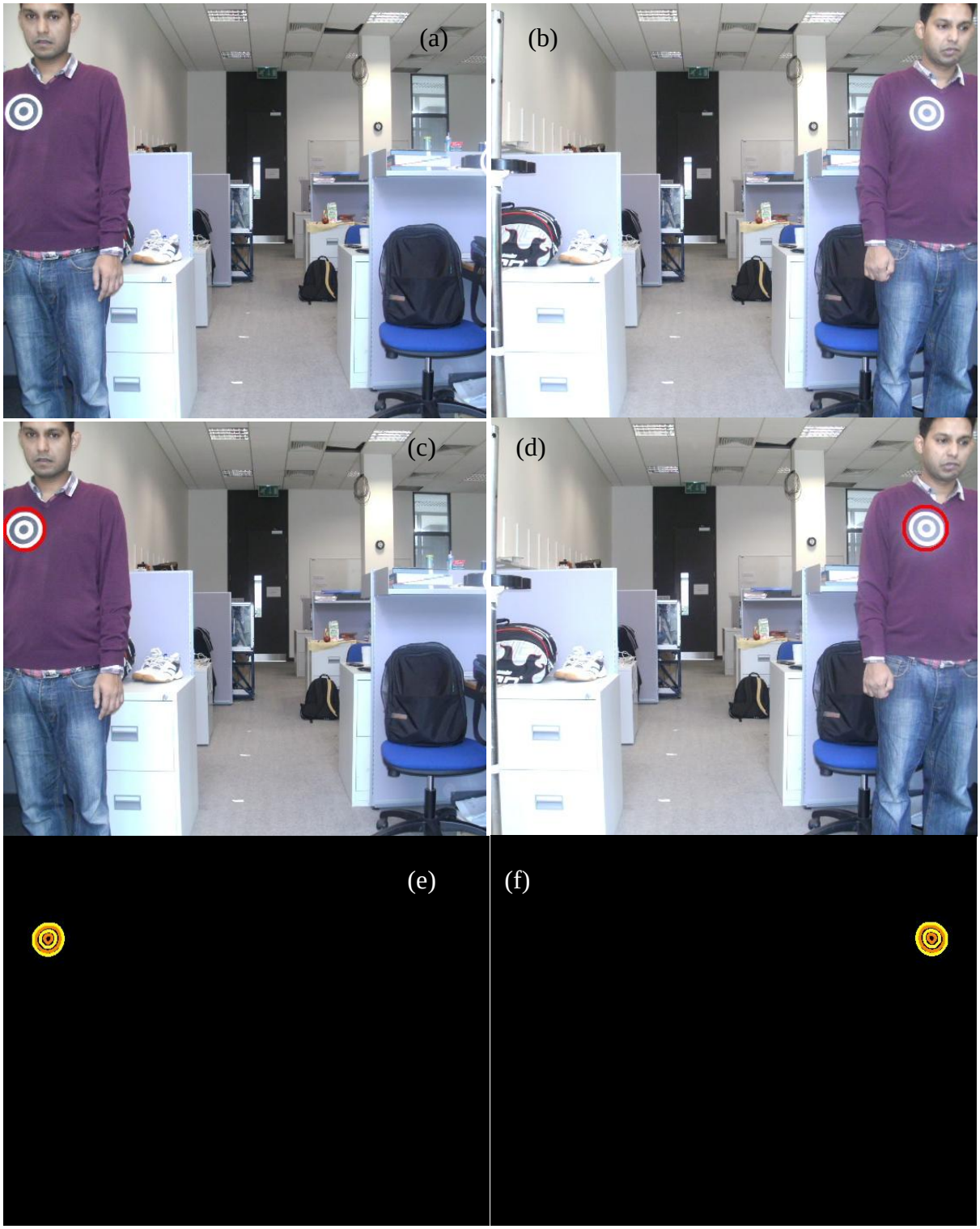


Figure 5. 14: (a), (b) Target frame containing a face and a badge at extreme left and right of the camera FOV (camera to badge distance is 1.5 m, moderate lighting condition), (c), (d) Detected badge using the developed badge classifier and (e), (f) Detected badge using contour analysis.

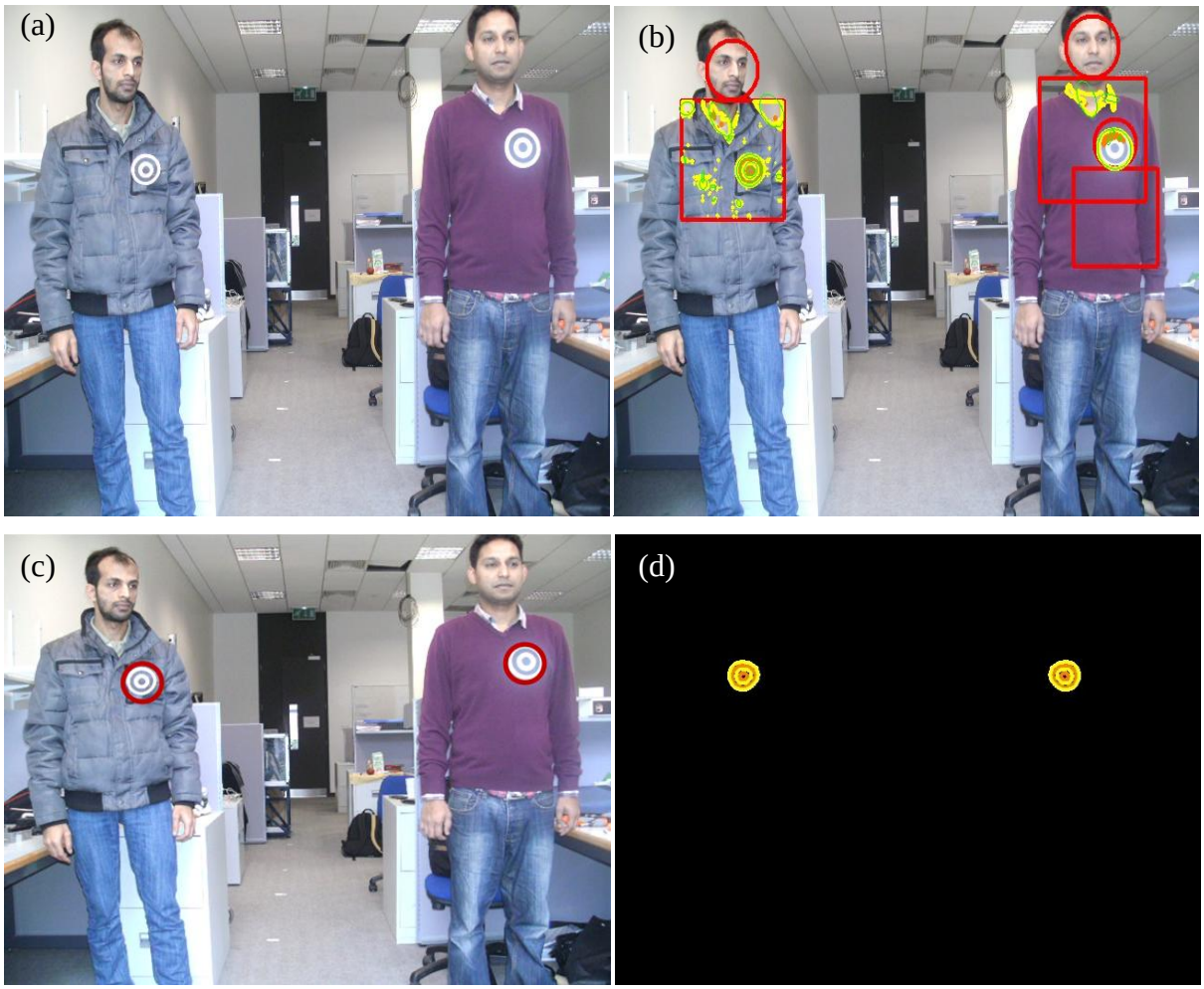


Figure 5. 15: (a) Target frame containing multiple persons and badges at extreme left and right of the camera FOV (camera to badge distance is 1.5 meters, moderate lighting condition), (b) Face detection and ROI labelling (multiple), (c) Detected badge using the proposed badge classifier and (d) Detected badges using contour analysis.

To test the system performance, a total number of 1000 image frames (670 with badges and 330 without) containing also persons were examined. Out of those sample frames 955 were detected correctly (i.e. if the image contains badges, the system detected them; if a face is found and it does not contains any badges within the ROI it leads to the detected face to be a potential intruder raising event alarm for the intruder detection). The intruder detection system was unable to detect 45 of the image frames accurately but it was observed that the face and badge of those frames were located far from the camera point of view and/or the face was not visible. The overall intruder monitoring system performance rate obtained was 95.5% within the range of 1.5 to 5 meters distance.

5.4.2.2 Recognising patients on the floor

The final classification decision is made by the Artificial Neural Network (ANN). In an experiment, the ANN was trained with 1007 positive samples and 503 negative samples. Data containing the FCC, DFCC, convexity defects and hue-saturation properties of the object (i.e. Human and non human classes) are the input to the multilayered network as outlined in the previous chapter. Neural Network parameters are as follows:

- Number of hidden nodes: 37
- Learning rate: 0.2
- Momentum term: 0.2
- Random rate with which the connections between the neurons were initialise: 0.1
- Maximum average error at which to stop the training: 0.005

Figure 5.16 presents an ANN training error graph per iteration. The prediction of future sample decisions (i.e. '1' for human lying on the floor and '0' otherwise) using the neural network consists of teaching the history of the variable in a selected limited sample data and applying the taught information to other examples.

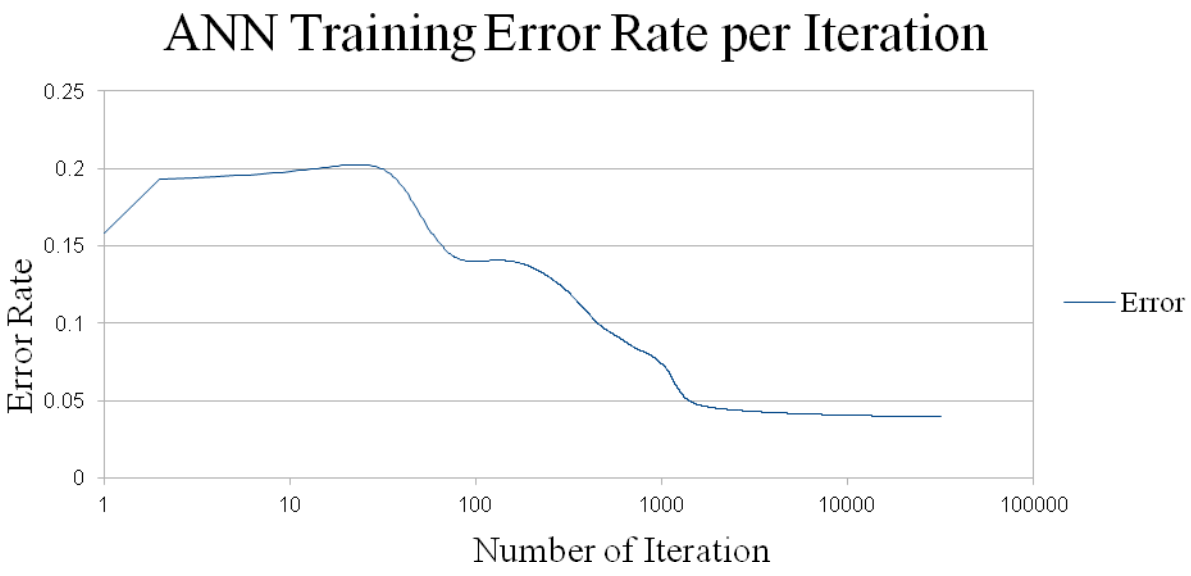


Figure 5. 16: ANN Training error rate per iteration.

Figure 5.17 presents some of the used blob of positive samples (i.e. patient lying on the floor) and Figure 4.17 presents some of the used blob of negative samples (i.e. no patient lying on the floor). The parameters detailed in section 4.4.6.6 were used during the training and testing phases.

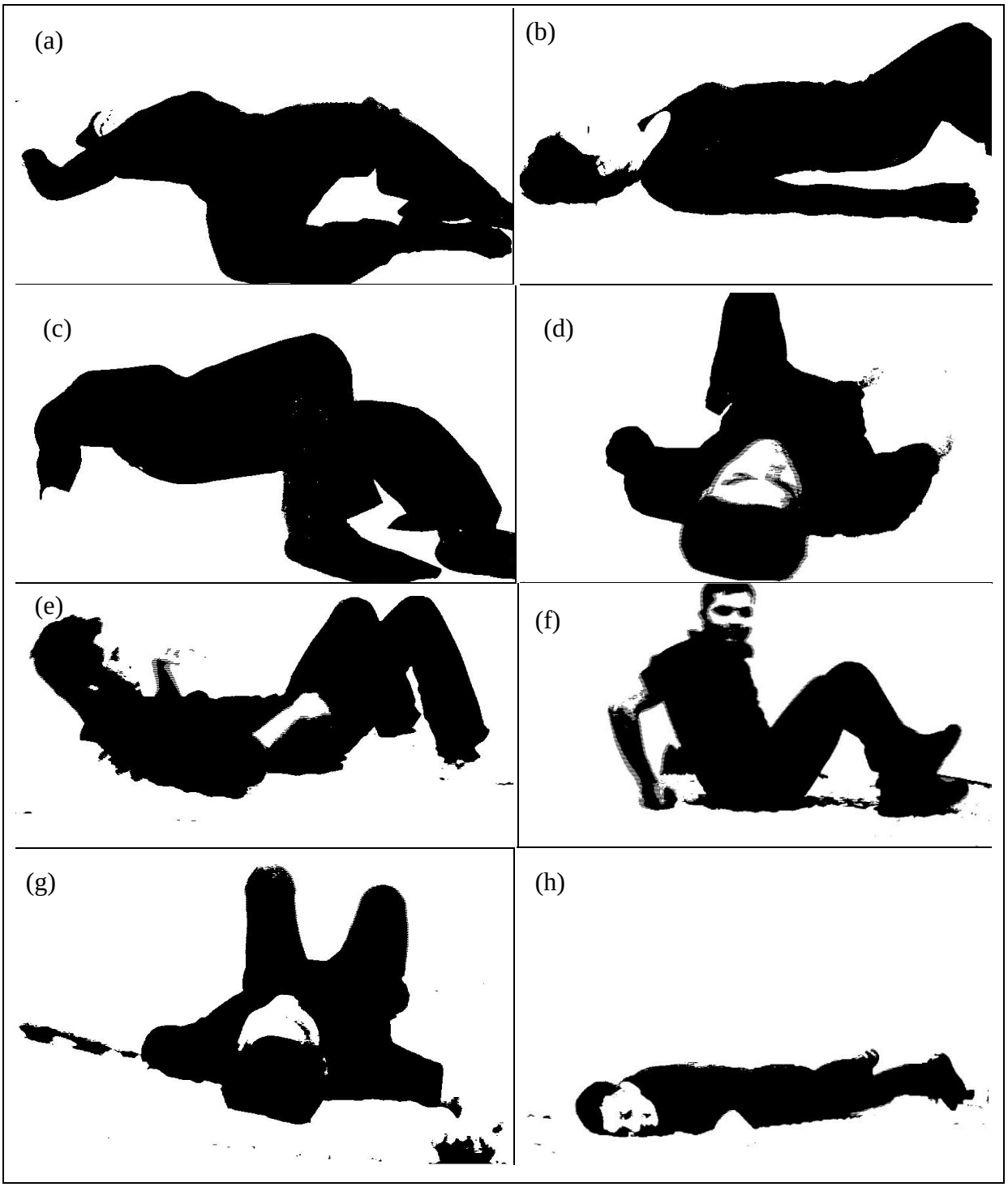


Figure 5. 17: Examples of positive blob samples (patient lying on the floor).

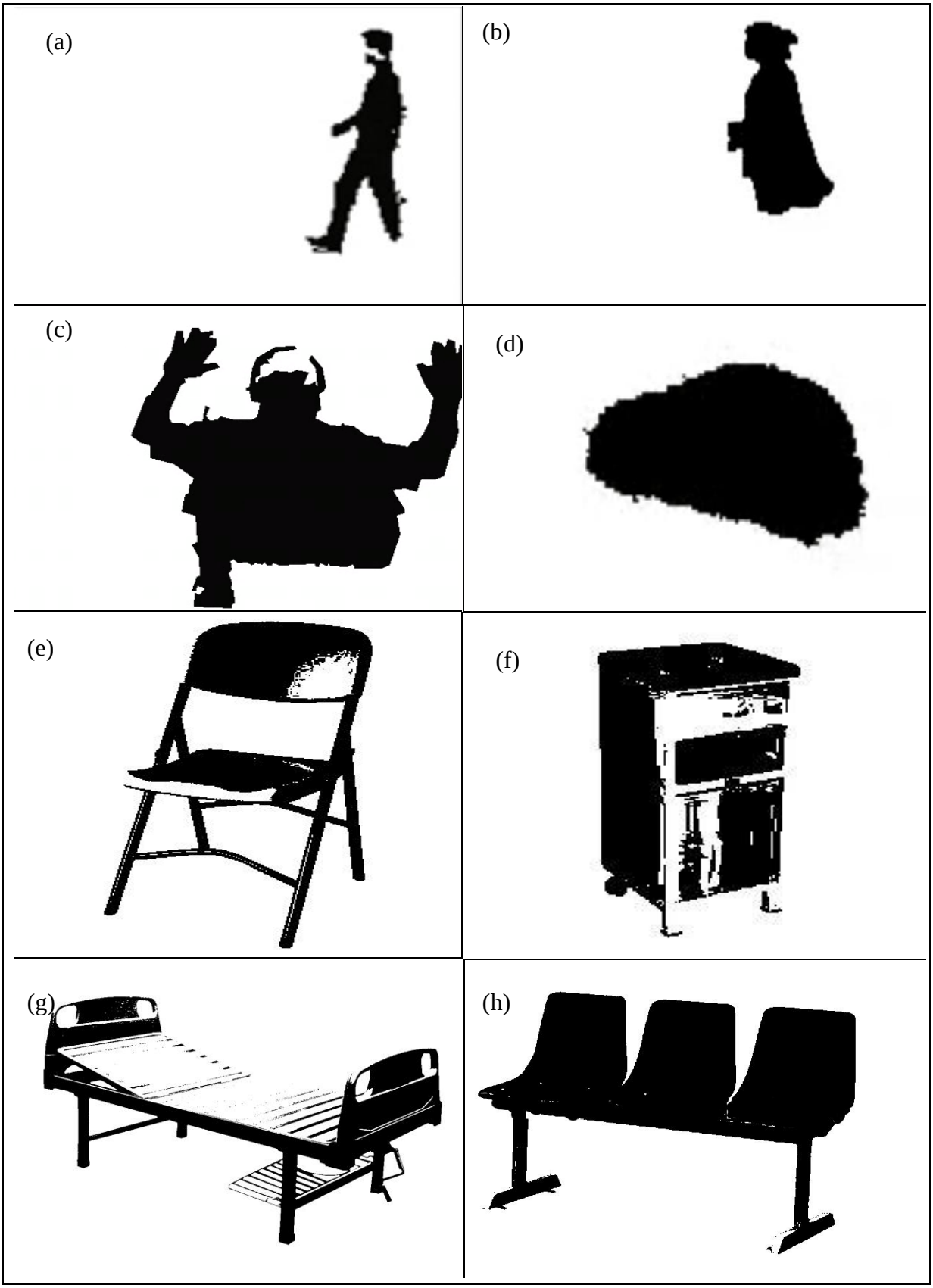


Figure 5. 18: Examples of negative blob samples (no patient lying on the floor) (a, b, c) Non-target class human objects and (d, e, f, g, h) Non-target static class objects.

The trained ANN was tested using 550 positive and 325 negative samples (which did not take part in the training process). Table 5.3 presents some results of the ANN decision. In the table, lines marked with green or yellow are false detections: green marking for patients lying on the floor who were not detected, and yellow marking for wrongly identified non-target object as patients on the floor.

Table 5. 3: Results of recalling the ANN.

Example [100]: Desired Output=1.000000 Calc.OUTPUT=0.032862 Diff=0.967138
Example [380]: Desired Output=1.000000 Calc.OUTPUT=0.999118 Diff=0.000882
Example [639]: Desired Output=1.000000 Calc.OUTPUT=1.000000 Diff=0.000000
Example [740]: Desired Output=0.000000 Calc.OUTPUT=0.000000 Diff=0.000000
Example [772]: Desired Output=0.000000 Calc.OUTPUT=1.000000 Diff=1.000000

More data samples and results are added in Appendix A. If the difference found between the desired and predicted classifications, the tolerance given to a correct decision was set to +/- 0.5 %. These desired output values of each test images were manually annotated (i.e. either 1/0). The ANN successfully classifies 821 test samples out of 875. The system classification accuracy is obtained 93.8%. After investigating the respective images as for the rest of the miss-classifications by the ANN, it is due to the inherent imperfections of the ANN itself and that perhaps not enough training samples were used with enough variations to encapsulate all the conditions of the testing samples.

5.4.3 Patient physical condition monitoring module

This section describes the data analysis and results of the patient physical condition monitoring module. The gathered sensor data (previously outlined in previous chapter) were analyzed to check the system performance and reliability.

The Equvital sensor provides the heart rate, which have been sampled in each data packet at a certain interval. Table 5.4 shows the sensor data logging pattern sample of the Equvital sensor while it was attached to a person to measure the vital signs. Figures 5.19, 5.20, and 5.21 show the graph of the sensor data for heart rate, respiration rate and skin temperature.

Table 5. 4: Sample of the Equvital sensor data logging chart.

Date/Time	HR ECG	Rb	Skin Temperature	Device S/N (Patient ID)
########	66	17	25.3	607007
########	66	17	25.1	607007
########	67	17	25.5	607007
########	66	17	25.6	607007
…				

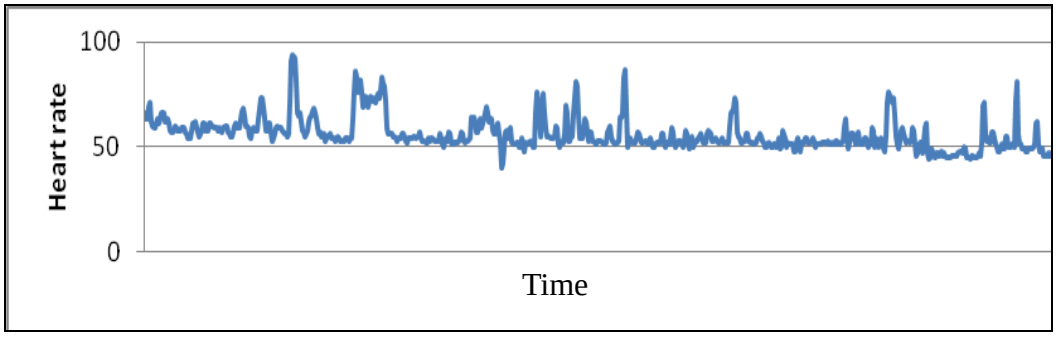


Figure 5. 19: Example of a graph of heart rate.

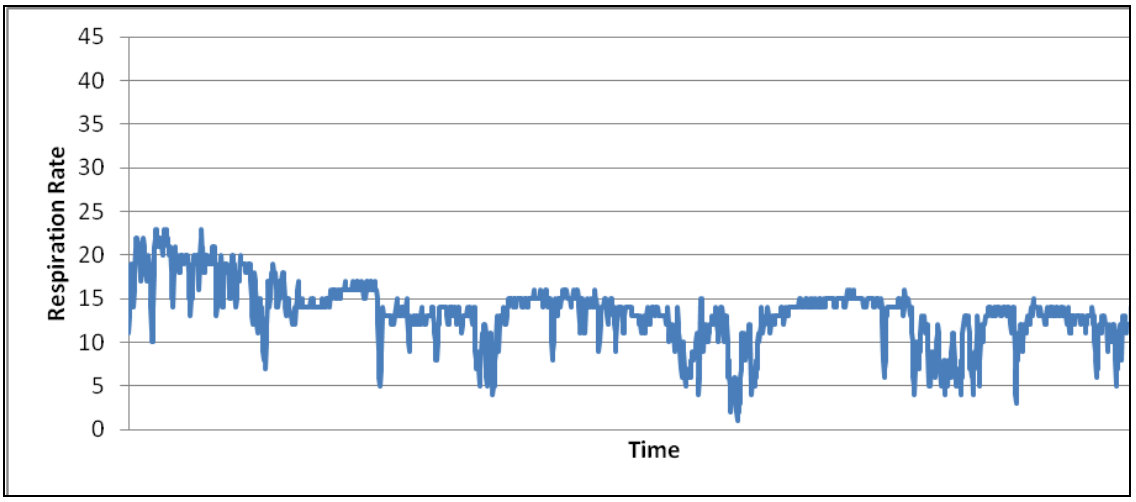


Figure 5. 20: Example of a graph of respiration rate.

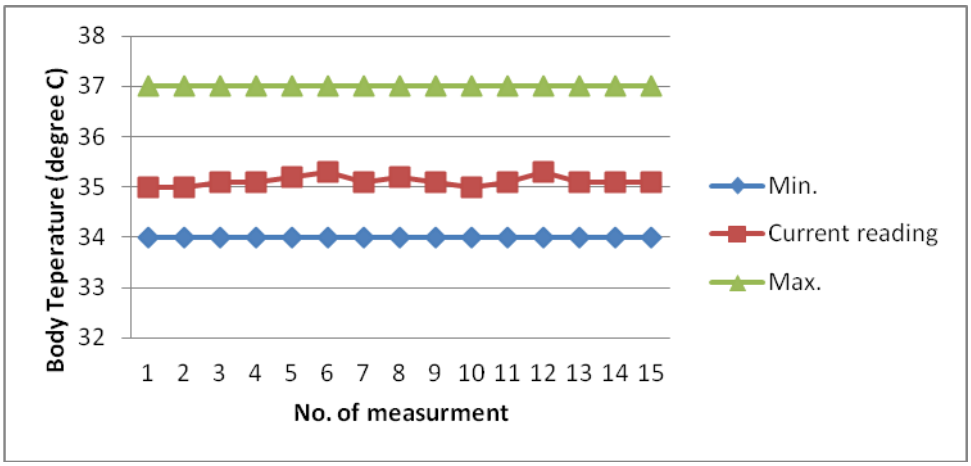


Figure 5. 21: Example of a graph of skin temperature compared with the normal range values.

5.5 Plug-and-Play tests

Plug-and-play tests were mainly performed during the hospital tests of the robot system. After the completion of the iWARD project the mobile robots remained with their developers, so a simulated hardware manager component was developed in order to be able to test what happens when a module is inserted into one of the drawers of the robot. In all cases the module software components were able to register themselves with the hardware manager after the Gumstix embedded computers of the modules had booted up.

In order to obtain sample test data from the modules, an external software component is required. Since (as was explained earlier) those external components are usually installed on the computers of the mobile robot bases, test software components were developed for each service module to mimic the interaction between the service modules and the external components through the Orca interfaces.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Thesis summary

In the framework of this research project, the following three service modules were developed: environmental condition monitoring, situation recognition (intruder monitoring and patient detection on the floor) and patient physical condition monitoring. The service modules can be attached to any of the hospital robots that were developed in the iWARD project. The developed modules are fully interchangeable; they have standard mechanical, power and communication connections with the mobile robot platforms.

Communication between the module software components with other components of the system (like the hardware manager and the mission controller, both running on the main computers of the robots) is done using pre-defined interfaces of the Orca middleware software. Each module can be controlled externally. They also have self-monitoring capabilities. They can operate in powersave mode when inactive.

The environment monitoring module monitors several parameters of the hospital environment: temperature, humidity and smoke and raises an alarm if an event occurs. This module can provide dynamic sensor data of the hospital environment. The measured data is stored in the shared knowledge of the robot system and can be used for optimisation purposes.

The situation recognition module uses non-intrusive machine vision-based concepts, which can monitor the environment in order to detect an intruder, or a patient lying on the floor. The module can correctly identify persons (potential intruders and/or patients lying on the floor) in the majority of cases.

The patient condition monitoring system remotely measures the following patient physical conditions: body temperature, heart rate, respiratory rate, and others, using sensors attached to the patient's body and able to raise an alarm in the case of any abnormality detected. The module can be used as part of general monitoring or it can gather data from specific patients.

The overall system is effective in terms of price and performance. Manufacturing cost would be low comparing to the purchase cost of specialised service robots.

The outcome of this research has significant contribution to the robotics application area in the hospital environment. Overall, there are several notable results in this work.

The sensors of the developed modules mounted on a mobile robot platform provide dynamic data for surveillance monitoring, as opposed to approaches using stationary sensors. All the developed modules are standalone, interchangeable and showed their pioneer applications in the healthcare sector.

This study provides a state-of-the-art concept of non-intrusive machine vision which introduced new techniques of image analysis (pattern recognition) to detect and classify objects. Advanced imaging techniques include dynamic thresholding, sensor fusion, machine learning, advanced contour analysis techniques, and others.

Image sensor fusion was achieved by adding depth (distance) information to 2D colour images. This can dramatically ease foreground/background separation and makes object recognition and classification much more robust.

A novel differential chain code has been developed to characterise contours. The differential chain code histogram can eliminate the problems inherent to the standard chain codes; it can describe a contour much more effectively.

A new object recognition and classification system has been developed. It is based on a trained artificial neural network with inputs that characterise shapes such as Freeman chain code histogram, differential chain code histogram, hue-saturation histogram and convexity defects histogram.

The modules were successfully tested both in laboratory and hospital environments.

6.2 Conclusions

The iWARD project and this research proved the concept of using a large number of small, self-navigating robots equipped with interchangeable service modules in hospitals. From the results of this research, the following major conclusions can be drawn:

- The concept of interchangeable robot service modules, equipped with standard mechanical, power and communication connections, can increase the flexibility of the

robot system. It allows for easy reconfiguration, improved scheduling, easy expansion of the system with new or improved modules.

- The self-contained service modules, equipped with their own computing equipment, are capable of solving the tasks of the identified hospital scenarios. They are able to communicate with the other software components of the robot system trough pre-defined interfaces.

- The three developed service modules for environment monitoring, patient condition monitoring, and situation recognition, can effectively contribute to improving hospital healthcare. They can also improve the working conditions of hospital staff and reduce staff shortages.

- The research project, as a whole, has achieved its major goals. The developed modules can cater for all scenarios that were identified in the project specification.

6.3 Future work

Although the robot service modules have proved to be functioning properly, further enhancements would increase their robustness. The following are recommendations for future research:

- Improved laser sensor: the sensor that was used in this research was of the first commercially available series from the company. Although the sensor was a breakthrough in imaging technologies, it still had one major flaw: distance measurement depended on the reflectivity of the measured surface. This was especially emphasised with highly reflective surfaces. In order to eliminate this negative effect, it was necessary to compensate measurement errors manually. An improved laser sensor would reduce the effect of reflectivity and would increase measuring accuracy.

- Improved video camera: the Axis PTZ camera that was selected for the final version of the situation recognition module is a high-end professional video camera that can operate at low lighting conditions. However, it can only operate in the visible spectrum of the light waves, and at low lighting conditions measured colours are

distorted. This makes blob colour-based image separation difficult. Using a camera in the infrared range (with infrared lighting) would improve image recognition.

- Sensor fusion: perhaps the most effective way of sensor fusion would be to incorporate a video camera and a laser sensor into one physical sensor. Such a sensor could be calibrated in advance, which would eliminate the need for distortion compensation and transforming one image to match the other image. The output of such a sensor would, in fact, be a true fused signal that can provide both colour and depth information of an image seamlessly.

- Increased number of training samples: the generalisation capabilities of the artificial neural network (ANN) depend on the number of examples that the training set contains. The ANN in this research was trained with a fair amount of training samples, which resulted in a high correct-vs-incorrect classification ratio. Preparing training samples and training an ANN with a large number of samples is extremely time consuming, and time limitations did not allow increasing the number of training samples. Adding more training data would increase the robustness of the classifier system.

- Improved face detector: the face detector used in this research copes best with frontal face images; the ratio of misclassification increases with faces in profile. An improved face detection system should be developed. Again, this is a time-consuming task that was out of scope of this research.

- Additional service modules: this research only concentrated on the development of three service modules. However, module interchangeability allows for easy expansion of the system with new (or modified) modules. A possible option would be the development of a remote patient consultation module that could expand the capabilities of the developed patient condition monitoring module. This module could even utilise some of the existing equipment (like the video camera); the camera has a built-in web server and allows easy access to it through the internet, so it can easily be teleoperated from a distance.

PUBLICATIONS ARISING FROM THIS WORK

Journal Papers

 T. Szecsi, K. A. Mamun, K. Hasan, A. Islam, C. Griffin, M. Hoque. 2010. Hospital Robot Development in the iWARD Project, *VIMation Journal,* (Issue-1), pp.6-12.

Conference Papers

 T. Szecsi, K. A. Mamun, K. Hasan. 2012. Situation Recognition Module for Hospital Robots. 13th Mechatronics forum international conference, Linz, Austria (In Preparation).

 K. A. Mamun, T. Szecsi, K. Hasan.2011. Surveillance module, A Robotics Application in the Hospital environment. 28th *International Manufacturing Conference*, Dublin, Ireland, pp. 272-279.

 T. Szecsi, K. A. Mamun, K. Hasan, A. Islam, C. Griffin, M. Hoque. 2010. Service Modules for Hospital Robots. *In T. Kennel, ed., Proceedings of the 12-th Mechatronics Forum Biennial International Conference (MECHATRONICS 2010),* ETH, Zurich, Switzerland, pp. 108 - 113.

 K. A. Mamun, R. Sherry, and T. Szecsi. 2009. Vision system of a hospital robot surveillance module. *Proceedings of the 5-th Virtual International Conference on Intelligent Production Machines and Systems (IPROMS),* Cardiff, UK, pp. 310-315.

 Mamun, K. and Szecsi, T., 2008. Development of a surveillance module for hospital robots. *In: D.T. Pham et al., eds., Proceedings of the 4-th Virtual International Conference on Intelligent Production Machines and Systems (IPROMS),* Cardiff, UK, Elsevier Ltd., Oxford, pp. 567-572.

 T. Szecsi, K. A. Mamun, K. Hasan, A. Islam, C. Griffin, M. Hoque. 2008. Hospital Robot Module Development in the IWARD Project. *In: Proceedings of the 6th CIRP International Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '08*, Naples, Italy.

REFERENCES

1. W. Jia; W. Bo-yu; Z. Ye; C. Hu. 2009. Design of an autonomous mobile robot for hospital, *IEEE International Symposium on IT in Medicine & Education*, vol.1, pp.1181-1186.

2. T. Szecsi, K. A. Mamun, K. Hasan, A. Islam, C. Griffin, M. Hoque. 2010. Hospital Robot Development in the iWARD Project, *VIMation Journal,* (Issue-1), pp.6-12.

3. European information society and TNO innovation for life, *Robotics for Healthcare*, [Online] Available from: http://ec.europa.eu/information_society/activities/health/docs/studies/robotics_healthcare/robotics-in-healthcare.pdf [Accessed December 2010].

4. B. Maurits. et. al., Robotics for Healthcare, Final report, *TNO innovation for life.* [Online] http://www.tno.nl/downloads/TNOKvL_report_RoboticsforHealthcare1.pdf [Accessed December 2010].

5. Skilligent, *Healthcare/Eldercare Robot based on Skilligent Technology.* [Online] Available from: http://www.skilligent.com/products/documents/docs/Skilligent-Health-Elder-Care-Robot-Study.pdf [Accessed December 2010].

6. R. Ambrose, R. S. Askew. 1995. An Experimental Investigation of Actuators for Space Robots, IEEE *International Conference on Robotics and Automation*, pp. 2625 – 2630.

7. A. T. Hayes, A. Martinoli, R.M. Goodman. 2002. Distributed Odour Source Localization. *IEEE Sensors Journal*, Vol. 2(3).

8. G. J. Monkman, and R. D. Taylor. 1993 .Thermal Tactile Sensing. *IEEE Transactions on Robotics and Automation*, Vol. 9.

9. M.Y Shieh, J.C. Hsieh, C.P. Cheng. 2004. Design of an intelligent hospital service robot and its applications. *IEEE International Conference on Systems, Man and Cybernetics,* Vol.5, pp. 4377- 4382.

10. M. Takahashi, T. Suzuki, F. Cinquegrani, R. Sorbello, E. Pagello. 2009. A mobile robot for transport applications in hospital domain with safe human detection algorithm. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp.1543-1548.

11. C. Rafflin, A. Fournier. 1995. Learning with a friendly interactive robot for service tasks in hospital environments. *IEEE/RSJ International Conference on Human Robot Interaction and Cooperative Robots Proceedings*, Vol.3, pp.492-497.

12. University of Massachusetts Amherst, *uBot-5.* [Online] Available from: http://www-robotics.cs.umass.edu/Robots/UBot-5 [Accessed December 2010].

13. University of Massachusetts Amherst, *Robots assist with elder care.* [Online] Available from: http://www.cs.umass.edu/csinfo/announce/robots_elder.html [Accessed December 2010].

14 . Adept Technology Inc., *Robot Applications.* [Online] Available from: http://www.mobilerobots.com/RobotApplications.aspx [Accessed January 2011].

15. Y Y. U. Cao, A. S. Fukunaga and A. B. Khang. 1997. Cooperative mobile robotics: antecedents and directions. *Autonomous Robots,* Vol. 4 (1), pp. 7–27.

16. R. A. Brooks. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, vol.2 (1), pp. 14–23.

17. M. J. Matari´c. 1995. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, Vol.16, pp. 321–331.

18. J. K. Rosenblatt. 1997. DAMN: A distributed architecture for mobile navigation. *Journal of Experimental Theoretical Artificial Intelligence*, pp. 339–360.

19. L. E. Parker. 1997. L-ALLIANCE: Task-oriented multi-robot learning in behavioral-based systems. *Advanced Robotics*, Vol.11 (4), pp.305–322.

20. B. R. Donald. 1995. Information invariants in robotics. *Artificial Intelligence*, Vol. 72 (1–2), pp.217–304.

21. L. E. Parker. 1998. Toward the automated synthesis of cooperative mobile robot teams, *Proceeding of the SPIE Mobile Robots XIII*, Vol. 3525, pp. 82–93.

22. Jacarta. [Online] Available from: http://jacarta.com/Home.aspx [Accessed October 2011]

23. Dickson. [Online] Available from: http://www.dicksondata.com [Accessed October 2011]

24. Lighthouse worldwide solutions. [Online] Available from: http://www.golighthouse.com/default.asp, [Accessed October 2011]

25. OPEN XTRA. [Online] Available from: http://www.openxtra.co.uk, [Accessed October 2011]

26. Dickson, *A Guide to Hospital Environmental Monitoring*. [Online], http://cosmeticsurgerytimes.modernmedicine.com/cosmeticsurgerytimes/data/articlestandard//cosmeticsurgerytimes/232009/600986/article.pdf, [Accessed October 2011].

27. W. Hu, T. Tan, L. Wang, and S. Maybank. 2004. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, And Cybernetics - Part C: Applications And Reviews*, Vol. 34, No. 3, pp. 334-352.

28. M. Valera and S. A. Velastin. 2005. Intelligent distributed surveillance systems: a review. I*EEE Proceedings of Vision, Image, and Signal Processing*, Vol. 152, No. 2, pp. 192-204.

29. A. Milella, G. Cicirelli, and A. Distante. 2008. RFID-Assisted Mobile Robot System for Mapping and Surveillance of Indoor Environments. *Industrial Robot: An International Journal*, 35(2), pp. 143-152.

30. Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, Y.-Y. Hsieh. 2007. iMouse: An Integrated Mobile Surveillance and Wireless Sensor System. *IEEE Computer*, Vol. 40, No. 6, pp. 60-66.

31. irobot. [Online] Available from: http://www.irobot.com [Accessed December 2010].

32. Centre for applied Autonomous Sensor System, *Robotic security guard.* [Online] Available from: http://www.oru.se/aass [Accessed December 2010].

33. P. Biber, S. Fleck, T. Duckett. 2005. 3D Modeling of Indoor Environments for a Robotic Security Guard. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR Workshop*, pp.124-124.

34. J. Garcia, N. V. Lobo, M. Shah, J. Feinstein. 2005. Automatic detection of heads in colored images. *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, pp. 276– 281.

35. S. G. Miaou, P. H. Sung, C.Y. Huang. 2006. A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information. *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare*, pp. 39-42.

36. H. Nait-Charif, S.J. McKenna. 2004. Activity summarisation and fall detection in a supportive home environment. *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 4, pp. 323 – 326.

37. V. Girondel, L.Bonnaud, A. Caplier, M. Rombaut. 2005. Static human body in video sequences using the belief theory. *IEEE International Conference on Image Processing*, Vol. 2, pp. II- 45-8.

38. B. Bernard, B. Francois, T. Monique. 2006. Applying 3D human model in a posture recognition system, *Pattern Recognition Letters, Vision for Crime Detection and Prevention*, Vol. 27 (15), pp. 1788-1796.

39. N. D. Guilherme and C. K. Avinash. 2002. Vision for Mobile Robot Navigation: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 237-267.

40 . H. K. Nishihara. 1983. PRISM: A Practical Real-time Imaging Stereo Matcher. *Proceedings of SPIE*, pp. 134-142.

41. S. B. Pollard, J. E. W. Mayhew and J. P. Frisby. 1991. PMF: A Stereo Correspondence Algorithm Using a Disparity Gradient Limit. *IN*: John E. W. Mayhew, John P. Frisby (eds.) *9D Model Recognition from Stereoscopic Cues*. The MIT Press, Cambridge, pp. 11-24.

42. H. Moribe, M. Nakano, T. Kuno and J. Hasegawa. 1987. Image Preprocessor of Model-based Vision System for Assembly Robots. *Proceedings of IEEE International Conference on Robotics and Automation*, pp.366-371.

43. T. Braunl. 2003. *Embedded Robotics*, Springer-Verlag, Berlin, Heidelberg.

44. O. Faugeras. 1993, *Three-dimensional Computer Vision*. The MIT Press.

45. R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Visio*, 2nd edition. Cambridge University Press.

46. Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck. 1998. Smart pixel - photonic mixer device (PMD), *Proceedings of International Conference on Mechatronic and Machine Vision*, pp. 259-264.

47. R. Lange. 2000. 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology. Ph.D. dissertation, University of Siegen.

48. JVC. *Zcam.* [Online] Available from: http://pro.jvc.com/prof/attributes/features.jsp?model_id=MDL101309 [Accessed December 2010].

49. K. Barbara.1998. Assessing Vital Signs. *Fundamentals of Nursing*, (5th ed.), Addison – Wesley, Canada, pp.425-448.

50. DRAEGER. *Patient monitoring device*. [Online] Available from: http://www.draeger.co.uk [Accessed March 2011].

51. Equivital, [Online] Available from: http://www.equivital.co.uk/Equivital%20Systems/Sensors.asp [Accessed March 2011].

52. Mind Media B.V., Netherlands. [online] Available from: www.mindmedia.nl/english/nexus4.php dia.nl [Accessed March 2011].

53. NexGen Ergonomics Inc., U.S.A. [online] Available from: www.nexgenergo.com/medical/procompinf.html [Accessed February 2011].

54. M. V. Bergh and L. V. Gool. 2011. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. *Computer Vision (WACV)*, pp. 66-72.

55. D. M. Gavrila, J. Giebel, and S. Munder. 2004. Vision-based pedestrian detection: The protector system. *Proceddings of IEEE Intelligent Vehicle Symposium*, pp.13–18.

56. T. Fong, I. Nourbakhsh, and K. Dautenhahn. 2003. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42 (3-4), pp. 143–166.

57. M. J. Mataric, J. Eriksson, D.J. Feil-Seifer and C.J. Winstein. 2007. Socially assistive robotics for post-stroke rehabilitation. *Journal of Neuroengineering and Rehabilitation*, 4 (5).

58. H. Asoh, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, N. Vlassis, R. Bunschoten, and B. Krose. 2001 . An office robot that communicates and learns. *IEEE Intelligent Systems*, 16 (5), pp.46–55.

59. W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun. 1999. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114, pp. 1–2.

60. N. Dalal and B. Triggs. 2005. Histograms of Oriented Gradients for Human Detection. *In IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886 – 893.

61. A. Mohan, C. Papageorgiou, and T. Poggio. 2001. Example-based object detection in images by components. *IEEE Conference on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 349–361.

62. Y. Ke and R. Sukthankar. 2004. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 506–513.

63. A. A. Mykhaylo, R. Stefan Roth, and S. Bernt. 2009. Pictorial Structures Revisited: People Detection and Articulated Pose Estimation. *IEEE Conference on Computer Vision and Pattern Recognition,* pp.1014-1021.

64. Q. Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. 2006. Fast human detection using a cascade of histograms of oriented gradients. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1491–1498.

65. W. Zhang, G. Zelinsky, and D. Samaras. 2007. Real-time accurate object detection using multiple resolutions. *IEEE 11th International Conference on Computer Vision,* pp.1-8.

66. J. Begard, N. Allezard, and P. Sayd. 2008. Real-time human detection in urban scenes: Local descriptors and classifiers selection with adaboost-like algorithms. *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8.

67. S. Avidan. 2007. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 261–271.

68. H. Grabner and H. Bischof. 2006. On-line boosting and vision. *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 260-267.

69. B. Wu and R. Nevatia. 2007. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, vol. 75, pp. 247–266.

70. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. 1997 . Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, pp. 780-785

71. I. Haritaoglu, D. Harwood and L. S. Davis. 1998. W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People. *Proceedings of the International Conference on Face and Gesture Recognition*.

72. S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. 2000. Tracking Groups of People. *Computer Vision and Image Understanding*, Vol. 80, pp. 42-56.

73. J. Connell, A.W. Senior, A. Hampapur, Y-L Tian, L. Brown and S. Pankanti. 2004. Detection and Tracking in the IBM People Vision System. *IEEE ICME*.

74. L. M. Fuentes and S. A. Velastin. 2001. People Tracking in surveillance application. *Proceddings 2nd IEEE International Workshop on PETS*.

75. X. Zabulis, H. Baltzakis, A. Argyros. 2009. Vision-based Hand Gesture Recognition for Human-Computer Interaction. In: *The Universal Access Handbook*, LEA.

76. S. S. Beauchemin and J. L. Barron. 1995. The Computation of Optical flow. *ACM Computing Surveys*, Vol.27, pp. 433 – 466.

77. Y. L. Tian and A. Hampapur. 2005. Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance. *Proceedings of IEEE Computer Society Workshop on Motion and Video Computing*.

78. C. Stauffer and W. E. L. Grimson. 1999. Adaptive background mixture models for real tracking. *International Conference on Computer Vision and Pattern Recognition*, Vol.2, pp.246-252.

79. A. Elgammal, R. Duraiswami, D. Harwood and L. Davis. 2002. Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance. *Proceeding of the IEEE*, Vol.90, No.7.

80. K. Kim, T. H. Chalidabhongse, D. Harwood and L. Davis. 2004. Background Modeling and Subtraction by Codebook Construction. *IEEE International Conference on Image Processing (ICIP)*.

81. L. Zhao and C. E. Torpe. 2000. Stereo- and Neural Network Based Pedestrian Detection,. *IEEE Trans. Intelligent Transportation System*, Vol.1, No.3. pp.148-154.

82. C. BenAbdelkader and L. Davis. 2002. Detection of People Carrying Objects : a Motion-based Recognition Approach. *5th IEEE International Conference on Automatic Face and Gesture Recognition*.

83. J. W. Lee, M. S. Kim, I. S. Kweon. 1995. A Kalman filter based visual tracking algorithm for an object moving in 3D. *IEEE International Conference on Intelligent Robots and System*, Vol.1.

84. A. Senior, A. Hampapur, Y. L. Tian, L. Brown, S. Pankanti and R. Bolle. 2001. Appearance Models for Occlusion Handling. *Proceedings of Second International workshop on Performance Evaluation of Tracking and Surveillance systems in conjunction with CVPR'01*.

85. M. B.Capellades, D. DeMenthon and D. Doermann. 2004. An Appearance-based Approach for Consistent Labeling of Humans and Objects in Video. *Pattern Analysis and Applications.*

86. C. E. Erden, B. Sankur and A. M. Tekalp. 2004. Performance Measure for Video Object Segmentation and Tracking. *IEEE Transactions Image Processing*, Vol.13, No.7, pp.937-951.

87. X. N. Zhang; J. Jiang; Z. Liang; C.L. Liu. 2010. Skin colour enhancement based on favourite skin colour in HSV colour space. *IEEE Transactions on Consumer Electronics*, vol.56, no.3, pp.1789-1793.

88. N. Ogale. 2006. A survey of techniques for human detection from video. Master's thesis, *University of Maryland*.

89. M. Bennewitz, F. Faber, D. Joho, M. Schreiber, and S. Behnke. 2005. Towards a humanoid museum guide robot that interacts with multiple persons. *5th IEEERAS Int. Conf. on Humanoid Robots*, pp. 418–423.

90. G. Hollinger, Y. Georgiev, A. Manfredi, B. Maxwell, Z. Pezzementi, and B. Mitchell. 2006. Design of a social mobile robot using emotion-based decision mechanisms. *International Conference on Intelligent Robots and Systems.*

91. R. Munoz-Salinas, E. Aguirre, and M. Garcia-Silvente. 2007. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25 (6), pp. 995–1007.

92. S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G.A. Fink and G. Sagerer. 2003. Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot. *International Conference on Multimodal Interfaces*, pp. 28–35.

93. D. Geronimo, A. Lopez and A. Sappa. 2007. Computer vision approaches to pedestrian detection: Visible spectrum survey. *Lecture Notes in Computer Science*, 4477. pp. 547–554.

94. T. Gandhi and M. M. Trivedi. 2007. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 8 (3), pp. 413–430.

95. D. Gavrila and S. Munder. 2007. Multi-Cue Pedestrian Detection and Tracking from a Moving Vehicle. I*nternational Journal of Computer Vision*, Vol. 73, No. 1, pp. 41-59.

96. D. Gavrila. 2001. Sensor-Based Pedestrian Protection. *IEEE Intelligent Systems*, vol. 16, no. 6, pp. 77-81.

97. A. N. Rajagopalan, P. Burlina and R. Chellappa. 1999. Detection of people in images. *Proceddings of the IEEE Int. Joint Conference on Neural Networks (IJCNN'99)*, Washington.

98. A. N. Rajagopalan and R. Chellappa. 2001. Higher order statistics-based detection of people/vehicles in images. *INSA-A Journal: Special Issue on Image Processing, Vision and Pattern Recognition,* 67, pp. 157-166.

99. S. Nayar and V. Branzoi. 2003. Adaptive Dynamic Range Imaging: Optical Control of Pixel Exposures over Space and Time. *Proceedings International Conference Computer Vision*, vol. 2, pp. 1168-1175.

100. S. Marsi, G. Impoco, A. Ukovich, S. Carrato, and G. Ramponi. 2007. Video Enhancement and Dynamic Range Control of HDR Sequences for Automotive Applications. *EURASIP Journal : Advances in Signal Processing*, p. 9.

101. P. Knoll. 2007. HDR Vision for Driver Assistance. *High-DynamicRange (HDR) Vision*, B. Hoefflinger, ed., pp. 123-136.

102. A. Broggi, M. Bertozzi, and A. Fascioli. 2001. Self-Calibration of a Stereo Vision System for Automotive Applications. *Proceedings IEEE International Conference Robotics and Automation*, pp. 3698-3703.

103. T. Dang and C. Hoffmann. 2004. Stereo Calibration in Vehicles. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 268-273.

104. M. Bertozzi, A. Broggi, M. Carletti, A. Fascioli, T. Graf, P. Grisleri, and M.-M. Meinecke. 2003. IR Pedestrian Detection for Advanced Driver Assistance Systems. Proceedings *of Pattern Recognition Symposium*, pp. 582-590.

105. A. Broggi, P. Grisleri, T. Graf, and M.-M. Meinecke. 2005. A Software Video Stabilization System for Automotive Oriented Applications. *Proceedings of Vehicular Technology Conference*, vol. 5, pp. 2760-2764.

106. L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani. 2006. An Evaluation of Monocular Image Stabilization Algorithms for Automotive Applications. *Proceedings IEEE International Conference Intelligent Transportation Systems*, pp. 1562-1567.

107. D. Hoiem, A. Efros, and M. Hebert. 2006. Putting Objects in Perspective. *Proceedings IEEE International Conference Computer Vision and Pattern Recognition*, vol. 2, pp. 2137-2144.

108. A. Sappa, F. Dornaika, D. Ponsa, D. Gero´nimo, and A. Lo´pez. 2008. An Efficient Approach to Onboard Stereo Vision System Pose Estimation. *IEEE Transactions on* I*ntelligent Transportation Systems*, vol. 9, no. 3, pp. 476-490.

109. A. Ess, B. Leibe, and L. VanGool. 2007. Depth and Appearance for Mobile Scene Analysis. *Proceedings International Conference Computer Vision*.

110. A. Ess, B. Leibe, K. Schindler, and L. VanGool. 2008. A Mobile Vision System for Robust Multi-Person Tracking. *Proceedings IEEE International Conference Computer Vision and Pattern Recognition.*

111. D. Gero´nimo, A. Sappa, A. Lo´pez, and D. Ponsa. 2007. Adaptive Image Sampling and Windows Classification for On-Board Pedestrian Detection. *Proceedings 5th International Conference Computer Vision Systems.*

112. C. Papageorgiou and T. Poggio. 2000. A Trainable System for Object Detection. *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15-33.

113. C. Wojek, G. Dorko´ , A. Schulz, and B. Schiele. 2008. Sliding-Windows for Rapid Object Class Localization: A Parallel Technique. *Proceedings of Symposium German Association for Pattern Recognition*, pp. 71-81.

114. F. Miau, C. Papageorgiou, and L. Itti. 2001. Neuromorphic Algorithms for Computer Vision and Attention. *Proceedings of International Symposium on Optical Science and Technology*, pp. 12-23.

115. L. Itti, C. Koch, and E. Niebur. 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254-1259.

116. A. Broggi, A. Fascioli, I. Fedriga, A. Tibaldi, and M.D. Rose. 2003. Stereo-Based Preprocessing for Human Shape Localization in Unstructured Environments. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 410-415.

117. A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. 2000. Shape-Based Pedestrian Detection. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 215-220.

118. M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. 2003. Shape-Based Pedestrian Detection and Localization. *Proceedings IEEE International Conference* Intelligent Transportation Systems, pp. 328-333.

119. M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse. 2004. Pedestrian Localization and Tracking System with Kalman Filtering. *Proceedings IEEE* I*ntelligent Vehicles Symposium*, pp. 584-589.

120. M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M.-M. Meinecke. 2004. Pedestrian Detection for Driver Assistance Using Multiresolution Infrared Vision. *IEEE Transactions of Vehicular Technology*, vol. 53, no. 6, pp. 1666-1678.

121. F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. 2006. Pedestrian Detection Using Infrared Images and Histograms of Oriented Gradients. *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 206-212.

122. A. Broggi, R. Fedriga, A. Tagliati, T. Graf, and M.-M. Meinecke. 2006. Pedestrian Detection on a Moving Vehicle: An Investigation about Near Infra-Red Images. *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 431-436.

123. T. Tsuji, H. Hattori, M.Watanabe, and N. Nagaoka. 2002. Development of Night-Vision System. *IEEE Transactions Intelligent Transportation Systems*, vol. 3, no. 3, pp. 203-209.

124. Q. Tian, H. Sun, Y. Luo, and D. Hu. 2005. Nighttime Pedestrian Detection with a Normal Camera Using SVM Classifier. *Proceedings of international Symposium on Neural Networks*, pp. 189-194.

125. M. Oberla¨nder. 2005. Hypermutation Networks—A Discrete Approach to Machine Perception. *Proceedings of 3rd Weightless Neural Networks Workshop*.

126. U. Meis, M. Oberla¨nder, and W. Ritter. 2004. Reinforcing the Reliability of Pedestrian Detection in Far-Infrared Sensing. *Proceedings IEEE Intelligent Vehicles Symposium,* pp. 779-783.

127. M. Ma¨hlisch, M. Oberla¨nder, O. Lo¨hlein, D. Gavrila, and W. Ritter. 2005. A Multiple Detector Approach to Low-Resolution FIR Pedestrian Recognition. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 325-330.

128. I. Haritaoglu, D. Harwood, and L. S. Davis. 2000. W4 : Real-time surveillance of people and their activities. I*EEE Transactions Pattern Anal. Machine Intelligent*, vol. 22, pp. 809–830.

129. S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. 2000. Tracking groups of people. *Computer Vision Image Understanding*, Vol. 80, No. 1, pp. 42–56.

130. C. Stauffer and W. Grimson. 1999 . Adaptive background mixture models for real-time tracking. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252.

131. A. J. Lipton, H. Fujiyoshi, and R. S. Patil. 1998. Moving target classification and tracking from real-time video. *Proceedings IEEE Workshop Applications of Computer Vision*, pp. 8–14.

132. D. Meyer, J. Denzler, and H. Niemann. 1998. Model based extraction of articulated objects in image sequences for gait analysis. *Proceedings IEEE International Conference on Image Processing*, pp. 78–81.

133. D. Meyer, J. Psl, and H. Niemann. 1998. Gait classification with HMM's for trajectories of body parts extracted by mixture densities. *Proceedings British Machine Vision Conference*, pp. 459–468.

134. H. Elzein, S. Lakshmanan, and P. Watta. 2003. A Motion and ShapeBased Pedestrian Detection Algorithm. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 500-504.

135. J. Barron, D. Fleet, and S. Beauchemin. 1994. Performance of optical flow techniques. I*nternational Journal of Computer Vision*, vol. 12, no. 1, pp. 42–77.

136. B. Leibe, N. Cornelis, K. Cornelis, and L. VanGool. 2007. Dynamic 3D Scene Analysis from a Moving Vehicle. *Proceedings IEEE conference on Computer Vision and Pattern Recognition*, pp. 1-8.

137. I. Haritaoglu, M. Flickner, D. Beymer. 2002. Ghost3D: detecting body posture and parts using stereo. *Workshop on Motion and Video Computing*, Orlando, Florida.

138. K. Hayashi, M. Hashimoto, K. Sumi, K. Sasakawa. 2004. Multiple-person tracker with a fixed slanting stereo camera. *International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea.

139. Y. A. Ivanov, A. F. Bobick, J. Liu. 2000. Fast lighting independent background subtraction. I*nternational Journal of Computer Vision*, Vol. 37, No. 2. pp. 199–207.

140. S. N. Lim, A. Mittal, L. S. Davis, N. Paragios. 2005. Fast illuminationinvariant background subtraction using two views: error analysis, sensor placement and applications. *Computer Vision and Pattern Recognition,* San Diego, CA.

141. M. T. Yang, Y. C. Shih, S.C. Wang. 2004. People tracking by integrating multiple features. *International Conference on Pattern Recognition,* Cambridge, UK.

142. S. Iwase, H. Saito. 2004. Parallel tracking of all soccer players by integrating detected positions in multiple view images. I*nternational Conference on Pattern Recognition,* Cambridge, UK.

143. A. Mittal, L. S. Davis. 2002. M2Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene using regionbased stereo. *European Conference on Computer Vision,* Copenhagen, Denmark.

144. A. Mittal, L. S. Davis. 2003. M2Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, Vol. 51, No. 3, pp. 189–203.

145. D. B. Yang, H. H. G. Banos, L. J. Guibas. 2003. Counting people in crowds with a real-time network of simple image sensors. *International Conference on Computer Vision,* Nice, France.

146. G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe. 2004. 3D Vision Sensing for Improved Pedestrian Safety. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 19-24.

147. S. Krotosky and M. M. Trivedi. 2007. On Color-, Infrared-, and Multimodal-Stereo Approaches to Pedestrian Detection. *IEEE Transactions Intelligent Transportation Systems*, Vol. 8, No. 4, pp. 619-629.

148. R. Labayrade, D. Aubert, and J. Tarel. 2002. Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry through ‘vDisparity’ Representation. P*roceedings IEEE Intelligent Vehicles Symposium*, vol. 2, pp. 17-21.

149. L. Zhao and C. Thorpe. 2000. Stereo and Neural Network-Based Pedestrian Detection. *IEEE Transactions Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148-154.

150. M. Soga, T. Kato, M. Ohta, and Y. Ninomiya. 2005. Pedestrian Detection with Stereo Vision. *Proceedings IEEE International Conference on Data Engineering Workshops*, pp. 1200.

151. S. Krotosky and M.M. Trivedi. 2007. Multimodal Stereo Image Registration for Pedestrian Detection. *Proceedings IEEE International Conference on Intelligent Transportation Systems*, pp. 109-114.

152. D. Comaniciu, V. Ramesh, P. Meer. 2003. Kernel-based object tracking. I*EEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 25, No. 5, pp. 564–575.

153. M. Hu, W. Hu, T. Tan. 2004. Tracking people through occlusion. *International Conference on Pattern Recognition,* Cambridge, UK.

154. S. J. McKenna, S. Jabri, Z. Duric, H. Wechsler. 2000. Tracking Interacting People. I*nternational Conference on Automatic Face and Gesture Recognition,* Grenoble, France.

155. K. Okuma, A. Taleghani, N. D. Freitas, J. J. Little. David G. Lowe, 2004. A boosted particle filter: multitarget detection and tracking. *European Conference on Computer Vision,* Prague, Czech Republic.

156. L.Q. Xu, P. Puig. 2005. A hybrid blob- and appearance-based framework for multi-object tracking through complex occlusions. *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* Beijing, China.

157. T. Zhao, R. Nevatia. 2004. Tracking multiple humans in crowded environments. C*omputer Vision and Pattern Recognition,* Washington DC, USA.

158. J. Kang, I. Cohen, G. Medioni. 2005. Persistent objects tracking across multiple non-overlapping cameras. *IEEE Workshop on Motion and Video Computing (MOTION'05), Breckenridge, Colorado.*

159. S. Khan, M. Shah. 2000. Tracking people in presence of occlusion. *Asian Conference on Computer Vision,* Taipei, Taiwan.

160. D. Roth, P. Doubek, L.V. Gool. 2005. Bayesian pixel classification for human tracking. *IEEE Workshop on Motion and Video Computing (MOTION'05),* Breckenridge, Colorado.

161. C. Yang, R. Duraiswami, L. Davis. 2005. Fast multiple object tracking via a hierarchical particle filter. *International Conference on Computer Vision, Beijing,* China.

162. B. H. Sotelo, A. Oliva, and A. Torralba. 2005. Human Learning of Contextual Priors for Object Search. *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 86-93.

163. A. Torralba and A. Oliva. 2007. The Role of Context in Object Recognition. *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520-527.

164. D. Hoiem, A. Efros, and M. Hebert. 2008. Closing the Loop in Scene Interpretation. *Proceedings IEEE Conference Computer Vision and Pattern Recognition.*

165. D. Gavrila. 2000. Pedestrian Detection from a Moving Vehicle. *Proceedings European Conference on Computer Vision*, Vol. 2, pp. 37- 49.

166. H. Nanda and L. Davis. 2002. Probabilistic Template Based Pedestrian Detection in Infrared Videos. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 15-20.

167. S. Munder and D. Gavrila. 2006. An Experimental Study on Pedestrian Classification. I*EEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 28, No. 11, pp. 1863-1868.

168. P. Viola, M. Jones, and D. Snow. 2003. Detecting Pedestrians Using Patterns of Motion and Appearance. *Proceedings International Conference on Computer Vision*, Vol. 2, pp. 734-741.

169. M. Jones and D. Snow. 2008. Pedestrian Detection Using Boosted Features over Many Frames. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition.*

170. V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

171. F. Xu, X. Liu, and K. Fujimura. 2005. Pedestrian Detection and Tracking with Night Vision. I*EEE Transactions Intelligent Transportation Systems*, Vol. 6, No. 1, pp. 63-71.

172. A. Mohan, C. Papageorgiou, and T. Poggio. 2001. Example-Based Object Detection in Images by Components. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 23, No. 4, pp. 349-361.

173. L. Andreone, F. Bellotti, A.D. Gloria, and R. Lauletta. 2005. SVM-Based Pedestrian Recognition on Near-Infrared Images. *Proceedings 4th Internationl Symposium on Image and Signal Processing and Analysis*, pp. 274-278.

174. L. Zhang, B. Wu, and R. Nevatia. 2007. Pedestrian Detection in Infrared Images Based on Local Shape Features. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

175. C. Wojek and B. Schiele. 2008. A Performance Evaluation of Single and Multi-Feature People Detection," Proceeding DAGM Symposium, pp. 82-91.

176. Y. Freund and R. Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences,* Vol. 55, No. 1, pp. 119-139.

177. R. Schapire and Y. Singer. 1999. Improved Boosting Algorithms Using Confidence-Rated Predictions. *Machine Learning*, Vol. 37, No. 3, pp. 297-336.

178. J. Friedman, T. Hastie, and R. Tibshirani. 2000. Additive Logistic Regression: A Statistical View of Boosting. *Annals of Statistics*, Vol. 28, No. 2, pp. 337-407.

179. C. Huang, H. Ai, B. Wu, and S. Lao. 2004. Boosting Nested Cascade Detector for Multi-View Face Detection. *Proceedings International Conference Pattern Recognition*, Vol. 2, pp. 415-418.

180. X. Xu and E. Frank. 2004. Logistic Regression and Boosting for Labeled Bags of Instances. *Proceedings Pacific Asia Conference Knowledge Discovery and Data Mining*.

181. C. Bishop. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.

182. M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. 2005. Pedestrian Detection with Convolutional Neural Networks. *Proceeding IEEE Intelligent Vehicles Symposium*, pp. 224-229.

183. K. Levi and Y. Weiss. 2004. Learning Object Detection from a Small Number of Examples: The Importance of Good Features. *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 53- 60.

184 . D. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110.

185. S. Maji, A. Berg, and J. Malik. 2008. Classification Using Intersection Kernel Support Vector Machines is Efficient. *Proceeding IEEE Conference Computer Vision and Pattern Recognition.*

186. B. Wu and R. Nevatia. 2007. Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature Based Classifier. *Proceeding IEEE Conference Computer Vision and Pattern Recognition*, pp. 1-8.

187. A. Shashua, Y. Gdalyahu, and G. Hayun. 2004. Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance. *Proceeding IEEE Intelligent Vehicles Symposium*, pp. 1-6.

188. P. Felzenszwalb, D. McAllester, and D. Ramanan. 2008. A Discriminatively Trained, Multiscale, Deformable Part Model. *Proceeding IEEE Conference Computer Vision and Pattern Recognition*.

189. P. Dolla´r, B. Babenko, S. Belongie, P. Perona, and Z. Tu. 2008. Multiple Component Learning for Object Detection. *Proceeding European Conference Computer Vision*, pp. 211-224.

190. Z. Lin and L. Davis. 2008. A Pose-Invariant Descriptor for Human Detection and Segmentation. *Proceeding European Conference Computer Vision,* Vol. 4, pp. 423-436.

191. P. Sabzmeydani and G. Mori. 2007. Detecting Pedestrians by Learning Shapelet Features. *Proceeding IEEE Conference Computer Vision and Pattern Recognition*, pp. 1-8.

192. N. Dalal. 2006. Finding People in Images and Videos. PhD thesis, *Institute National Polytechnique de Grenoble/INRIA Rho˚ne-Alpes*.

193. D. Comaniciu. 2003. An Algorithm for Data-Driven Bandwidth Selection. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 25, No. 2, pp. 281-288.

194. S. Agarwal A. Awan, and D. Roth. 2004. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 26, No. 11, pp. 1475-1490.

195. Weiming Hu; Tieniu Tan; Liang Wang; S. Maybank. 2004. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 34, No. 3, pp.334-352.

196. O. Javed and M. Shah. 2002. Tracking and object classification for automated surveillance. *Proceeding European Conference Computer Vision*, Vol. 4, pp. 343–357.

197. G. Welch and G. Bishop. 2002. An Introduction to The Kalman Filter. *Technical report, Dept. of Computer Science, University of North Carolina at Chapel Hill.*

198. L. Zhang, Y. Li, and R. Nevatia. 2008. Global Data Association for Multi-Object Tracking Using Network Flows. *Proceeding IEEE Conference Computer Vision and Pattern Recognition.*

199. S. Gammeter, A. Ess, T. Ja¨ggli, K. Schindler, B. Leibe, and L. VanGool. 2008. Articulated Multi-Body Tracking under Egomotion. *Proceeding European Conference Computer Vision.*

200. B. Leibe, A. Leonardis, and B. Schiele. 2008. Robust Object Detection with Interleaved Categorization and Segmentation. *International Journal of Computer Vision*, Vol. 77, No. 1-3, pp. 259-289.

201. M. Adnriluka, S. Roth, and B. Schiele. 2008. People-Tracking-byDetection and People-Detection-by-Tracking. *Proceeding IEEE Conference Computer Vision and Pattern Recognition.*

202. JPL. 1997. Traffic surveillance and detection technology development. Sensor Development Final Report, *Jet Propulsion Laboratory Publication*, No. 97-10.

203. J. Malik, S. Russell, J. Weber, T. Huang, and D. Koller. 1994. A machine vision based surveillance system for Californaia roads. *University of California, PATH project MOU-83 Final Report.*

204. H. Z. Ning, L.Wang, W. M. Hu, and T. N. Tan. 2002. Articulated model based people tracking using motion models. *Proceeding International Conference on Multi-Model Interface*s, pp. 115–120.

205. A. Galata, N. Johnson, and D. Hogg. 2001. Learning variable-length Markov models of behavior. *Computer Visio Image Understanding*, Vol. 81, No. 3, pp. 398–413.

206. Y. Wu and T. S. Huang. 2001. A co-inference approach to robust visual tracking. *Proceeding International Conference on Computer Vision*, Vol. II, pp. 26–33.

207. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel. 1994. Toward robust automatic traffic scene analysis in real-time. *Proceeding International Conference on Pattern Recognition*, Israel, pp. 126–131.

208 . J. Malik and S. Russell. 1997. Traffic Surveillance and Detection Technology Development: New Traffic Sensor Technology. University of California, Berkeley, *California PATH Research Final Report*, UCB-ITS-PRR-97-6.

209. N. Paragios and R. Deriche. 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions Pattern Analysis Machine Intelligent*, Vol. 22, pp. 266–280.

210. N. Peterfreund. 2000. Robust tracking of position and velocity with Kalman snakes. *EEE Transactions Pattern Analysis Machine Intelligent*, Vol. 22, pp. 564–569.

211. R. Polana and R. Nelson. 1974. Low level recognition of human motion. *Proceedings IEEE Workshop Motion of Non-Rigid and Articulated Objects*, Austin, TX, pp. 77–82.

212. B. Schiele. 2000. Vodel-free tracking of cars and people based on colour regions. *Proceedings IEEE International Workshop Performance Evaluation of Tracking and Surveillance*, Grenoble, France, pp. 61–71.

213. C. A. Pau and A. Barber. 1996. Traffic sensor using a colour vision method. *Proceedings SPIE—Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, vol. 2902, 1996, pp. 156–165.

214. B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. 1998. Areal-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Res.: Part C*, vol. 6, no. 4, pp. 271–288.

215. T. J. Fan, G. Medioni, and G. Nevatia. 1989. Recognizing 3-D objects using surface descriptions. *IEEE Transactions Pattern Recognition Machine Intelligent*, vol. 11, pp. 1140–1157.

216. D.-S. Jang and H.-I. Choi. 2000. Active models for tracking moving objects. *Pattern Reorganisation*, vol. 33, no. 7, pp. 1135–1146, 2000.

217. I. A. Karaulova, P. M. Hall, and A. D. Marshall. 2000. A hierarchical model of dynamics for tracking people with a single video camera. *Proceedings British Machine Vision Conference*, pp. 262–352.

218. S. Ju, M. Black, and Y. Yaccob. 1996. Cardboard people: a parameterized model of articulated image motion. *Proceedings IEEE International Conference Automatic Face and Gesture Recognition*, pp. 38–44.

219. S. A. Niyogi and E. H. Adelson. 1994. Analyzing and recognizing walking figures in XYT. *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pp. 469–474.

220. K. Rohr. 1994. Toward model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, vol. 59, no. 1, pp. 94–115.

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

222. C. Sminchisescu and B. Triggs. 2001. Covariance scaled sampling formonocular 3D body tracking. *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, Kauai, HI, pp. I:447–I:454.

223. R. Plankers and P. Fua. 2001. Articulated soft objects for video-based body modelling. *Proceedings International Conference Computer Vision,* Vancouver, BC, Canada, pp. 394–401.

224. T. N. Tan, G. D. Sullivan, and K. D. Baker. 1998. Model-based localization and recognition of road vehicles. *International Journal Computer Vision*, vol. 29, no. 1, pp. 22–25.

225. W. F. Gardner and D. T. Lawton, 1996. Interactive model-based vehicle tracking. *Transactions Pattern analysis Machine Intelligent*, vol. 18, pp. 1115–1121.

226. H. Yang, J. G. Lou, H. Z. Sun, W. M. Hu, and T. N. Tan. 2001. Efficient and robust vehicle localization. *Proceedings IEEE International Conference Image Processing*, pp. 355–358.

227. D. Koller, K. Daniilidis, and H.-H. Nagel. 1993. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal Computer Vision*, vol. 10, no. 3, pp. 257–281.

228. H. Kollnig and H.-H. Nagel. 1997. 3D pose estimation by directly matching polyhedral models to gray value gradients. *International Journal Computer Vision*, vol. 23, no. 3, pp. 283–302.

229. M. Haag and H.-H. Nagel. 1999. Combination of edge element and optical flow estimates for 3D-model-based vehicle tracking in traffic image sequences. *International Journal Computer Vision*, vol. 35, no. 3, pp. 295–319.

230. C.-Y. Chan and F. Bu. 2005. Literature Review of Pedestrian Detection Technologies and Sensor Survey. Technical report, *Institute of Transportation Studies, University of California at Berkeley*.

231. B. Fardi, U. Schuenert, and G. Wanielik. 2005. Shape and Motion Based Pedestrian Detection in Infrared Images: A Multi Sensor Approach. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 18-23.

232. C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto. 2007. A Lidar and Vision-Based Approach for Pedestrian and Vehicle Detection and Tracking. *Proceedings IEEE International Conference on Intelligent Transportation Systems*, pp. 1044-1049.

233. B. Sheng, D. Liang. 2006. Human Segmentation in a Complex Situation Based on Properties of the Human Visual System. *The Sixth World Congress on Intelligent Control and Automation, WCICA 2006*, Vol.2, pp.9587-9590.

234. M. Lindner, A. Kolb, K. Hartmann. 2007. Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images. *International Symposium on Signals, Circuits and Systems, ISSCS 2007*, vol.1, no., pp.1-4.

235. M. Bertozzi, A. Broggi, M. Felisa, G. Vezzoni, and M. DelRose. 2006. Low-Level Pedestrian Detection by Means of Visible and Far Infra-Red Tetra-Vision. *Proceedings IEEE Intelligent Vehicles Symposium*, pp. 231-236.

236. Y. Caspi and M. Irani. 2002. Spatio-temporalalignmentofsequences. *IEEE Transactions Pattern Analysis Machine Intellegent*, vol.24, pp.1409–1424.

237. T. H. Chang, S. Gong, and E. J. Ong. 2000. Tracking multiple people under occlusion using multiple cameras. Proceedings British Machine Vision Conference, pp. 566–576.

238. L. Z. Manor and M. Irani. 2002. Multi-view constraints on homographies. *IEEE Transactions Pattern Analysis Machine Intelligent*, vol. 24, pp. 214–223, Feb. 2002.

239. T. Schlegelis. 2008. Final word - Swarming nurses - It may not be long before swarms of tiny mobile robots will be giving a hand to nurses and medical orderlies in hospitals. *Computing & Control Engineering Journal*, vol.18(1), pp.48-48.

240. Newcastel university, [Online] Available from: http://www.ncl.ac.uk/mech/research/project/3109 [Accessed March 2011].

241. Gumstix. [Online] Available from: http://www.gumstix.com/store/product_info.php?products_id=201 [Accessed March 2011].

242. Temperature sensor. [Online] Available from: http://www.chipcatalog.com/Docs/FEE0878DF555FDFD1F46C41A7F735068.pdf?pass=587388419775348 [Accessed May 2011].

243. Sencera Co., Ltd.. Humidity sensor. [Online] Available from: http://www.sensorelement.com/humidity/808H5V5.pdf [Accessed May 2011].

244. T. Cleary. 2007. Residential Smoke Alarm Performance. *Building and Fire Research Laboratory, National Institute of Standards and Technology, UL Smoke and Fire Dynamics Seminar*.

245. Fire and Life Safety in Mission-Critical Applications. 2011. *Life Safety Magazine*.

246. Canadian Nuclear Society. [Online] Available from: http://media.cns-snc.ca/pdf_doc/ecc/smoke_am241.pdf [Accessed March 2011].

247. LabJack. USB DAQ. [Online] Available from: http://labjack.com/u3 [Accessed in September 2011].

248. RSIOADXR. DAQ Hardware. [Online] Available from: http://www.controlanything.com/Relay/Device/RSIOADXR [Accessed March 2011].

249. K. A. Mamun, S. Rayhan, R. Sherry and T. Szecsi . 2008. Development of a surveillance module for hospital robots. *Proceedings of the 4-th Virtual International Conference on Intelligent Production Machines and Systems (IPROMS)*, Cardiff, UK, Elsevier Ltd., Oxford, pp.567-572.

250. Canon VC-C4 camera. [Online] Available from: http://www.usa.canon.com [Aaccessed in September 2008].

252. Axis communication. [Online] Available from: www.axis.com [Accessed in June 2011].

253. G. Bradski and A. Kaehler. 2008. *Learning OpenCV*. O'Relly media Inc.USA.

254. Open CV. [Online] Available from: http://opencv.willowgarage.com [Accessed June 2011].

255. Z. Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.22, pp. 1330–1334.

256. D. C. Brown, 1971. Close-range camera calibration. *Photogrammetric Engineering*,

vol.37, pp.855–866.

257. I. Getting. 1993. The Global Positioning System. *IEEE Spectrum*, vol.30(12),pp.36–47.

258. M. Erol-Kantarci, H.T. Mouftah and S. Oktug. 2011. A Survey of Architectures and Localization Techniques for Underwater Acoustic Sensor Networks. *Communications Surveys & Tutorials, IEEE* , vol.13, no.3, pp.487-502.

259. M. Hazas and A. Ward. 2002. A novel broadband ultrasonic location system. *In Proceedings of UbiComp 2002: Ubiquitous Computing*, G¨oteborg, Sweden, pp. 264–280.

260. M. Hazas and A. Ward. 2007. Calibration of the intensity-related distance error of the PMD ToF-camera. *In Proceedings SPIE, Intelligent Robots and Computer Vision*, vol. 6764, pp. 6764–35.

261. Ifm electronics, PMD 3D. [Online] Available from: http://www.ifm.com/ifmirl/web/pmd.htm [Accessed March 2011].

262. Electronic123. PIR sensors. [Online] Available from: http://www.electronics123.com/s.nl/it.A/id.62/.f [Accessed March 2011].

263. Vivometrics. Vivoresponder. [Online] Available from: http://www.vivometrics.com/hazmat/about_the_system/what_is_the_lifeShirt_system.php [Accessed September 2008].

264. Taumaz. Sensium™. [Online] Available from: http://www.toumaz.com/products/sensium.html [Accessed September 2008].

265. cURL. [Online] Available from: http://curl.haxx.se/ [Accessed May 2009].

266. Xml Remote procedure calling. [Online] Available from: http://www.xmlrpc.com [Assessed May 2011].

267. Driver Download. [Online] Available from: http://sourceforge.net/projects/xmlrpcpp [Accessed May 2011].

268. R. Kjeldsen and J. Kender. 1996. Finding Skin in Color Images. *Proceedings of the Second International Conference Automatic Face and Gesture Recognition,* pp. 312- 317.

269. S. McKenna, S. Gong, and Y. Raja. 1998. Modelling Facial Colour and Identity with Gaussian Mixtures. *Pattern Recognition*, vol. 31, no. 12, pp. 1883-1892.

270. S. McKenna, S. Gong, and Y. Raja. 1998. Modelling Facial Colour and Identity with Gaussian Mixtures. *Pattern Recognition*, vol. 31, no. 12, pp. 1883-1892.

obust Real-time Object Detection. *International Journal of Computer Vision, IJCV 2001*.

272. H. P. Graf, T. Chen, E. Petajan, and E. Cosatto. 1995. Locating Faces and Facial Parts. *Proceedings First International Workshop Automatic Face and Gesture Recognition*, pp. 41-46.

273. H. P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen, and E. Petajan. 1996. Multimodal System for Locating Heads and Faces. *Proceedings of the Second International Conference Automatic Face and Gesture Recognition*, pp. 88-93.

274. H. M. Hunke. 1994. Locating and tracking of human faces with neural networks. *Master's thesis, University of Karlsruhe*.

275. H. Rowley, S. Baluja and T. Kanade. 1998. Neural network-based face detection. *In IEEE Transactions on PAMI*, vol. 20 (1), pp. 23-38.

276. Q. Aipeng. 2010. An improved skeletonization algorithm of color cerebral vascular image based on color level set mode. *2nd International Conference on Networking and Digital Society (ICNDS),* vol.2, pp.500-503.

277. J. H. Relethford. 2000. Human skin color diversity is highest in sub-Saharan African populations. *Human biology; an international record of research,* vol. 72 (5), pp. 773–80.

278. G. Jablonski and G. C. Jablonski. 2000. The evolution of human skin coloration". *Journal of human evolution*, vol.39 (1), pp.57–106.

279. A. Kuranov, R. Lienhart, and V. Pisarevsky. 2002. An Empirical Analysis of Boosting Algorithms for Rapid Objects With an Extended Set of Haar-like Features. *Intel Technical Report MRL-TR-July02-01*.

280. J. Malik, S. Belongie, T.K. Leung and J. Shi. 2001. Contour and texture analysis for image segmentation. *International Journal Computer Vision*, vol. 43(1), pp.7–27.

281. U. Montanari. 1971. On the optimal detection of curves in noisy pictures. *Comm. Ass. Computer*, vol.14, pp.335–345.

282. P. Parent and S. Zucker. 1989. Trace inference, curvature consistency, and curve detection. *IEEE Transactions Pattern Analysis Machine Intelligent*, vol.11 (8), pp.823–839.

283. A. Sha'ashua and S. Ullman. 1988. Structural saliency: The detection of globally salient structures using a locally connected network. *Proceedings 2nd International Conference Computer Vision*, Tampa, FL, USA, pp. 321–327.

284. L. Williams and D. Jacobs. 1995. Stochastic completion fields: A neural model of illusory contour shape and salience. *Proceedings of the 5th International Conference Computer Vision*, Cambridge, MA, pp. 408–415.

285. R. C. Gonzales and R. E. Woods. 2002. *Digital Image Processing*. 2nd Ed. Upper Saddle River, N. J.: Prentice-Hall, Inc.

286. B. Jahne. 2005. *Digital Image Processing*. 6th edition, New York: Springer.

287. L. O'gorman, M. Sammon and M. Seul. 1999. *Practical Algorithms for Image Analysis: Description, Examples and Code.* edisi kebrp. USA: Cambridge University Press.

288. A. McAndrew, 2004. *Introduction to Digital Image Processing With Matlab.* USA: Thomson Course Technology, pp.353400118.

289. H. Freeman. 1967. The classification of line-drawing data. *Models for the Perception of Speech and Visual Form,* pp. 408–412.

290. Y. K. Liu, B. Zalik. 2005. An efficient chain code with Huffman coding. *Pattern Recognition*, vol.38 (4), pp.553-557.

291. E. Bribiesca. 1999. A new chain code2. *Pattern Recognition*, vol. 32, pp. 235–251.

292. H. Sánchez-Cruz, R. M. Rodríguez-Dagnino. 2005. Compressing bi-level images by means of a 3-bit chain code. *Optical Engineering, SPIE*. Vol.44 (9), pp.1-8.

293. M. M. Youssef, K. V. Asari, R. C. Tompkins and J. Foytik. 2010. Hull convexity defects features for human activity recognition. *Applied Imagery Pattern Recognition Workshop (AIPR), IEEE 39th* , pp.1-7, 13-15.

294. P. P. Roy, U. Pal, J. Llado´s, and F. Kimura. 2008. Convex hull based approach for multi oriented character recognition from graphical documents. *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Tampa, pp. 8–11.

295. K. K. Homma and E-I. Takenaka. 1985. An image processing method for feature extraction of space-occupying lesions. *Journal of Nuclear Medicine*, vol. 26, pp. 1472–1477.

296. S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distribution, and the Bayes-ian retoration of images. *IEEE Transactions Pattern Analysis Machine Intelligent*, vol. 6, pp.721–741.

297. D. Mumford and J. Shah. 1989. Optimal approximations by piecewise smooth functions, and associated variational problems. *Comm. Pure Math.*, vol.42, pp.577–684.

298. J. R. Smith and S. F. Chang. 1996. VisualSeek: A fully automated content based image query system. *ACM Multimedia Conference*, Boston.

299. A. Vadivel, S. Sural and A.K. Majumdar. 2005. Human color perception in the HSV space and its application in histogram generation for image retrieval. *International Proceedings SPIE, Color Imaging X: Processing, Hardcopy, and Applications*, vol. 5667, pp. 598–609.

300. Z. X. Guo and Z. Q. Shun. 2007. The Neural Network and The Realization of The Matlab. *Publishing House Of Electronic Industry*, Beijing, China.

301. Pulse and respiration, [Videocassette], "Vital signs Series 1", *School of Nursing, Dublin City University*, Ireland.

302. J. D. Bronzino. 1995. Principles of Electrography. *Berbari E.J. ed., The Biomedical Engineering Handbook*, CRC Press Inc., Florida, pp. 183.

303. J. G. Webster. 1995. The Origin of Biopotentials. *Clark, J.W. ed., Medical Instrumentation Application and Design*, John Wiley and Sons, Inc., U.S.A., pp. 182-183.

List of Abbreviations

A

AASS –Applied Autonomous Sensor Systems

ANN - Artificial Neural Network

α-β tracker – Alpha-Beta tracker (i.e. modified Kalman filter)

B

BVP - Blood Volume Pulse

BP network – Back Propagation network

C

CCTV - Closed Circuit Television

CMOS- Complementary Metal Oxide Semiconductor

CCD - Charge Coupled Device

D

DCU - Dublin City University

DAQ- Data acquisitions

E

EU FP6 – European Union Sixth Framework Programme

EEG - Electroencephalography (to measure the brain)

EMG - Electromyography (to measure the muscles)

ECG/EKG - Electrocardiogram (to measure the heart)

EOG - Electrooculography (to measure the eye)

EOHs - Edge Orientation Histograms

F

FIRST - Friendly Interactive Robot for Service Tasks

G

GPS - Global Positioning System

GM - Gaussian model

GMM - Mixture of Gaussian

H

HR - Heart Rate

HOG - Histogram of oriented gradients

HSCs - Human Size Constraints

HDR camera- High Dynamic Range camera

HIR – Human Robot Interface

HWs - Haar wavelets

HSV- Hue, Saturation and value (Brightness)

I

iWARD - Intelligent Robot Swarm for Attendance, Recognition, Cleaning and Delivery

IHSR - Intelligent Hospital Service Robot

IR – Infrared

IP- Internet Protocol

IST- Irish Standard Time

J

JPL - Jet Propulsion Laboratory

L

LRFs - Local Receptive Fields

M

MKR - Muratec Keio Robot

MoG - Mixture of Gaussian

MC – Mission Controller

P

PMD - Photonic Mixing Device

PCA - Principal Components Analysis

PIR – Passive Infrared

PTZ – Pan Tilt Zoom

R

RGB- RED, GREEN and BLUE

RMS - Root Mean Square

ROI - Regions of Interest

RTDs - Resistive Thermal Devices

RF - Radio Frequency

RPC- remote procedure calling

S

SIFT - Scale-Invariant Feature Transform

SVMs - Support Vector Machines

SEM - Sensor Electronics Module

T

TOF - Time-of-flight

TIR - Thermal Infrared

TTL- Transistor Transistor Logic

TDOA - Time-Difference-of-Arrival

U

USB – Universal Serial Bus

V

VS- Visible Spectrum

X

XML- Extensible Markup Language

Y

YAML - Ain't Markup Language

Z

ZCam - TOF camera products by 3DV Systems, integrating full-color video with depth information

Numerical

1D - One dimensional

2D - Two dimensional

3D -Three Dimensional

List of Software

External Library Software:

- OpenCV (Image processing)
- cURL/libcurl (Data transfer)
- XML-RPC (TCP/IP communication)

Hardware Driver Software for Linux OS (developed by the author):

- Phillips USB Camera driver
- Canon (VC-C4) Camera PAN/TILT control driver
- IP Camera communication driver
- XML-RPC client software driver for PMD 3D sensor
- Equvital sensor communication driver

Module Software (developed by the author):

- Environmental Condition Monitoring
- Situation Recognition (Intruder monitoring and recognising patients on the floor)
    1. Face detector
    2. Badge detector
    3. Image descriptor (FCC, DFCC, Convexity Defect and H-S histogram)
    4. Video Camera and PMD 3D image fusion software
    5. Decision maker (Artificial Neural Network/ANN)
- Patient Physical Condition Monitoring
- Orca Functions (Robot SWARM communication software)

Appendix A

Software and Experimental Results

Orca interface definition for the basic module

```
<?xml version="1.0" ?>
<sci xmlns="http://www.fatronik.com/sci" version="1.0">
	<header>
		<author>Simon Thiel</author>
		<date>2008-11-12</date>
		<modifications>Tamas Szecsi</modifications>
		<date>2009-05-19</date>
	</header>
	<definitions type="slice">
 <![CDATA[
    #include <orca/orca.ice>

module iward
{
 enum ModuleErrortype {MODULEHARDWARE, MODULEFILEIO, MODULESOFTWARE,
MODULEOTHER};
 enum ModuleActionRequestType {MODULESHUTDOWN, MODULERESET, MODULECONTINUE,
MODULERETRY};
 enum ModuleHWtype {MODULEPORT, MODULESENSOR, MODULECAMERA,
MODULEAUXDEVICE};
 enum ModuleFILEIOtype {MODULEFILEOPEN, MODULEFILECLOSE, MODULEFILEAPPEND,
MODULEFILEDELETE};

 class ModuleErrorEvent  extends orca::OrcaObject
 {
  ModuleErrortype failType;
  string errormessage;
  ModuleActionRequestType actionRequest;
 };

 class HardwareModuleErrorEvent extends ModuleErrorEvent
 {
  ModuleHWtype HWFailType;
  string HWname;
 };

 class FileIOModuleErrorEvent extends ModuleErrorEvent
 {
  ModuleFILEIOtype fileiofailtype;
  string filename;
 };

 class SoftwareModuleErrorEvent extends ModuleErrorEvent
 {
  string functionname; //Example: "CalcConeDistance()"
 };

};

 ]]>
	</definitions>

	<interface name="BasicModule" module="iward" lockable="true">
		<doc>
			This interface must be implemented by all components providing access to a
hardware module.
```

The interface provides access to basic behaviour that is required for all modules.

The subscription will be raised, in case a module dependent error occurs.

```
		</doc>
		<subscription type="ModuleErrorEvent" />
		<operation name="shutdown">
			<doc>
				the module is ordered to shut down
			</doc>
			<return type="int"/>
	</operation>

		<operation name="reset">
			<doc>
				reset the module
			</doc>
			<return type="int"/>
	</operation>
	</interface>
</sci>
```

Orca interface definition for the Environmental condition monitoring module

```
<?xml version="1.0" ?>
<sci xmlns="http://www.fatronik.com/sci" version="1.0">
        <header>
                <author>Simon Thiel</author>
                <date>2008-12-17</date>
                <modified>Tamas Szecsi</modified>
                <date>2009-06-03</date>
        </header>
        <definitions type="slice">
 <![CDATA[
    #include <orca/orca.ice>

module iward
{
 enum EnvMonEvents {
  ENVMONNOEVENT, ENVMONSMOKE,
  ENVMONHIGHTEMPERATURE, ENVMONLOWTEMPERATURE,
  ENVMONHIGHHUMIDITY, ENVMONLOWHUMIDITY
 };

struct EnvMonNormData {
   string location;
   float normalmintemp;
   float normalmaxtemp;
   float normalminhumid;
   float normalmaxhumid;
   float measuredtemp;
   float measuredhumid;
   string measuretime;
   bool isCorrect;
};

 class EnvMonData  extends orca::OrcaObject
 {
  EnvMonEvents event;
  string location;
  float normalmintemp;
  float normalmaxtemp;
  float normalminhumid;
  float normalmaxhumid;
  float measuredtemp;
  float measuredhumid;
  string measuretime;
  bool isCorrect;
 };

};
 ]]>
        </definitions>
         <interface name="MIEnvSensor" module="iward" lockable="true">
                <doc>
                        Provides access to the environmental sensors module.

                        Once activated the module updates the shared model with sensor data
```

```
                </doc>

        <subscription type="EnvMonData" />

        <operation name="activate">
                        <doc>
                         This commands activates the sensors in the environmental sensor module
                        </doc>
         </operation>
         <operation name="deactivate">
                        <doc>
                         This commands deactivates the sensors in the environmental sensor module
                        </doc>
         </operation>

        <operation name="getLastEnvData">
                        <doc>
                                This command returns the last measured temperature/humidity/smoke
values
                        </doc>
                    <return type="EnvMonNormData"/>
         </operation>


       </interface>
</sci>
```

Orca interface definition for the situation recognition module

```
<?xml version="1.0" ?>
<sci xmlns="http://www.fatronik.com/sci" version="1.0">
	<header>
		<author>Simon Thiel</author>
		<date>2009-02-10</date>
		<modified>Tamas Szecsi</modified>
		<date>2009-06-10</date>
	</header>
	<definitions type="slice">
 <![CDATA[
    #include <orca/orca.ice>

		module iward
		{
			enum SituationDetectedType {
				SITRECINTRUDERDETECTED, SITRECEMERGENCYDETECTED
			};

			class SituationDetectedEvent extends orca::OrcaObject
			{
				SituationDetectedType type;
				string eventID;
				string eventLink;
				double probability;

			};
		};
 ]]>
	</definitions>
	<interface name="SituationRecognition" module="iward" lockable="true">
		<doc>
			Provides access to the situation recognition component


		</doc>
	<subscription type="SituationDetectedEvent" />


	<operation name="startPatientRecognition">
			<doc>
			 This command activates the patient recognition part of the situation recognition
component
			</doc>
	</operation>

	<operation name="stopPatientRecognition">
			<doc>
			 This command deactivates the patient recognition part of the situation recognition
component
			</doc>
	</operation>

	<operation name="startIntruderRecognition">
			<doc>
			 This command activates the intruder recognition part of the situation recognition
component
```

```
                </doc>
        </operation>

        <operation name="stopIntruderRecognition">
                <doc>
                 This command deactivates the intruder recognition part of the situation recognition
component
                </doc>
        </operation>

                <operation name="startSampleRecorder">
                <doc>
                 This command activates the SampleRecorder of the situation recognition component
                </doc>
        </operation>

                <operation name="stopSampleRecorder">
                <doc>
                 This command deactivates the SampleRecorder of the situation recognition
component
                </doc>
        </operation>

           <operation name="fireIntruderDetEvent">
                <doc>
                 This command fires an Intruder Detection Event
                </doc>
        </operation>

                <operation name="firePatientRecEvent">
                <doc>
                 This command fires an Patient Recognition Event
                </doc>
        </operation>

        <operation name="enableSimulationMode">
                <doc>
                 This command fires an Patient Recognition Event
                </doc>
        </operation>

        <operation name="disableSimulationMode">
                <doc>
                 This command fires an Patient Recognition Event
                </doc>
        </operation>

        <operation name="enableControlCamMode">
                <doc>
                 This command fires an Patient Recognition Event
                </doc>
        </operation>

        <operation name="disableControlCamMode">
                <doc>
                 This command fires an Patient Recognition Event
                </doc>
        </operation>

        <operation name="toggleDummyImage">
```

```
                    <doc>
                     Toggle the dummy image: blank > patient > stuff > intruder
                    </doc>
        </operation>


        </interface>
</sci>
```

Situation recognition (Scenario: intruder monitoring) module Log file

Intruder Monitoring Parameters Obtained. Time: Mon Jun 01 11:21:40 2009

Camera URL:

or Local Camera Image (Phillips Wbcam Image)

Frequently updated image file: img_iward_cam.jpg

Frontal face classifier filename: haarcascade_frontalface_alt2.xml

Intruder image file base name: Intruder

Intruder Monitoring Log File: IntMon_Logfile.txt

Logfile Max Length: 10000 characters

Operation Mode: 0 (0:LIMITED; 1:FULL_ORCA)

Print Mode: 1 (1: print to display; 0: do NOT print)

Image Display Mode: 0 (1: display; 0: do NOT display)

Error Log Mode: 1 (1: log errors to local file; 0: do NOT log)

Process Log Mode: 1 (1: log processes to local file; 0: do NOT log)

Event Log Mode: 1 (1: log events to local file; 0: do NOT log)

Upper body proportions: Width: 4.000000 Height: 4.000000 (multiples of face radius)

Min. number of coinciding contours for badge: 3

Max. distance between coinciding contours: 5.000000 pixels

Max. number of contours in an image: 200

Intensity-Threshold: Intt:40 Th1:80 Int2:90 Th2:120 Int3:150 Th3:200

Badge search period after face found: 8 (sec)

Wait period after badge detected, before new detection: 5 (sec)

Global Variables Initialised. Time: Mon Jun 01 11:21:40 2009

Intruder Monitoring Module Activated. Time: Mon Jun 01 11:21:40 2009

Person found, start searching badge. Time: Mon Jun 01 11:21:40 2009

Event: Intruder Detected Time: Mon Jun 01 11:21:49 2009

Image file with detected INTRUDER: Time: Mon Jun 01 11:21:49 2009

IntruderMon_Jun_01_11_21_49_2009.jpg Time: Mon Jun 01 11:21:49 2009

Intruder Monitoring Module Deactivated. Time: Mon Jun 01 11:21:49 2009

Artificial Neural Network (ANN) PARAMETERS.TXT file.

```
/* ---------------------- PARAMETERS.TXT ------------------------- */
/* -------Image descriptors Data file for ANN system ------------ */
/*--------------------Situation Recognition.-------------------  */
/* ------------------------------------------------------------- */
/* --------------------- Common Processing Parameters --------- */
INPUTS             37
IMAGE_SAMPLES      3000
/* ----------------- Input/Output Intervals -------------------- */
INPUT_0  0   1
INPUT_1  0   1
INPUT_2  0   1
INPUT_3  0   1
INPUT_4  0   1
INPUT_5  0   1
INPUT_6  0   1
INPUT_7  0   1
INPUT_8  0   1
INPUT_9  0   1
INPUT_10 0   1
INPUT_11 0   1
INPUT_12 0   1
INPUT_13 0   1
INPUT_14 0   1
INPUT_15 0   1
INPUT_16 0   1
INPUT_17 0   1
INPUT_18 0   1
INPUT_19 0   1
INPUT_20 0   1
INPUT_21 0   1
INPUT_22 0   1
INPUT_23 0   1
INPUT_24 0   1
INPUT_25 0   1
INPUT_26 0   1
INPUT_27 0   1
INPUT_28 0   1
INPUT_29 0   1
INPUT_30 0   1
INPUT_31 0   1
INPUT_32 0   1
INPUT_33 0   1
INPUT_34 0   1
INPUT_35 0   1
INPUT_36 0   1
OUTPUT   0   1
/* --------------------- Neural Network Parameters ------------ */
HIDDEN_NODES       37
MAX_TOTAL_ERROR    0.005
LEARNING_RATE      0.2
MOMENTUM_TERM      0.2
RANDOM_RATE        0.1
ITER_RESULT_NN     1
ITER_STOP_NN      1000000
```

```
/* ------- Descriptors (FCC-8, DFCC-17, CD-3 and HS-9)----------------- */
/* ----- of the Training samples; Last digit '1' = positive and '0'for
negative---- */
0.214286 0.000000 0.214286 0.142857 0.071429 0.071429 0.214286 0.071429
0.600000 0.200000 0.000000 0.200000 0.000000 0.000000 0.400000 1.000000
1.000000 0.333333 0.000000 0.333333 0.000000 0.000000 1.000000 0.200000
0.400000 0.761905 0.000000 0.000000 0.095238 0.000000 0.000000 0.000000
0.000000 0.047619 0.047619 0.000000 0.047619 1

0.375000 0.000000 0.062500 0.125000 0.250000 0.062500 0.062500 0.062500
0.454545 0.272727 0.000000 0.272727 0.000000 0.000000 0.454545 1.000000
1.000000 0.600000 0.000000 0.600000 0.000000 0.000000 1.000000 0.272727
0.454545 0.800000 0.050000 0.050000 0.050000 0.000000 0.050000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 1

0.272727 0.121212 0.060606 0.030303 0.303030 0.121212 0.030303 0.060606
0.647059 0.058824 0.000000 0.294118 0.000000 0.000000 0.941176 1.000000
1.000000 0.090909 0.000000 0.454545 0.000000 0.000000 0.812500 0.312500
1.000000 0.823529 0.058824 0.000000 0.000000 0.058824 0.000000 0.000000
0.058824 0.000000 0.000000 0.000000 0.000000 0

.
.
.
.

0.166667 0.000000 0.333333 0.000000 0.166667 0.000000 0.333333 0.000000
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 0.500000 1.000000
0.000000 1.000000 0.000000 0.000000 0.000000 0.000000 1.000000 0.000000
0.250000 0.750000 0.187500 0.062500 0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000 0.000000 1
```

Table A-1: ANN results of recalling the test samples.

```
/*------------------------------RESULTS.TXT -------------------------*/
/*-----Results of Recalling samples (Total samples 875)-------------*/
Ex[0]:  Desired Output=0.000000  Calc.OUTPUT=0.000000  Diff=0.000000
Ex[1]:  Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[2]:  Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[3]:  Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[4]:  Desired Output=0.000000  Calc.OUTPUT=0.000000  Diff=0.000000
Ex[5]:  Desired Output=0.000000  Calc.OUTPUT=0.000000  Diff=0.000000
Ex[6]:  Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[7]:  Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[8]:  Desired Output=1.000000  Calc.OUTPUT=0.999992  Diff=0.000008
Ex[9]:  Desired Output=0.000000  Calc.OUTPUT=0.817399  Diff=0.182601
.
.
.
.
.
.
.
Ex[872]: Desired Output=1.000000  Calc.OUTPUT=0.032862  Diff=0.967138
Ex[873]: Desired Output=1.000000  Calc.OUTPUT=0.985856  Diff=0.014144
Ex[874]: Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
Ex[875]: Desired Output=1.000000  Calc.OUTPUT=1.000000  Diff=0.000000
```

Table A-2: Equvital Sensor data conversion (Hexadecimal to ASCII) [51].

Hexadecimal	ASCII	Definition	Data Prior to Coding
65	e	Primary ECG1	10 bit raw A/D reading + sequence no.
63	c	Secondary ECG2	10 bit raw A/D reading + sequence no.
66	f	Belt Sensor Respiration	10 bit raw A/D reading
76	v	Vertical Acceleration	normalised signed 12 bit value (2's comp) in 0.01 g, ie -20.48 to +20.47 g
6C	l	Lateral Acceleration	normalised signed 12 bit value (2's comp) in 0.01 g, ie -20.48 to +20.47 g
77	w	Longitudinal Acceleration	normalised signed 12 bit value (2's comp) in 0.01 g, ie -20.48 to +20.47 g
74	t	Skin Temperature	calibrated signed 12 bit value (2's comp) in 0.1 °C, ie 0 to +204.7 °C
69	i	Impedance Respiration	10 bit raw A/D reading
	r	R-R Interval	9 bit value in 1/256 s units i.e. 0.000 to 1.996 (sec + sequence number)
75	u	Belt Respiration Rate	12 bit value in 0.1 bpm (0.0 to 409.5 bpm)
68	h	Heart Rate	12 bit value in 0.1 bpm (0.0 to 409.5 bpm)
62	b	R-wave Derived Heart Rate	12 bit value in 0.1 bpm (0.0 to 409.5 bpm)
	j	Impedance Respiration Rate	12 bit value in 0.1 bpm (0.0 to 409.5 bpm)
	d	EDR Respiration Rate	12 bit value in 0.1 bpm (0.0 to 409.5 bpm)
6E	n	Indication	
6F	o	Indication Clear	
	z	Report Configuration or Calibration Value	
7B	{	Sensor ID Part 1	ls 12 bits
7D	}	Sensor ID Part 2	ms 12 bits
	s	Sleep Mode	
	x	Battery Voltage	12 bit value in millivolts
71	q	Fault	
61	a	ECG Signal Quality	0-100%
67	g	Heart Rate Confidence	0-100%
6B	k	Belt Respiration Signal Quality	0-100%
	m	EDR Signal Quality	0-100%
70	p	Breathing Rate Confidence	0-100%
79	y	R-wave Signal Quality	0-100%
7E	~	Impedance Respiration Signal Quality	0-100%
7C	|	VSDS Algorithm Confidence	0-100%
60	‘	Heart Rate Correlation	0-100%