“Missed Knock”

A Computer Vision System for Logging Missed Visitors

A Dissertation Presented

by

JOHN MCVANN

A dissertation submitted to the University of Dublin,

in partial fulfilment of the requirements for the degree of

Master of Science in Ubiquitous Computing

2005

Declaration

I declare that this dissertation is my own work, except where otherwise accredited, and has not been previously submitted for a degree at this or any other university or institution.

_____

John McVann

May 13 2005

Declaration to Lend and/or Copy

I the undersigned agree that Trinity College may lend or copy this dissertation upon request.

_____

John McVann

May 13 2005

Abstract

"Missed Knock" is a Computer Vision system that attempts to detect Missed Visitors at a door. What the system aims to do is to detect a knocking motion on a door. If a knocking motion is detected a face image of this missed visitor is sent to a display application on the computer in the office.

When the office owner returns to their office they can then check the display application to see if they have missed any visitors while they were out. On the *display application* a picture will appear of the missed visitor and the time that the visitor called at.

The face image that is sent to the display application in the office has to be a good quality image of the missed visitor. Sending an unidentifiable image to the application defeats the purpose of the system. The image sent must be the most identifiable of the missed visitors presence in the corridor.

This report describes the development of this project and explains the Computer Vision techniques used to arrive at the working application.

Acknowledgements

I would like to thank Dr Kenneth Dawson-Howe for all his help, guidance and advice throughout the course of this project. He was always helpful and very enthusiastic in his approach to this project.

I would also like to thank my housemates John Fitzgibbon, John O'Driscoll, Alan Brosnan and Thomas McGuire for their assistance and comments throughout the development of this project.

Finally I would like to dedicate this thesis to my mother and late father who passed away during the course of this year. Without their support I would not be in the position to do this project or course.

Table of Contents

Declaration i
Declaration to Lend and/or Copy ii
Abstract iii
Acknowledgements iv
Table of Contents v
List of Figure's vii
1 Introduction 1
1.1 Project Objectives 2
1.2 Motivations 5
1.3 Project Components 6
1.4 Project Assumptions 9
2 Background Research 11
2.1 Principle Steps 11
2.2 Identifying and Tracking Movement in the Corridor 12
2.2.1 Colour Tracking 12
2.2.2 Active Contour Models 12
2.2.3 Background Subtraction 13
2.3 Noise Reduction 15
2.3.1 Binary Morphology 15
2.3.2 Closing 16
2.3.3 Opening 16
2.4 Connected Component Analysis 17
2.5 Knock detection 18
2.5.1 Template Matching 18
2.5.2 Convex Hull Construction 20
2.6 Best Face Detection 21
2.6.1 Skin Colour Examination 21
2.6.2 Edge Detection 22
2.7 Communications 24
3 State of the Art 26
4 System Architecture 30
5 Image Analyses 34
5.1 Background Subtraction 34
5.2 Opening and Closing 36
5.3 Connected Component Analysis 39
5.4 Tracking 42
5.5 Knock Detection Convex Hull 44
5.5.1 Boundary Tracing 46
5.6 Head Detection 49
5.7 Best Face Detection Prewitt Edge Detection 51
6 Communications 55
6.1 The Client/ Server Model 55
6.2 Sockets 56
7 User Interface 58

8 Evaluation 59
8.1 Results 59
8.2 Difficulty's Encountered 64
8.2.1 Shadows 64
8.2.2 Colour Collisions 66
8.2.3 Problem with Tracking System 67
8.4 Future Work 70
8.5 Conclusion 71
9 References 72

List of Figure's

Figure 1: Empty corridor 3
Figure 2: Person enters the corridor moving away from the camera towards office door.. 3
Figure 3: Visitor Standing at door 3
Figure 4: Visitors Knocking Motion on Door..... 3
Figure 5: Person moving away from door 4
Figure 6: Visitor moving out of corridor 4
Figure 7: GUI to display best image 5
Figure 8: Person knocking on door..... 6
Figure 9: Unrecognisable face image 7
Figure 10: A clear face image..... 7
Figure 11: Person facing the camera..... 8
Figure 12: Person walking away from camera 8
Figure 13: Communication channels 9
Figure 14: Background subtraction formula 13
Figure 15: Image subtraction 14
Figure 16: Noise after Image Subtraction..... 15
Figure 17: Closing Operation..... 16
Figure 18: The Opening Operator..... 17
Figure 19: Labelling..... 18
Figure 20: Example knocking templates..... 19
Figure 21: Shadow cast by knocking action 19
Figure 22: Convex hull of a person knocking on a door..... 20
Figure 23: Example of skin colour detection in corridor..... 22
Figure 24: Masks used in edge detection..... 23
Figure 25: Calculates absolute magnitude of gradient..... 23
Figure 26: Communication channel between Image Processing and Office Computer ... 24
Figure 27: Overview of system architecture..... 31
Figure 28: Overview of Image Processing Architecture..... 33
Figure 29: Background Subtraction 35
Figure 30: Closing Operator 37
Figure 31: Opening Operator..... 38
Figure 32: Result of Labelling 40
Figure 33: Different direction state that a person can be in..... 43
Figure 34: Convex hull detection..... 45
Figure 35: Boundary tracing 47
Figure 36: Four possible boundary search directions 48
Figure 37: Pixel neighbour search sequence..... 48
Figure 38: Before Head detection 50
Figure 39: After head detection 50
Figure 40: Calculating head height..... 51
Figure 41: Face image mostly horizontal edges 52
Figure 42: Prewitt edge detection 53
Figure 43: Example of Prewitt edge detection on a blurred image..... 54
Figure 44: Client / Server Model 56

Figure 45: Output that the User will see in the office 58
Figure 46: Empty corridor 60
Figure 47: Person enters corridor and is moving up the corridor 60
Figure 48: Person standing at the door 61
Figure 49: Person knocking on an office door 61
Figure 50: Person Walking away from office door and down the corridor 62
Figure 51: Person leaves the corridor. Processing stops, as corridor is empty 63
Figure 52: Display GUI that appears on the office computer 64
Figure 53: Example of shadows problems 65
Figure 54: Shadow removal 65
Figure 55: Colour collision 67
Figure 56: Center point movement due to shadows 68
Figure 57: Row Movement in corridor 69

Chapter 1

1 Introduction

"*Computer vision as a field is an intellectual frontier. Like any frontier, it is exciting and disorganized, there is often no reliable authority to appeal to, many useful ideas have no theoretical ground and some theories are useless in practice. Developed area are widely scattered and one often looks completely inaccessible from the other"* [1]

Computer Vision is a sub-field of Artificial Intelligence. The purpose of Computer Vision is to program a computer to "understand" a scene or features in the scene. It attempts to replicate human vision. This is an extremely difficult field in computing due to the lack of understanding of how humans actually process images.

Until recently, active vision systems could only be achieved by making use of large, expensive and complex systems. Recent advances in technology, particularly in processor speeds, vision toolkits, micro-controllers and cameras have opened up a new area of study in Computer Vision. The range of applications and situations that Computer Vision can be applied to is constantly growing. Vision systems are becoming increasingly important to solve problems in a variety of different areas, such as manufacturing, surveillance, robotics and bio medics. Each Vision project is very specific to each problem it attempts to solve.

Computer Vision enables us to make assumptions about the real world based on single pixel values. It then combines these values into a single finding. The fact that it only requires a camera, computer hardware and software is the main advantage of Computer Vision. It is also unobtrusive, inexpensive and will work in existing unmodified environments.

1.1 Project Objectives

Computer Vision is a long way from being a complete solution but successes have been achieved in simple worlds. This project is an example of one such success. Nowadays there are many methods that notify people that someone tried to contact them but failed. A missed call list on a mobile phone is one example. This can be a very useful piece of information. A person knows that someone tried to get in contact with them and they can now reply to the missed call should they so wish.

This project attempts to develop a similar system for detecting missed visitors to an office door. When a person knocks on an office door and they are unable to gain entry, it is assumed that there is no one in the office and a missed visit has occurred. The office occupant will be notified of this "Missed Visitor". They will receive an image of the visitor, as well as the time that the visitor called. The project will apply Computer Vision techniques to enable intelligent video processing techniques to a camera in the corridor. This video processing will first attempt to detect a knocking motion on the door. If a knocking motion is detected the system will then forward a still picture of the missed visitor to the computer in the office.

The system will receive video images from a fixed camera in a corridor. This will not be aimed at one office door but will be similar to security cameras that are currently present in corridors throughout the college. This means that one camera can be used to monitor many doors in one corridor. The system will only start to process the video stream when there is motion in the corridor.

Below is an example of the input that the system will receive from the camera. In Figure 1 the corridor is empty. The visitor then enters the

corridor and begins to move away from the camera towards the office door in Figure 2.

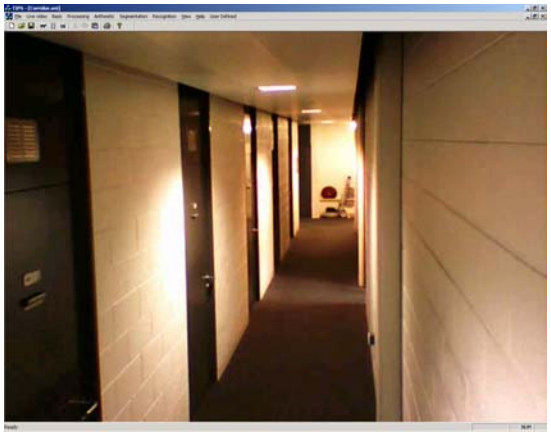


Figure 1: Empty corridor

Figure 2: Person enters the corridor moving away from the camera towards office door

In Figure 3 we can see that the visitor has stopped at an office door. In Figure 4 there is a knocking motion on the door. This is the action that will have to be detected in order to determine if a missed visit has occurred.

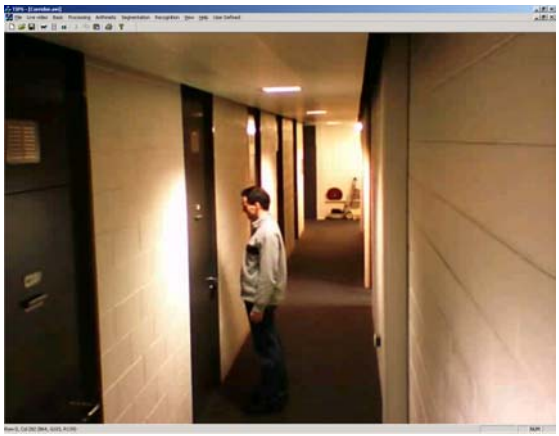


Figure 3: Visitor Standing at door

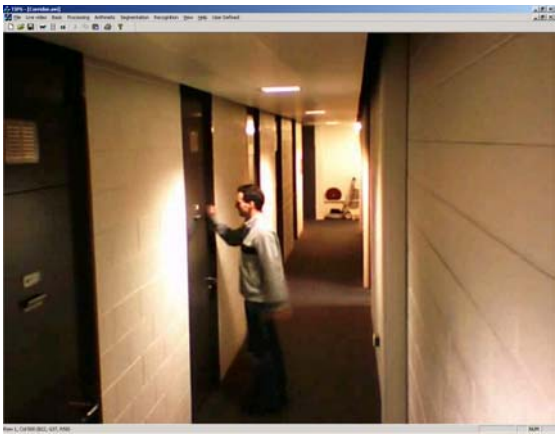


Figure 4: Visitors Knocking Motion on Door

There is no reply to the knock so the person then leaves the corridor returning in the direction from where they came.

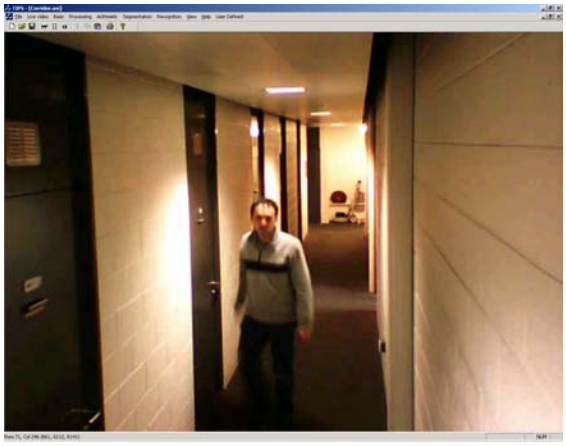


Figure 5: Person moving away from door

Figure 6: Visitor moving out of corridor

The objective of this project is to detect the knocking action that is present in Figure 4 and then attempt to capture a "*best face image*" of the person that just knocked. A *best face image* is the clearest image of a person while they are present in the corridor. Every person that enters the corridor will be tracked and the *best face image* of the person will be stored. If a knock is detected the *best face image* will be sent to a display application running on the computer in the office. The application must have the ability to track the person's movement in the corridor in order to identify which door the person knocked on. The *knock detection* and the *best face* detection will take place in the Image Processing Application.

In the office there will be a display application on the computer to display the *best face image* of the missed visitor along with the time that the visitor called. The display application in the office must have the ability to store all missed visitors. Figure 7 is an example of the final output that is expected on the office computer.

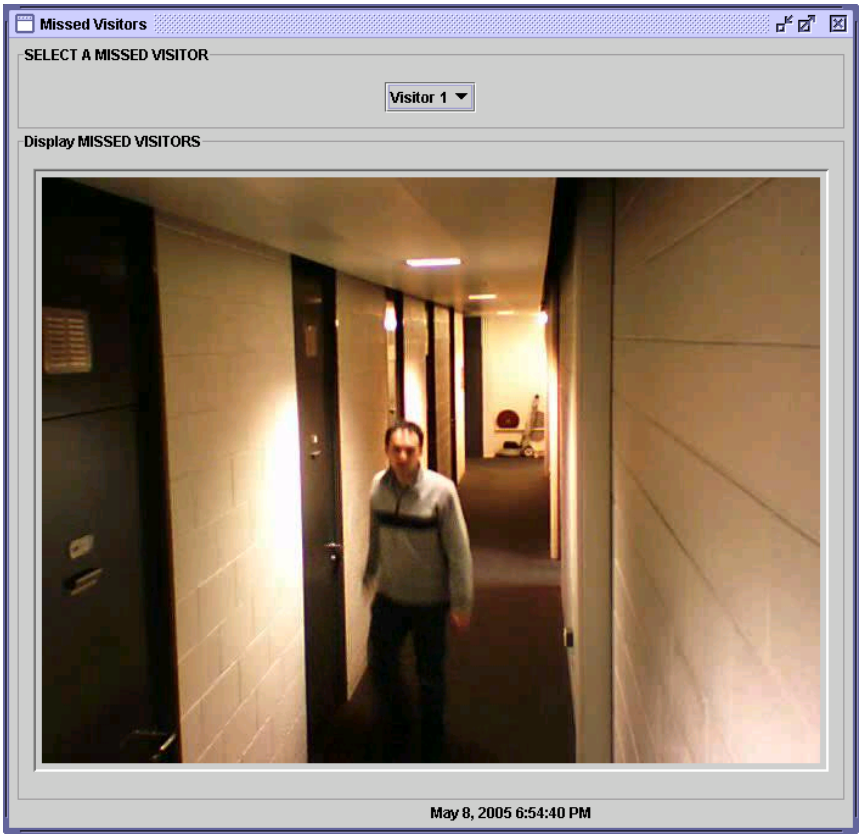


Figure 7: GUI to display best image

1.2 Motivations

Human Motion Analysis is receiving increased attention from Computer Vision researchers today. The ability to monitor human activities in areas such as city centres, airports and car parks are examples of areas that are receiving a large amount of research. Using a camera, computer hardware and software to process this information is a very cheap and efficient method of monitoring these activities.

One of the main motivations for this project is that, a system that automatically generates a picture of missed visitors to an office has not been done previously. It was felt that this would be an interesting and challenging project to attempt.

Another motivation for this project is that the infrastructure required for this system is already in place in most institutions. This can be seen in this college with the presence of cameras in the majority of corridors.

This system could also be used for security purposes. It could easily be expanded to log all entries and exits through a doorway. If anything is stolen or damaged it would be very simple to identify the culprit. This project could also be extended to track movement of a person throughout a building. This could be useful for security purposes also.

1.3 Project Components

The Project Components are described in this section. Attempting to identify if a person knocks on a door is no trivial task. Figure 8 shows an image of a person knocking on a door. This is the action that will have to be identified in order to determine if a missed visit has occurred.

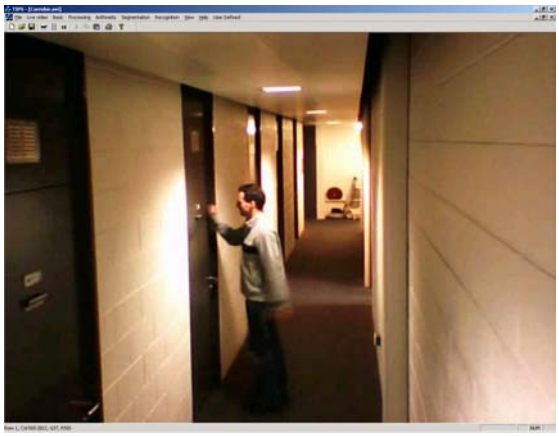


Figure 8: Person knocking on door

Another project component is to capture the *best face image* of a person in the corridor. The reason for trying to capture the *best face image* is to allow the office occupant every chance of identifying the

missed visitor. Due to the quality of the camera used, some images of the visitor are of a very poor quality. Figure 9 and Figure 10 are examples of two such contrasting images. The person in Figure 9 is virtually unrecognisable compared to the clear face image in Figure 10. For this reason a person in the corridor will have to be constantly monitored to capture a *best face image*.

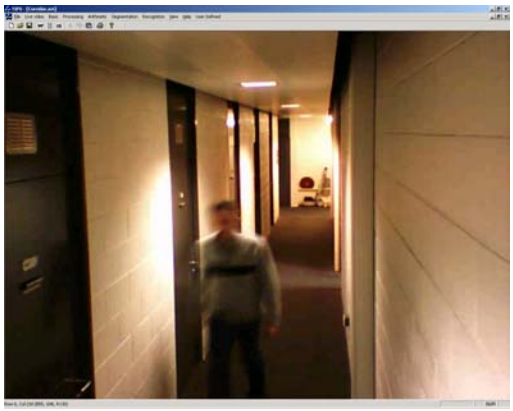


Figure 9: Unrecognisable face image

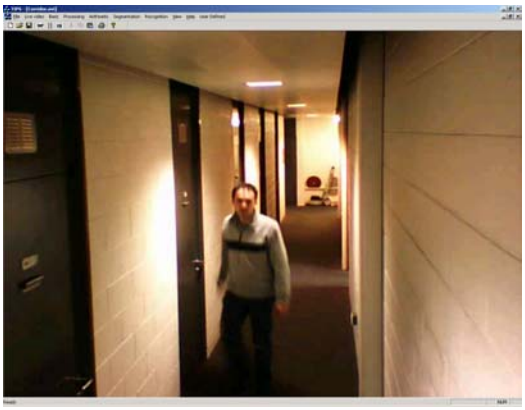


Figure 10: A clear face image

Tracking of the person in the corridor is also an important component of this project for several reasons. Firstly a person has to be tracked in order to determine which direction they were moving. Direction is important for grabbing a best image of the person in the corridor. In Figure 11, a person is moving towards the camera and in Figure 12, a person is moving away from the camera. It is obvious from these illustrations that an image where the person is moving away from the camera is worthless when trying to identify a missed visitor. Consequently, when a person is moving away from a camera these images will not be processed for a *best face image*.

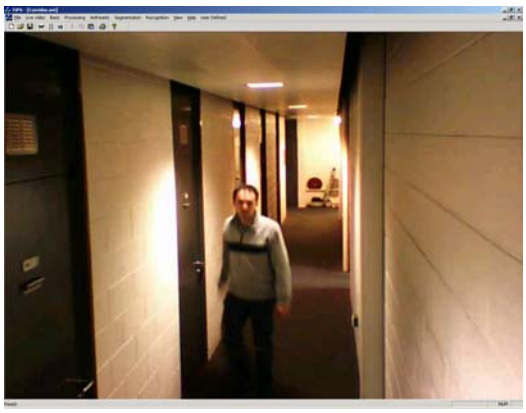


Figure 11: Person facing the camera

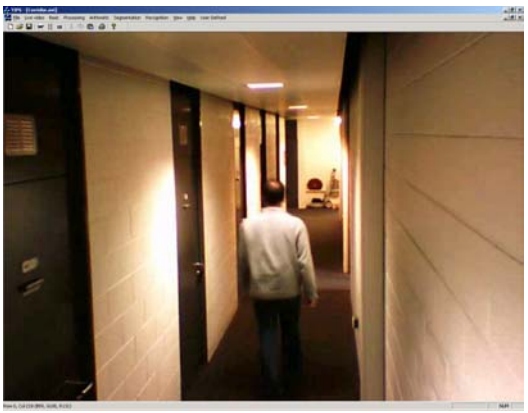


Figure 12: Person walking away from camera

Tracking is also important to identify which door the person knocks on, to enable that the *best face image* can be sent to the correct office. All of the above will take place in the Image Processing Application.

Another component that is central to the workings of this project is the communications channels. The camera used will be an IP camera. The Image Processing Application that will detect the *best face image* and the knocking gesture will connect to the camera in the corridor and receive images directly from it. The communication channel between the Image Processing Application and the display application running on the office computer is also very important. In the diagram below you can see the different areas where the communications channels are required to complete a working system. The communications channels are marked communication Channel A and communication Channel B.

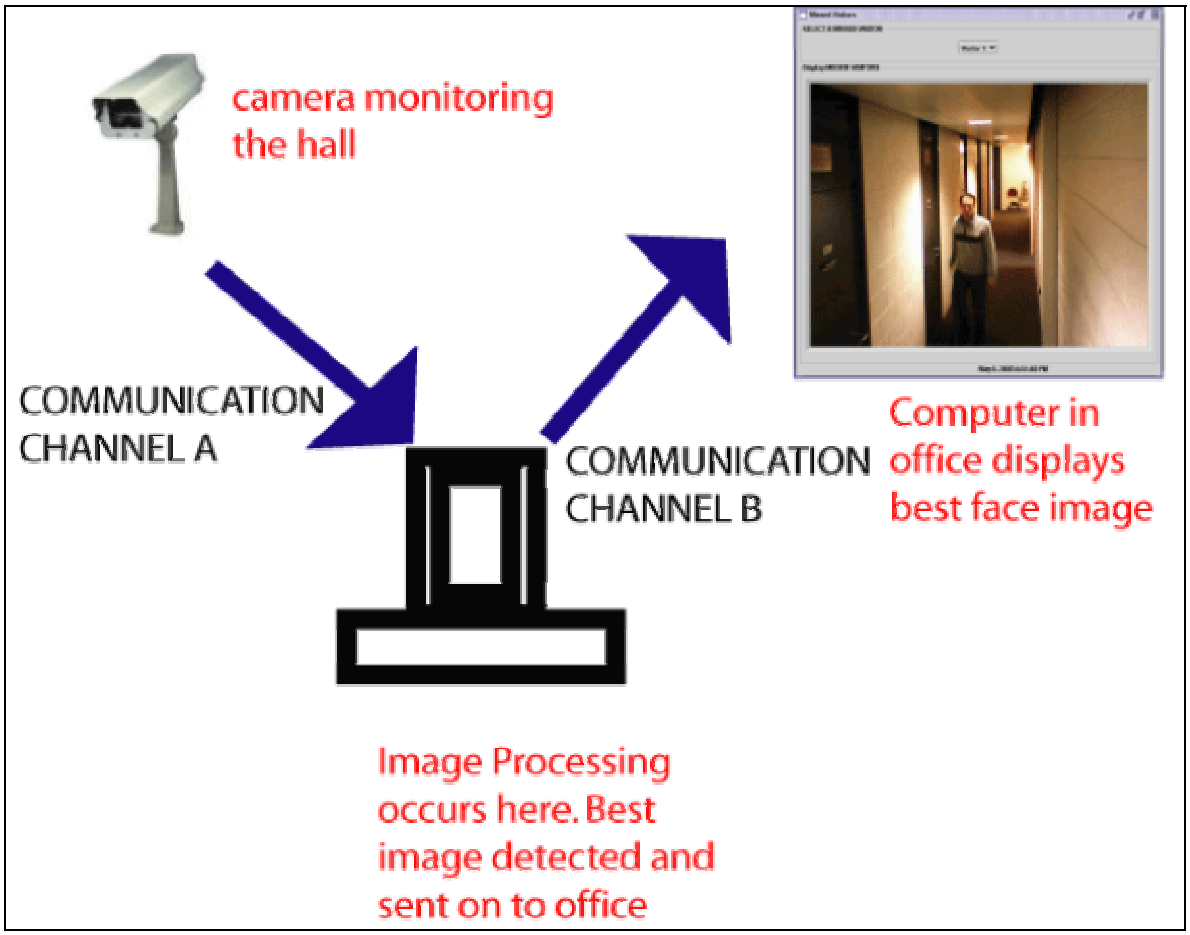


Figure 13: Communication channels

A display application will also have to be developed on the office computer to accept images from the Image Processing Application and to display these *best face images*.

1.4 Project Assumptions

The Trinity Image Processing System (TIPS) is the platform that is used in this project for the Image Analysis. TIPS is a system written in C++ and is developed by Kenneth Dawson-Howe. It is extended to be able to detect a knocking motion and to get a *best face image* of a person in a corridor.

The following assumptions are also made.

- The input is from a static camera without zoom.
- A background image of the corridor is available.
- Lighting in the corridor does not change drastically, i.e. lights in the corridor are not turned off during processing.
- There is only one person in the corridor at any given time.

Chapter 2

2 Background Research

In the background research into this project, it was discovered that no such system exists to detect a missed visitor at an office door. In this Chapter certain problems that were highlighted during background research are discussed. Different methods that could have been used to overcome these problems are then examined.

2.1 Principle Steps

In order for this project to succeed, the three main problem areas of *knock detection*, best image detection, and communication channels had to be divided up into smaller and more specific tasks. A lot of pre-processing had to occur before these tasks were fulfilled.

The principle steps required to complete the project are listed below:

1. Identify moving pixels in the corridor
2. *Noise reduction* is required as some pixels identified as moving pixels are actually background pixels
3. Grouping of moving pixels together and then identifying if the grouped pixel regions are in fact a person.
4. Track the movement of this person up and down the corridor.
5. Capture the best image of a person in the corridor.
6. Identify a knocking movement in the corridor.
7. Identify which door is knocked on.
8. Develop the communication channel that the best image will be sent over.

Possible solutions to these problems are discussed below.

2.2 Identifying and Tracking Movement in the Corridor

There is a wide range of object detection and tracking systems available today. A lot of research into this area has been completed. The identification of a person in a corridor is one of the most important steps in this project. Tracking is performed so that the location of the person in the corridor is known. Some possible solutions to object detection and tracking are mentioned below.

2.2.1 Colour Tracking

Colour, as an identifying feature could be used for tracking people in the corridor. The three-dimensional RGB colour space could be used to track a person. Only the location of the pixel that is most similar to the persons colour becomes the person's new position. The algorithm searches for the pixel most similar to the recorded colour around the previous pixel position of the person. One problem with this method is that the colour that is going to be tracked has to be initialised before the actual tracking begins. Another problem of colour tracking is the fact that it can also lock on to background colours that are the same as the colour being tracked. This made colour tracking inappropriate for use in this project.

2.2.2 Active Contour Models

Active Contours or *Snakes* are computer-generated curves that move within images to find object boundaries [2]. They are often used in Computer Vision and Image Analysis to detect and locate objects, and to describe their shape. This is basically a Complex Edge Finding Algorithm, which traces edges of objects and finds humans due to the unique shape that humans have. Active Contours contain two main parts, which are a list of control points and the energy function. Each

control point is an x, y point in the image, that when all the control points are taken together, represent the shape of the object being recognized. The energy function is a set of rules that govern, what change is possible for the location of the control points. Problems with this method are that Snakes cannot move toward objects that are too far away and that Snakes cannot move into boundary concavities or indentations (such as the top of the character U). A person knocking on a door will form one of these concavities and so it was decided not to use Active Contours as a method of identification and tracking.

2.2.3 Background Subtraction

The most straightforward approach to motion detection is Background or Image Subtraction [3]. The Current frame C in the video input sequence is subtracted from the background image R (an image of the empty corridor) yielding the difference image D,

$$D = \left|R_R - R_C\right| + \left|G_R - G_C\right| + \left|B_R - B_C\right|$$

Figure 14: Background subtraction formula

R, G and B denote the red, green and blue components of the current frame and the Background frame, respectively. If the difference D in each pixel, between the current frame and the background frame is greater than a set threshold, the pixel is considered to be a moving pixel. This is the method that was decided upon in this project.

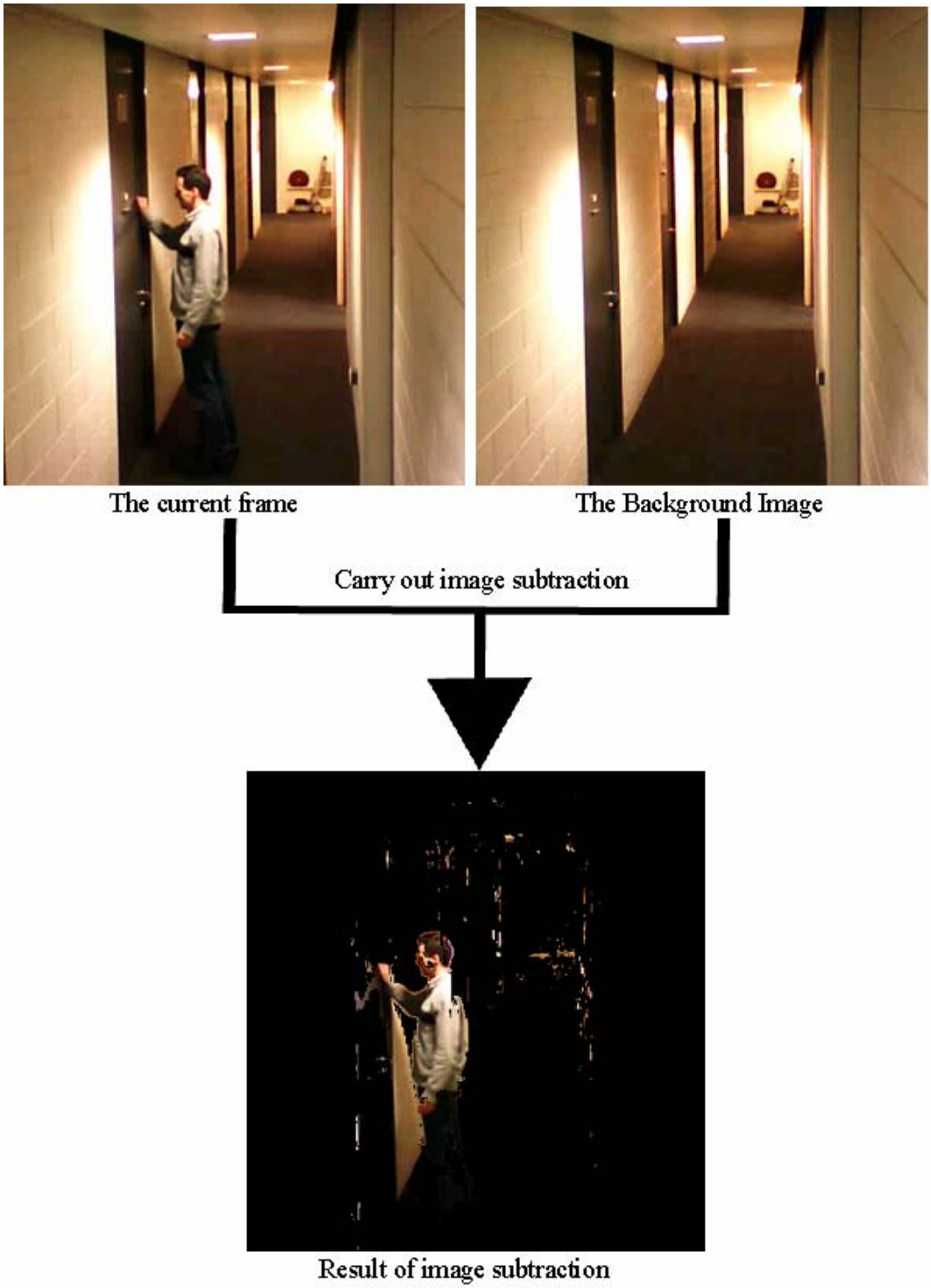


Figure 15: Image subtraction

2.3 Noise Reduction

After Background Subtraction has occurred there will inevitably be pixels in the resulting image that show up as moving pixels but are not. These false positives are called Noise [4]. Noise is caused by a number of factors such as compression, slight light changes and slight movements in the camera. Below you can see an example of Noise after Image Subtraction.

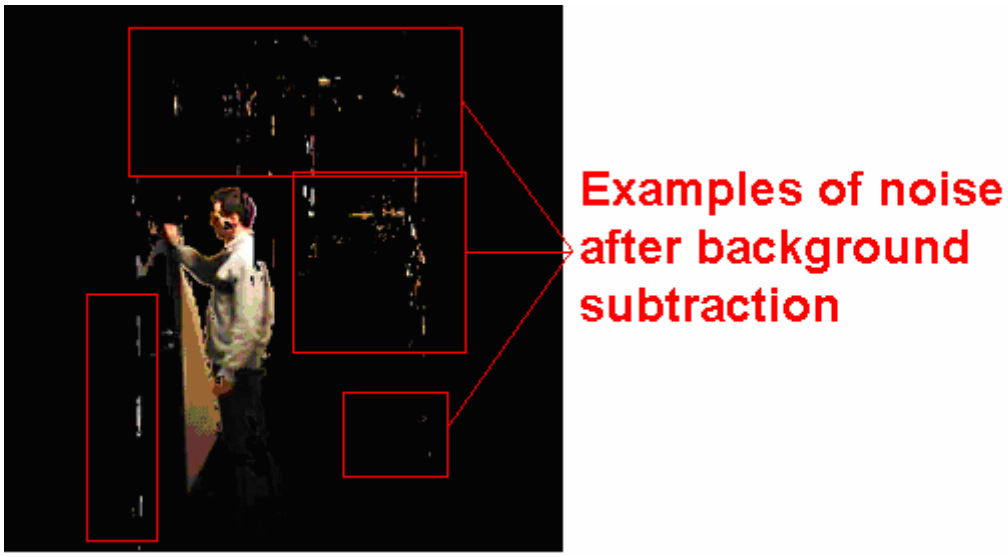


Figure 16: Noise after Image Subtraction

2.3.1 Binary Morphology

Binary Morphology is a set of fundamental operations on images [5]. The field of Mathematical Morphology contributes a wide range of operators to image processing and they are all based around a few simple mathematical concepts from set theories. The operators are particularly useful for the analysis of images and common usages include edge detection, Noise Removal, Image Enhancement and Image Segmentation. The fundamental operators of Binary Morphology are Erosion, Dilation, Closing and Opening.

2.3.2 Closing

Closing [6] takes two pieces of data as input: the image to have the Closing operator applied to it, and a Structuring Element (also known as a kernel). It is this Structuring Element that determines the precise effect the Closing has on the input image. The result of Closing is to connect pixels that are close together and to fill in small gaps that appear.

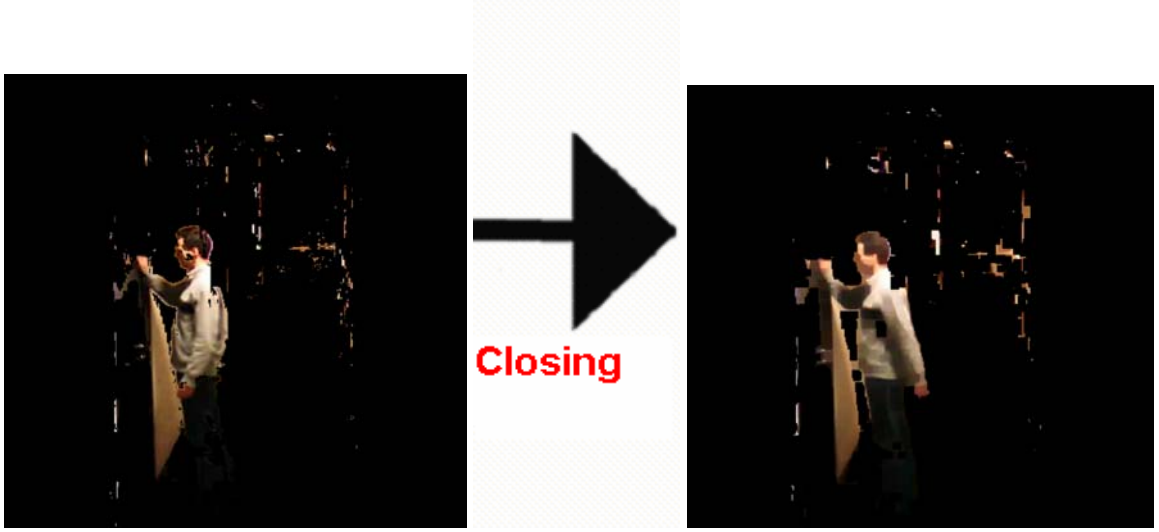


Figure 17: Closing Operation

2.3.3 Opening

The effect that an Opening [6] operation has on an image is to remove small noisy image elements. Opening also takes two pieces of data as input: an Image to be opened and a Structuring Element.

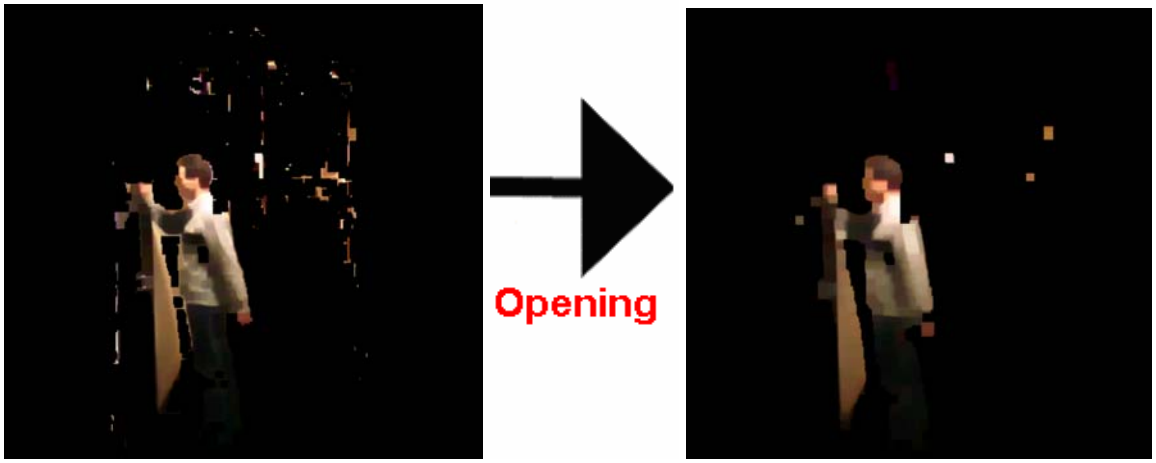


Figure 18: The Opening Operator

Both of the above Binary Morphological operations are used in this project. The Opening and Closing operations are performed on the image after Background Subtraction has occurred with Closing occurring first.

2.4 Connected Component Analysis

Connected Component Analysis [7] is central to this project. Labelling is required to determine if there is a person in the corridor, the location of the person in the corridor and the direction that the person is moving. The *knock detection* and *best face image* detection both use the labelled image to draw their conclusion. Detection of connected components between pixels in an image is a fundamental step in segmentation of an image into regions. Each region is assigned a unique label and other pixels within the same region are assigned the same label.

Below an example of labelling is shown. The image on the left is the input image before labelling occurs. The image on the right is the labelled output. In this case every colour represents a different label.

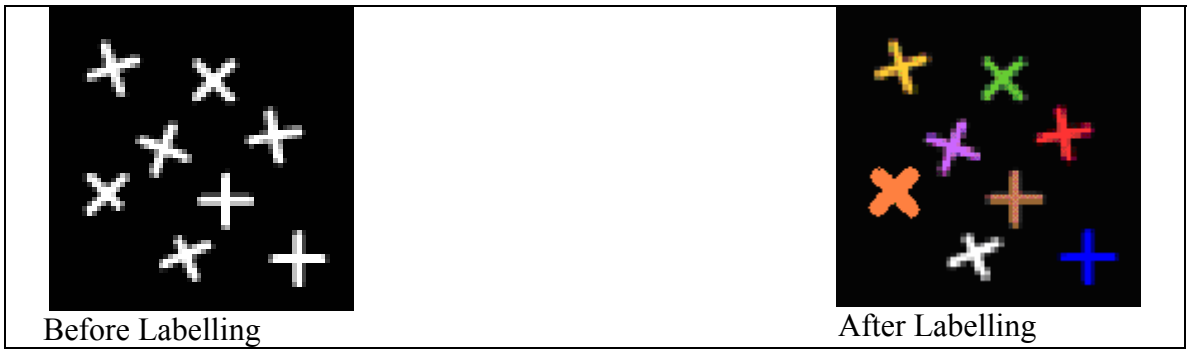


Figure 19: Labelling

Each image is searched row by row. Black pixels are considered to be background and any non-black pixels are considered to be moving and so are assigned a label. Every touching pixel in a region is assigned the same label.

2.5 Knock detection

The *knock detection* was one of the more difficult parts of this project. A lot of research was done to find the most accurate way of recognising this action. Below are some of the methods that were explored.

2.5.1 Template Matching

Template matching [8] is a technique used to isolate certain features in an image. These features can be single pixels, lines, edges or complete objects. For this project it was required to detect a knocking action. Figure 20 exhibits some of the possible templates for a person knocking on a door.

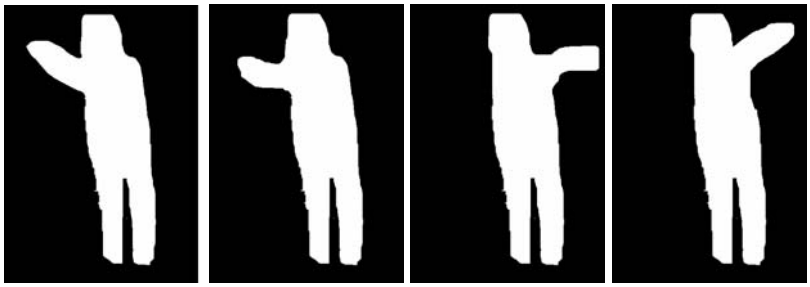


Figure 20: Example knocking templates

The problem with Template Matching is that a persons shape and scale changes dramatically through out the clip. A person that is knocking on a door near the camera is much bigger than a person that is further away. Another problem of using Template Matching is the effect that shadows have on a scene. Different shadows are created, depending on the person's position in the corridor, especially when they knock on the door. Below you can see an example of a person knocking on a door and the shadow that it casts. For these reasons, the number of templates that would be required to detect a knock would be unmanageable. Therefore to detect a knock in the corridor, Template Matching could not be used.

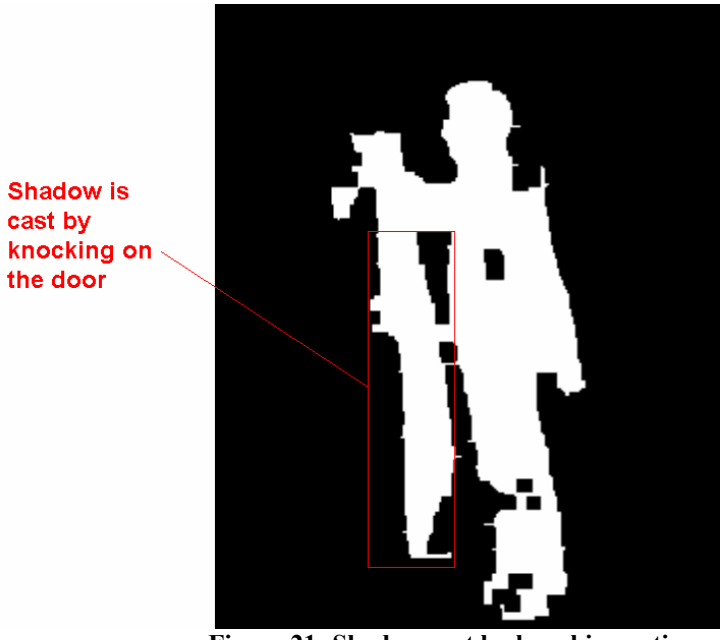


Figure 21: Shadow cast by knocking action

2.5.2 Convex Hull Construction

Formally, a Convex Hull is the smallest Convex set containing all points in an image. Informally it can be explained as if you put a rubber band around an object and every point that touches the rubber band is a vertex of the regions Convex Hull. Figure 22 is an example of how the Convex Hull on a person knocking on a door appears. The blue points in the image are the vertices of the Convex Hull.

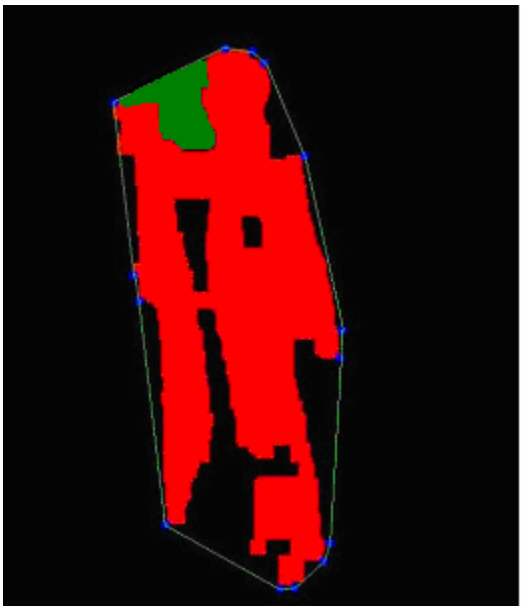


Figure 22: Convex hull of a person knocking on a door

From this image it is easy to see that a knocking motion creates a large hull close to the head region. This is the green area in Figure 22. As you can see there is a big difference in the area of the concavity on the side of the head where the person is knocking as opposed to the other side of the head. The area of the hull can be used to determine if a knock is occurring. As the diagram above shows, the presence of shadows does not affect this method. This is the method I chose to go with.

2.6 Best Face Detection

Best face detection is central to the success of this project. It is the objective of this project to enable the office occupant to recognise the missed visitor. If this is not possible from the image the objectives have not been met. There were many different possible implementation methods that could have been used to complete this task. Two that were researched are discussed below.

2.6.1 Skin Colour Examination

One method that was tested for detecting the best face in a video sequence was to use Skin Detection [8]. It was felt that by calculating the amount of skin pixels in the image, this would determine which was the best face picture. This turned out to be a false assumption. The problem with this method was that it takes each pixel one at a time, and then classifies it as either a skin pixel or not. Various pixels in the background also fall into the skin colour space. This is quite clear in Figure 23 when Skin Detection was performed on the input. As this is a bright corridor much of the background is determined to be skin pixels. Also certain pixels on the face are not classified as skin pixels due to shadows on the face or a change in illumination.

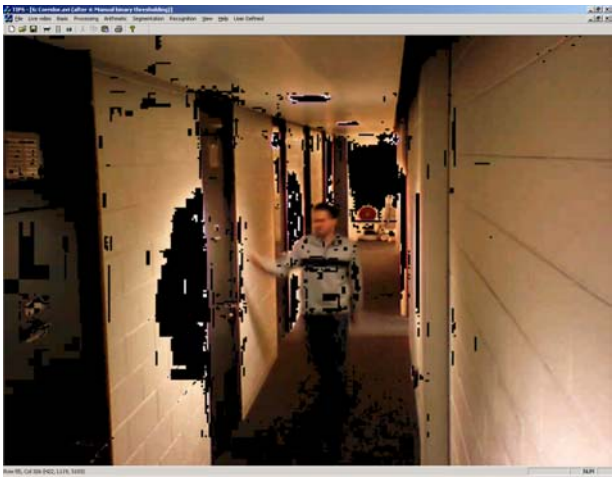


Figure 23: Example of skin colour detection in corridor

Due to the number of false positives that occurred in testing of Skin Colour Detection the decision was made not to use skin colour to capture the best face.

2.6.2 Edge Detection

The goal of Edge Detection is to mark the points in an image at which the intensity changes sharply. Sharp changes in image properties usually reflect changes in real world properties. Facial features such as eyes, nose and mouth all have very sharp edges and this factor means that Edge Detection could be used to detect the *best face*. There are many ways to perform Edge Detection. The majority of different methods may be grouped into two categories, Gradient and Laplacian. The Gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges. Prewitt edge detection is an example of a Gradient method.

Prewitt edge detection takes a greyscale image as its input. Prewitt produces an image where higher grey-level values indicate the presence of an edge between two pixels in an image. The Prewitt edge

detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). One kernel is simply the other rotated by 90°. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating the image a pixel at a time. The actual Prewitt masks are shown below in Figure 24.

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Gy *Gx*

Figure 24: Masks used in edge detection

These kernels are designed to respond primarily to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx and *Gy*). These can then be combined together to find the absolute magnitude of the gradient (That is vertical edges and horizontal edges combined) at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

Figure 25: Calculates absolute magnitude of gradient

This is the type of Edge Detection that is used in this project as it is more exact than the Skin Colour Detection method and is extremely accurate at detecting the *best face image*.

2.7 Communications

Communication is required between the Image Processing Application and the application that displays the best image on the office computer. One method for communication is the use of Sockets. Sockets are explained below.

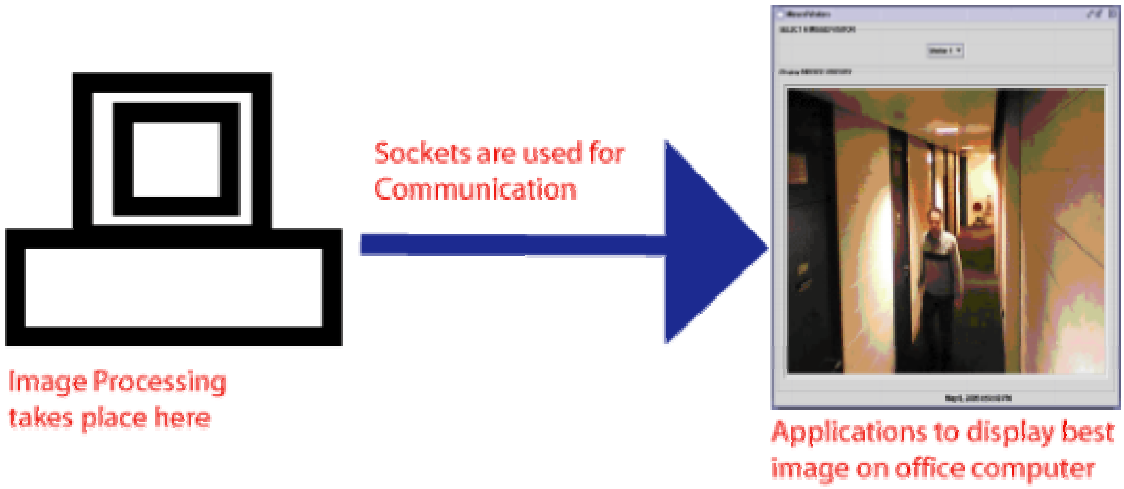


Figure 26: Communication channel between Image Processing and Office Computer

Sockets [10] provide a relatively simple mechanism for inter-process communication over network connections. Sockets allow bi-directional communication between processes. Sockets support a client-server model in that a socket server "listens" for connections and can send/receive data with "accepted" socket clients. A socket client "requests" that a connection be made with a socket server. The client must have prior knowledge of the server. The server does not need to have prior knowledge of the client.

The steps involved in establishing a socket on the client side are as follows:

- Create a socket with the `socket()` system call.
- Connect the socket to the address of the server using the `connect()` system call.

- Send and receive data. There are a number of ways to do this, but the simplest is to use the `read()` and `write()` system calls.

The steps involved in establishing a Socket on the server side are as follows:

- Create a Socket with the `socket()` system call.
- Bind the Socket to an address using the `bind()` system call. For a server socket on the Internet, an address consists of a port number on the host machine.
- Listen for connections with the `listen()` system call.
- Accept a connection with the `accept()` system call. This call typically blocks until a client connects with the server.
- Send and receive data.

There are two main types of Sockets. These are Stream Sockets and Datagram sockets. Stream Sockets use TCP (Transmission Control Protocol) for communication. Stream Sockets require that a connection be established before sending/receiving data. They guarantee delivery and ordering of packets that are sent. Datagram Sockets use UDP (Unix Datagram Protocol) for communication. They are "connection-less" and no prior connection is necessary. Datagram Sockets do not guarantee delivery or order of packet reception but are significantly faster than TCP sockets.

As images are being communicated it was decided that TCP sockets would be used. Speed was not a major factor as only still images are being transported, but it is important that every part of data that is sent is received and received in order.

Chapter 3

3 State of the Art

In this Chapter an overview of the field of Computer Vision is given and cutting edge technologies that are being currently implemented and discussed.

Computer Vision is the branch of Artificial Intelligence that focuses on providing computers with the functions typical of human vision. To date, Computer Vision has produced important applications in fields such as Industrial Automation, Robotic and Bio-medics. In the field of Industrial Automation alone, its applications include guidance for robots to correctly pick up and place manufactured parts.

Until a few years ago, chronic problems affected Computer Vision systems and prevented their widespread adoption. Since its start, Computer Vision has appeared as a computationally intensive and almost intractable field because its algorithms require a minimum of hundreds of MIPS (millions of instructions per second) to be executed in acceptable real time. Even the input–output of high-resolution images at video rate was traditionally a bottleneck for common computing platforms such as personal computers and workstations. In recent years, however, increased performance at the system level such as faster microprocessors, faster and larger memories, and faster and wider buses has made Computer Vision affordable on a wide scale.

The basic idea behind the use of Computer Vision in Human Computer Interaction (HCI) is that in several applications, computers can be instructed more naturally by human gestures than by the use of a keyboard or mouse. In one interesting application, computer scientist James L. Crowley of the National Polytechnical Institute of Grenoble in

France and his colleagues used human eye movements to scroll up and down a computer screen [11]. A camera located on top of the screen tracked the eye movements. The French researchers reported that a trained operator could complete a given task 32% faster by using his eyes rather than a keyboard or mouse to direct screen scrolling. In general, using cameras to sense human gestures, is much easier than making users wear cumbersome peripherals such as digital gloves.

Another application is 'The vOICe', developed at Philips Research Laboratories (Eindhoven, The Netherlands) by Peter B. L. Meijer [12]. The vOICe provides a simple yet effective means of augmented perception for people with partially impaired vision. In the virtual demonstration, the camera accompanies you in your wanderings. The camera periodically scans the scene in front of you and turns images into sounds, using different pitches and lengths to encode objects' position and size.

One example where a Product Quality Inspection system is implemented is at the Geoinformation Research Group, Hungary [13]. They implemented a food quality control and inspection system. 2D and 3D visual characteristics are collected about different kinds of food products that are in one way or another go through a mass production process. Characters measured for the products were size (width, length, volume, and area), shape, and color (dominant color, localized colors, average color). This study has shown that machine-based inspection of food products can be implemented effectively, reducing or even eliminating the need for both intensive human intervention and addition of conditioning chemicals to assure quality. The concepts and results presented in this study can be applied in solving more complicated pattern recognition problems.

Perhaps the most developed modern application of Computer Vision is video surveillance. Long gone are the days when video surveillance meant low-resolution, black-and-white, analog closed-circuit television. Nowadays, Computer Vision enables the integration of views from many cameras into a single, consistent "superimage." Such an image automatically detects scenes with people or vehicles or other targets of interest, classifies them in categories such as people, cars, bicycles, or buses, extracts their trajectories, recognizes limb and arm positions, and provides some form of behavior analysis. The basic goal is not to completely replace security personnel but to assist them in supervising wider areas and focusing their attention on events of interest. Although the critical issue of privacy must be addressed before society widely adopts these video surveillance systems, the recent need for increased security has made them more likely to win general acceptance. In addition, several technical countermeasures can be taken to prevent privacy abuses, such as protecting access to video footage by way of passwords and encryption.

At the University of Technology in Sydney [14], a system was developed and tested that can detect suspicious pedestrian behavior in parking lots. The approach is based on the assumption that a suspicious behavior corresponds to an individual's erratic walking trajectory. The rationale behind this assumption is that a potential offender will wander about and stop between different cars to inspect their contents, whereas normal users will maintain a more direct path of travel. The parking-lot surveillance system subtracts the static background image, distinguishes a person from moving vehicles, locates the head, and calculates the speed of the head in each frame. A neural network classifier is trained to recognize the suspicious behaviors, provides the behavior classification. In the experiments

performed, the system achieved good accuracy, with a reasonably limited number of false dismissals and false alarms—4% and 2%, respectively, among more than 100 test samples.

Chapter 4

4 System Architecture

In this chapter a high level view of how the whole system will work is described. There are three main areas of this project that had to be developed to produce a fully working system.

1. The TIPS platform had to be extended to detect the knocking motion and to determine the best image of a user in the corridor. This is the Image Processing Application.
2. A GUI to display the best image of the missed visitors to the person in the office.
3. The communications channel between the Image Processing Application and the GUI on office computer.

Figure 27 below is a diagrammatical over view of the whole system.

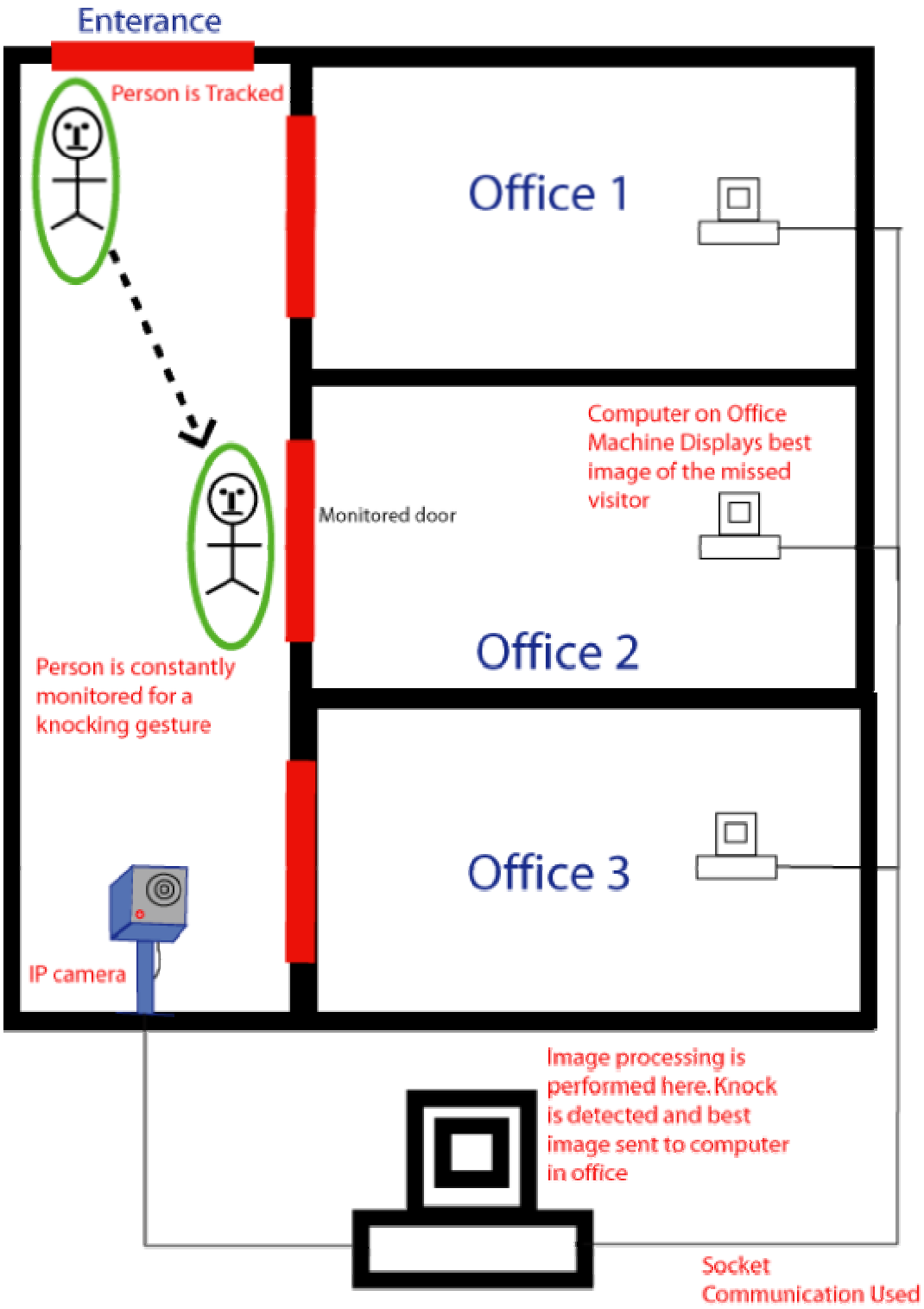


Figure 27: Overview of system architecture

Extending the TIPS platform to perform the image processing is obviously the major aspect of this project. From the diagram above the basic System Architecture is detailed. As you can see each office has a computer that has a connection to the Image Processing Application so it can receive a *best image* of the missed visitor.
The system has an IP camera monitoring the corridor. This IP camera is not pointed at any individual office door so every door in the corridor can be monitored. By using the IP addresses the TIPS platform used for the Image Processing can access live video stream directly from the camera. As mentioned above, the TIPS platform is where all the Image Processing is performed. When a person enters the corridor the system will continuously try to detect a best face of the person. At the same time the system is constantly monitoring the person for a knocking gesture. If a person knocks on a door it is noted that a knock occurred as well as which door the person knocked on and the time that the knock occurred. After the person walks away from the door that they knocked on, they are still monitored for a *best face image*. Once the person leaves the corridor the *best face image* is sent to an application running on the computer in the office via sockets communication. The TIPS platform and the application that sends the best image to the different offices will reside on the same machine.

As mentioned above the Image Analysis is the main area of this project. Figure 28 shows a diagram with all the steps required for the Image Processing to function correctly.

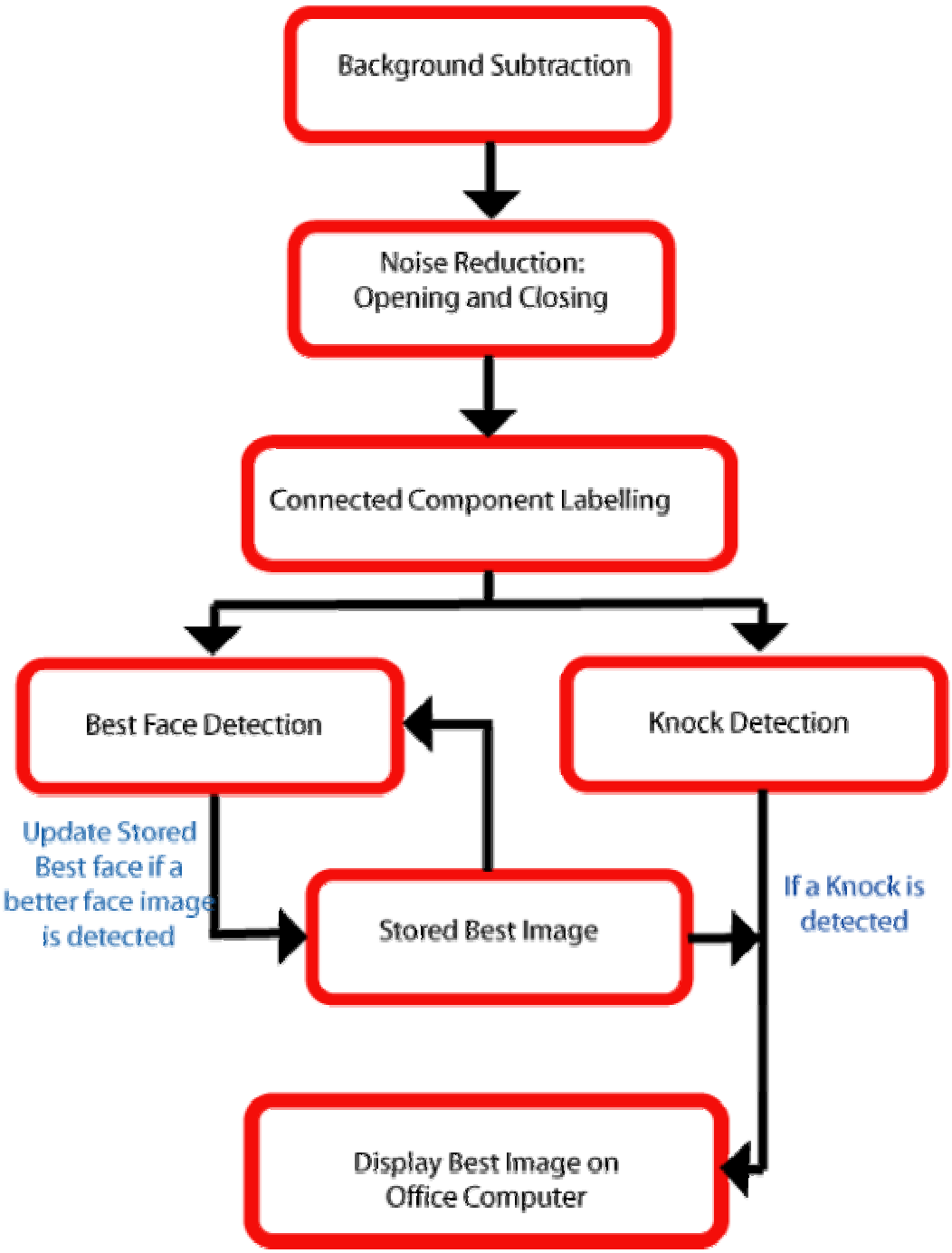


Figure 28: Overview of Image Processing Architecture

Chapter 5

5 Image Analyses

In this chapter a detailed description of how the image processing is implemented is explained. The objective of the Image Analysis stage is to detect a knocking gesture in the input and also to get the *best face image* of a person in the corridor.

5.1 Background Subtraction

Background Subtraction is the first step in the Image Processing stage. Background Subtraction is a method used to detect motion and track movement in a scene. It works by subtracting a background image with nothing moving in it against every new frame. Each pixel in each frame of the video sequence is subtracted from the equivalent pixel in the background image. This is shown below in Figure 29.

Person in current frame

The result of a Image subtraction on the current frame with the background image

Empty Background

Figure 29: Background Subtraction

With Background Subtraction each pixel in each frame of the video input is compared to the pixel in the background. If the difference in the RGB values is greater than a certain threshold the pixel is considered to be moving.

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
            Get background(r,c) pixel
            Get currentFrame(r,c) pixel
                  If DIFFERENCE BETWEEN background(r,c) and
                  currentFrame(r,c) > threshold
                        PIXEL IS MOVING
                  Else
                        PIXEL IS BACKGROUND
                  End If
                  Next Column
                  Next Row
```

5.2 Opening and Closing

The next step, after Background Subtraction has taken place is to perform noise reduction on the resulting image. The purpose of doing Opening and Closing is to remove objects too small to be valid trackable regions and is also required to connect objects that have been split by noise after the background subtraction.

Closing is applied after the image subtraction has taken place. This is shown in Figure 30 on the next page.

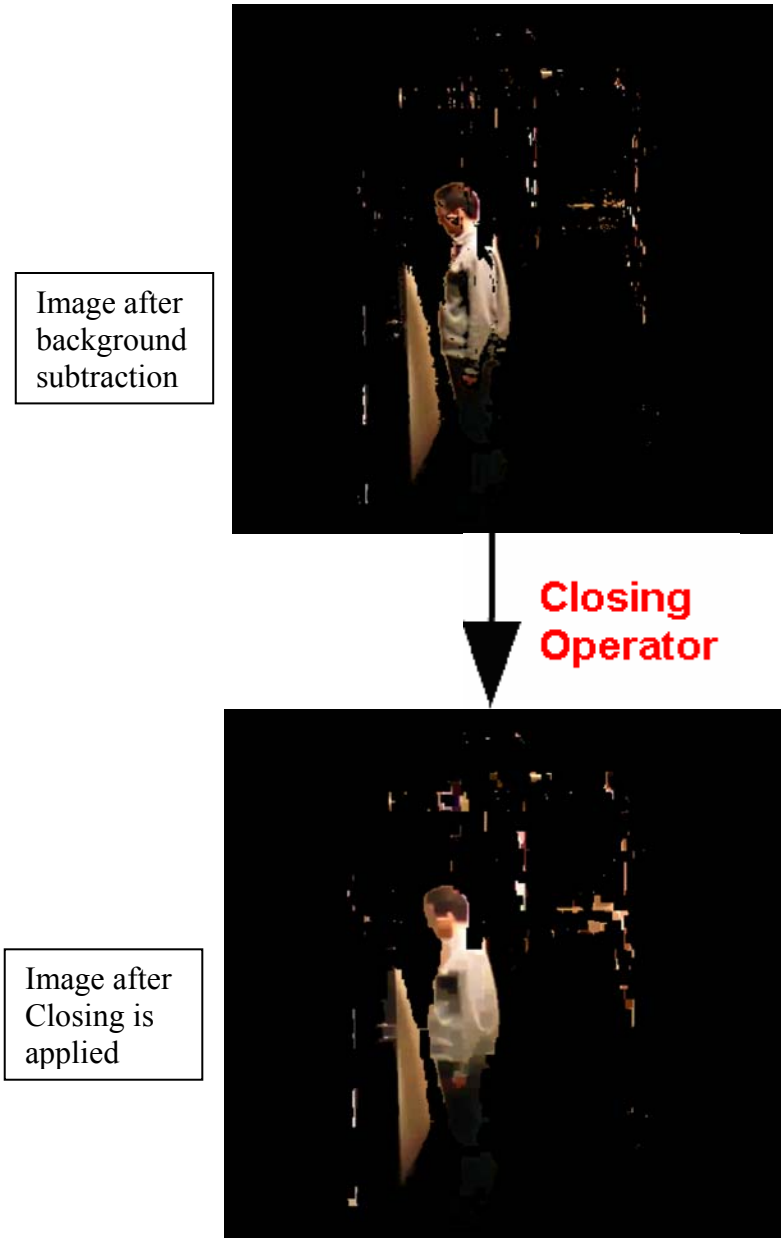


Figure 30: Closing Operator

Opening is applied after the closing operation has completed as shown below in Figure 31.

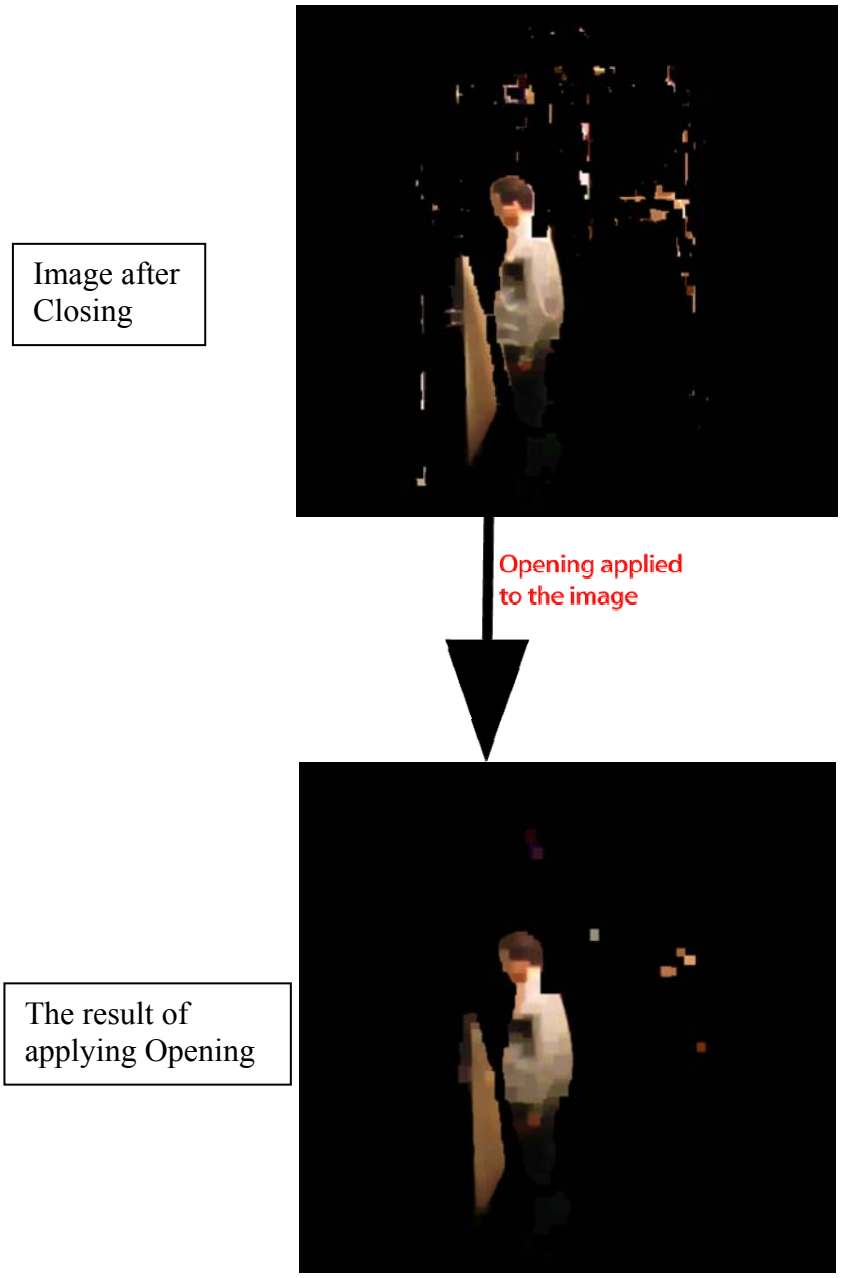


Figure 31: Opening Operator

5.3 Connected Component Analysis

After the Opening and Closing operation have completed the resulting output must be labelled. Labelling is needed to identify if there is a person in the corridor first of all. Labelling is also required in order to be able to track the person and identify their movements and position in the corridor. Figure 32 shows the connected components and a different colour assigned to each region, with each region having a different label. Only the largest region is deemed as output as the rest of the pixels are considered to be noise and so are ignored. If a region is larger than a specified threshold it is assumed to be a human and is deemed eligible for tracking. Figure 32 on the next page demonstrates the output of labelling.

Figure 32: Result of Labelling

Connected Component Analysis or Labelling is implemented in the following way. In this project an iterative approach to Connected Component Analysis is used. The image is searched from the top left corner. For every non-black pixel that it detects it checks the pixel to the left and the pixel on top of it. If none of these pixels (top or left of pixel grid) has a label assigned to it, the pixel is assigned a new label

and the label count is incremented by one. If the top has a label and the left does not have a label the pixel is assigned the same label as the top. Likewise if the left pixel has a label and the top pixel does not have a label the top pixel is assigned the label that the left pixel possesses. If the top pixel and the left pixel both have a label assigned to them we add this equivalence to an equivalence table. The lowest label is always selected as the label to be used and this is noted in the equivalence table.

So for example if the top label was 5 and the left label was 9 in the equivalence table it is noted that every occurrence of 9 must be replaced with 5 and the current pixel is labelled 5.

After every pixel that requires a label is assigned one, the equivalences need to be resolved. This is required to be done so that all connected pixels have the same label. Every occurrence of a label is also counted so that the largest region can be found.

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
            Get currentPixel(r,c)
            Get topPixel(r,c)
            Get leftPixel(r,c)
               If currentPixel(r,c) is moving
                   If topPixel(r,c) no label && leftPixel(r,c)no label
                        Assigned new label
                        Increment label by 1
                   Else if topPixel(r,c)label && leftPixel(r,c) nolabel
                        Currentlabel = topPixel label
                   Else if leftPixel(r,c)label && topPixel(r,c) nolabel
                        Currentlabel = leftPixel label
                   Else if Equivalence occurs
                        Add to equivalence table array
               Else label is 0
            Next column
      Next row
```

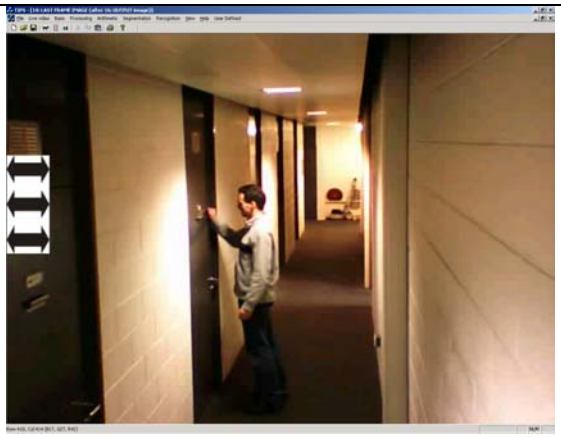
Connected Component Analysis is a computationally expensive part of the overall algorithm and must be run for every frame that requires processing.

5.4 Tracking

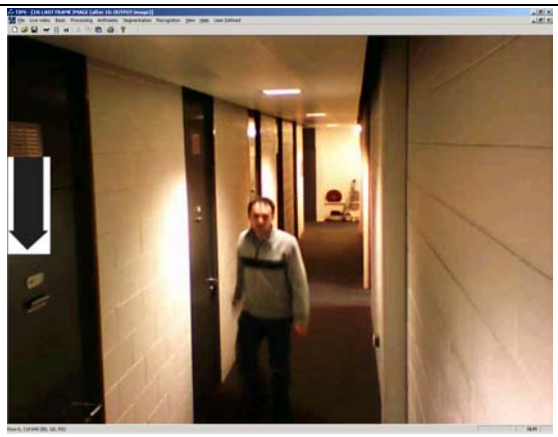
The next step after Labelling has occurred is to track the person's movement in the corridor. The optimum picture of a person is when they are walking towards the camera as opposed to away from the camera.

A person in the corridor is always in one of three states, they are moving up the corridor, moving down the corridor or they are stationary in the corridor. Tracking is also used to identify which door the person is knocking on in the corridor. This is important so that the image of the missed visitor can be sent to the correct machine.

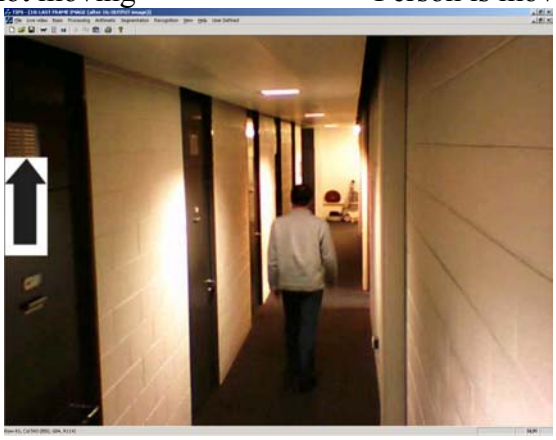
The tracking used in this project is a very simple algorithm. The program tracks the top of the users head. The lowest row value, which is the top of the person, is stored in an array. This array contains the lowest row value for the last ten frames. This is the top of the head for the last ten frames. The direction of movement is then determined from the analysis of these ten values. To avoid noise affecting the results it was decided that instead of comparing the current frame to the last frame, the current frame value is compared to the value of two frames previously. If the current row value is greater than the row value of two frames previous the down count is incremented. If the current row value is less than the row of two frames previous the up count is incremented. If they are equal the equal count is incremented. Figure 33 shows an example of the three states that a person may be in. An arrow is placed on the left hand side of the screen to show the current direction that the person is moving.



Person not moving



Person is moving towards the camera



Person is moving away from the camera

Figure 33: Different direction state that a person can be in

This method has the advantage of being computationally inexpensive but is prone to noise. The algorithm for his method can be seen on the next page.

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
            If pixel is first non zero valued
                  Add to DirectionArray
      Next column
Next row

For size of DirectionArray[]
            If DirectionArray[current]> DirectionArray[current +2]
                  Increment DOWN
            Else If DirectionArray[current]<DirectionArray[current +2]
                  Increment UP
            Else If DirectionArray[current]=DirectionArray[current +2]
                  Increment SAME
            End if
Next
```

Greatest value of DOWN, UP and SAME is the direction.

5.5 Knock Detection Convex Hull

After an image has been labelled the actions of the person in the corridor must be monitored to detect if they knock on a door. To detect if a person knocks on a door, Convex Hull Detection was used. In Figure 34 you can see the result of doing Convex Hull Detection on a person that is knocking on the door. The green area is a concave that is formed by the hand knocking on the door. In a knocking motion the hand comes up to almost the same level as the top of the head. The angle between these two points (X and Y in Figure 34) plus the area of the Convex (the green area) hull is used to detect a missed knock in the corridor.

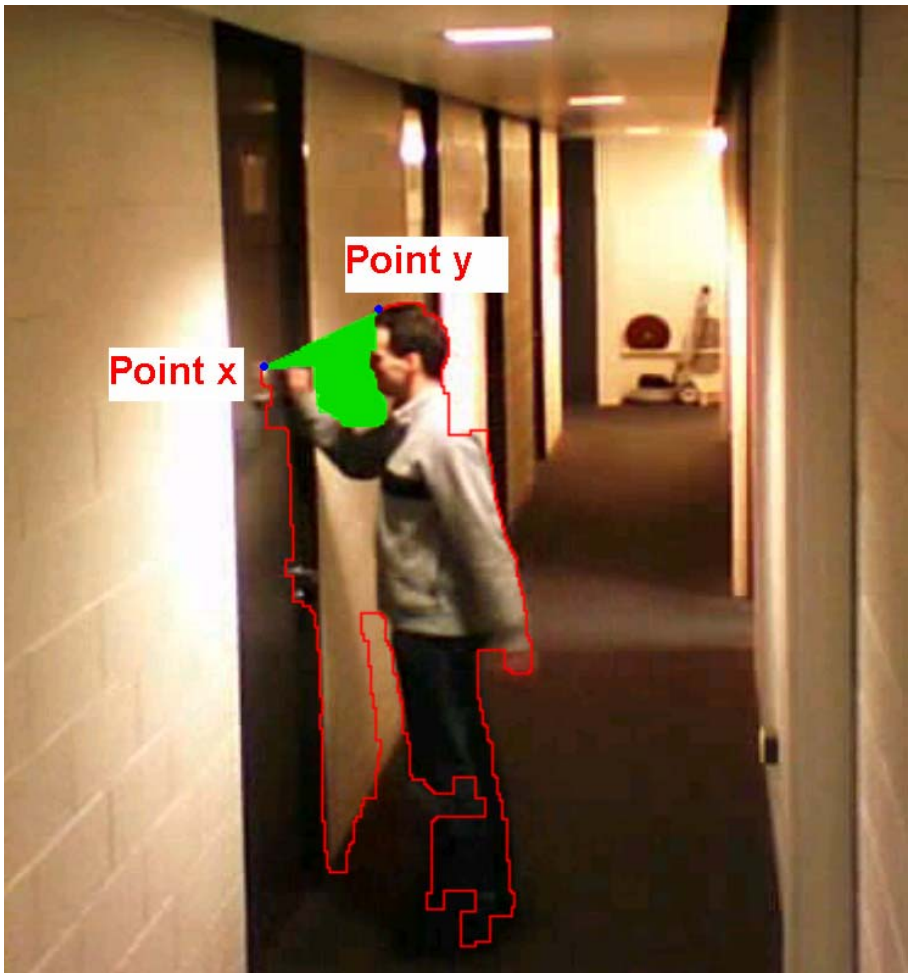


Figure 34: Convex hull detection

The Convex Hull construction was implemented in the following way. It starts off finding a starting pixel. This is the first pixel that is encountered. This starting pixel has the lowest row value and is the upper most pixel value on the region to be processed. The boundary of the region is searched in an anti-clockwise direction and the angle between the current pixel and every element on the boundary is calculated. The algorithm for a boundary search is no trivial task and how it was implemented is explained below in section 5.5.1. The pixel that has the least angle between it and the start pixel is a vertex of the Convex Hull. The search now moves to this pixel and again the angle of every boundary region is calculated to get the least angle again and the pixel with the least angle is another vertex. This process

continues until the current pixel equals the start pixel. All the points that had the least angle are the vertices of the region Convex hull.

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
       StartPixel equals lowest row with lowest column
        While currentpixel not equal Startpixel
         Search region boundary in anticlockwise direction(Boundary Al)
         Compute angle for every boundary point after currentpixel
         Get MinAngle between Currentpixel and boundary Points
            Move CurrentPixel to this point
            This point is an element of the region convex hull
         Repeat until CurrentPixel equals StartPixel
        End While
```

5.5.1 Boundary Tracing

Boundary Tracing is required for the Convex Hull detection to function. Figure 35 is the output of Boundary Tracing on a labelled object [Sonka].

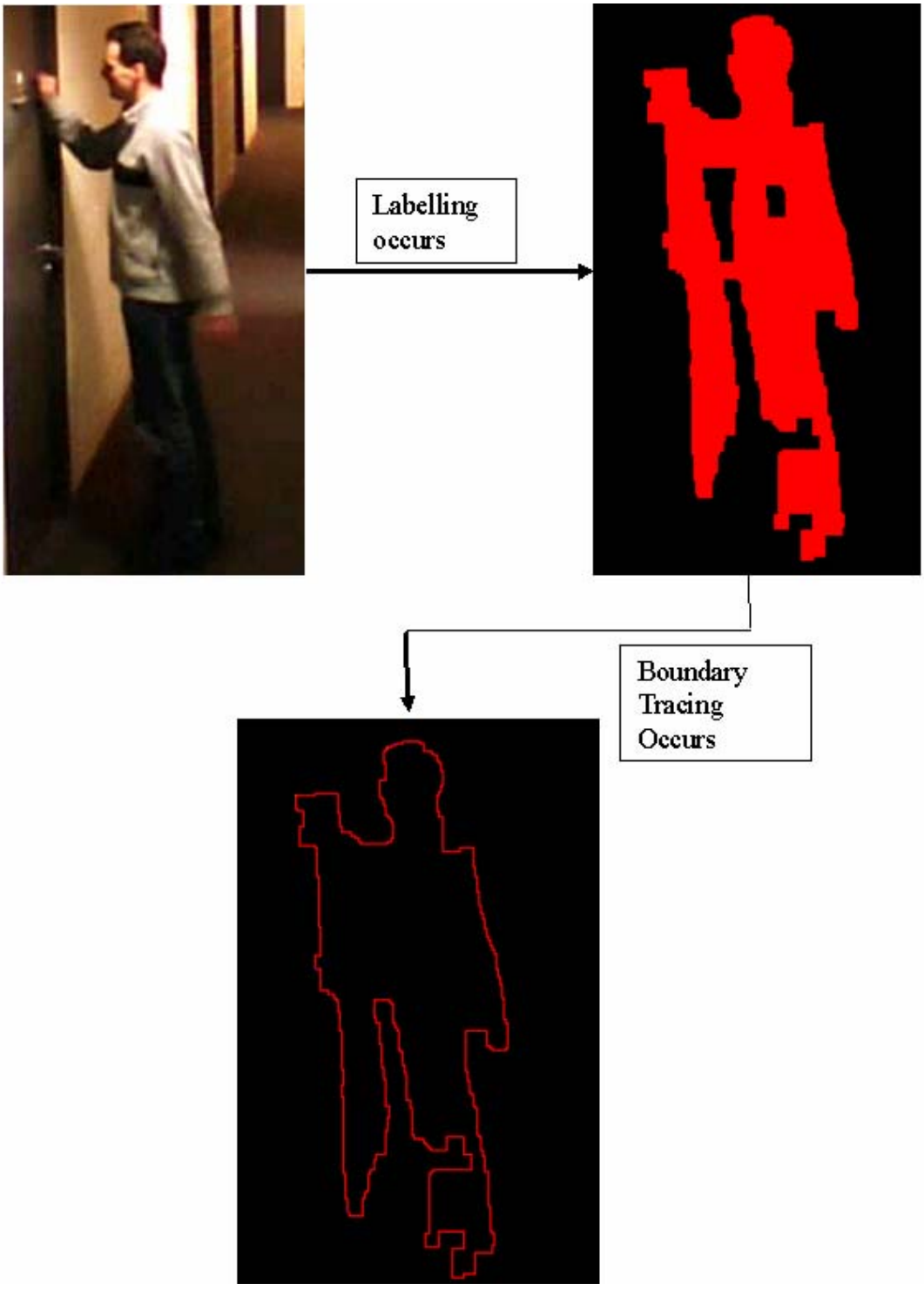


Figure 35: Boundary tracing

To find the boundary of a shape the direction of the previous move along the border is required. For this project there were four possible

direction values. These possible directions can be seen in Figure 36 [Sonka]. To find the next element of the boundary a 3x3 neighbourhood search is required. This is where the previous direction is required. The neighbourhood search begins in the last direction plus three. This means that if the direction of the last move along the border was 0 the search will begin at (0+3=3 therefore direction 3) position 3 from the current pixel. Diagram 37 [Sonka] shows an example of where the search begins when the previous direction was equal to three. The first pixel found with the same value as the current pixel is a new boundary element. The direction value is then updated depending on the direction that the new boundary element is. This search continues around the shape until the current pixel is equal to the start pixel.

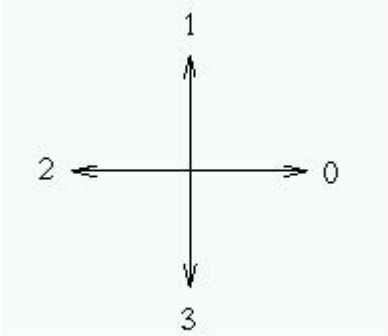


Figure 36: Four possible boundary search directions

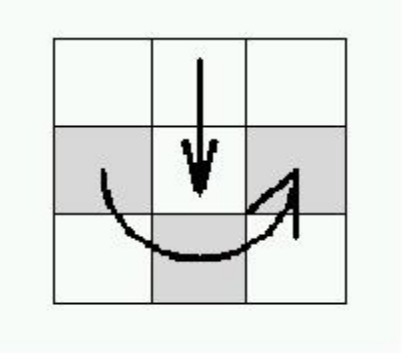


Figure 37: Pixel neighbour search sequence

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
       Find StartPixel with Min row and Min column value
       Direction will store direction of last move along boarder
      While newBoarderPixel not equal StartPixel
       Search 3x3 neighborhood of current pixel
        check to right of direction
        check in front of direction
        check to left of direction
        first pixel found with the same value a the current pixel is a
        newBoarderPixel.
        Update DIRECTION
       End While
      End for
End for
```

5.6 Head Detection

Head Detection occurs before the Prewitt operator so that only the head region is processed for Edge Detection. Figure 38 shows the edges that would be processed with out a Head Detection algorithm. Figure 39 shows the images that are processed after the head detection.

Figure 38: Before Head detection

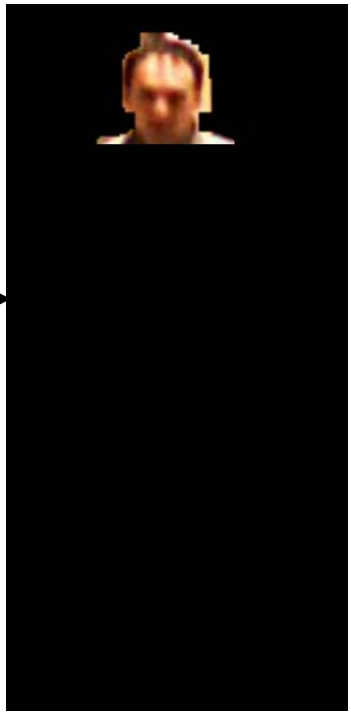


Figure 39: After head detection

To get the head region out of the image, many different methods were investigated. The method used calculates the head width first and then multiplies this by 1.5 to get the head height. The greatest distance between the left most pixel and the right most pixel that is with in fifteen pixels from the top of the head and are on the same row is taken to be the heads width. The head's height is then calculated by multiplying the width by 1.5 (fixed value). This leaves us with the head region. This algorithm works well for when a person is near or far away from a camera.

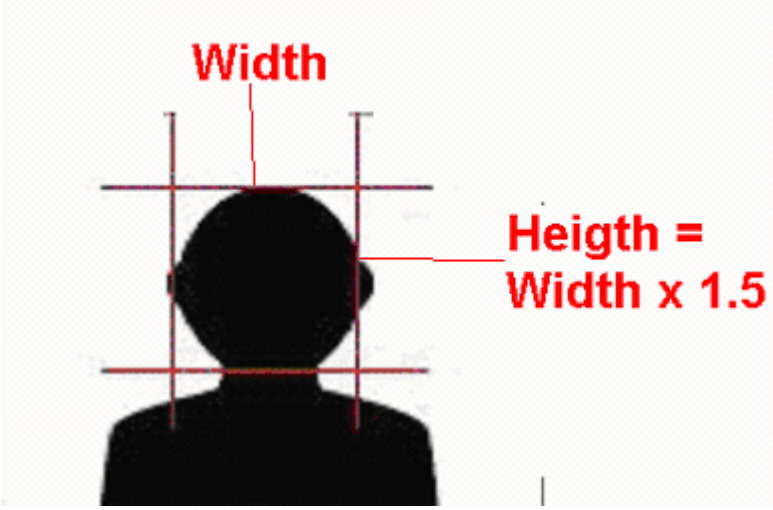


Figure 40: Calculating head height

Algorithm:

```
For each row (r) in the image
      For each column (c) in the image
       Get StartRow
        For each row within 15 pixels from StartRow
            Get lowestCol non zero column value
            Get highestCol non zero column value

      faceWidth equals lowestCol minus highestCol
      faceLength equals  faceWidth multiplied by 1.5
```

5.7 Best Face Detection Prewitt Edge Detection

In order to capture a recognisable image of the missed visitor in the corridor a *Best Face* Detection method had to be devised. *Best Face Detection* is running at the same time as the *Knock Detection*. The Best Face Detection also starts it's processing with the labelled image. To decide which image (from the time that a person enters a corridor until the time that they leave the corridor) is the best image of a person, Edge Detection was used. An edge is a jump in intensity from

one pixel to the next. Facial features such as eyes, nose and mouth have very sharp edges. This factor was used to detect the best face.

Prewitt is the type of Edge Detection that is used. For this project it was decided to just detect edges along the x-axis, as this is where facial features such as eyes and the mouth have sharpest edges. Figure 41 points out this point.

Figure 41: Face image mostly horizontal edges

The Edge Detection algorithm takes the head region of the person as an input. This can be seen in Figure 42. The first step is to convert this to greyscale. In TIPS there is a built in function to convert an image to grey scale. The Prewitt operator then attempts to detect edges on this grey scale image along the horizontal line only. If an edge is detected a high grey-level is put in the output. If no edge is detected at all a black value is placed in the output image.

Figure 42 shows the steps required to get to an image where higher grey-level values indicate the presence of an edge between two pixels in an image.

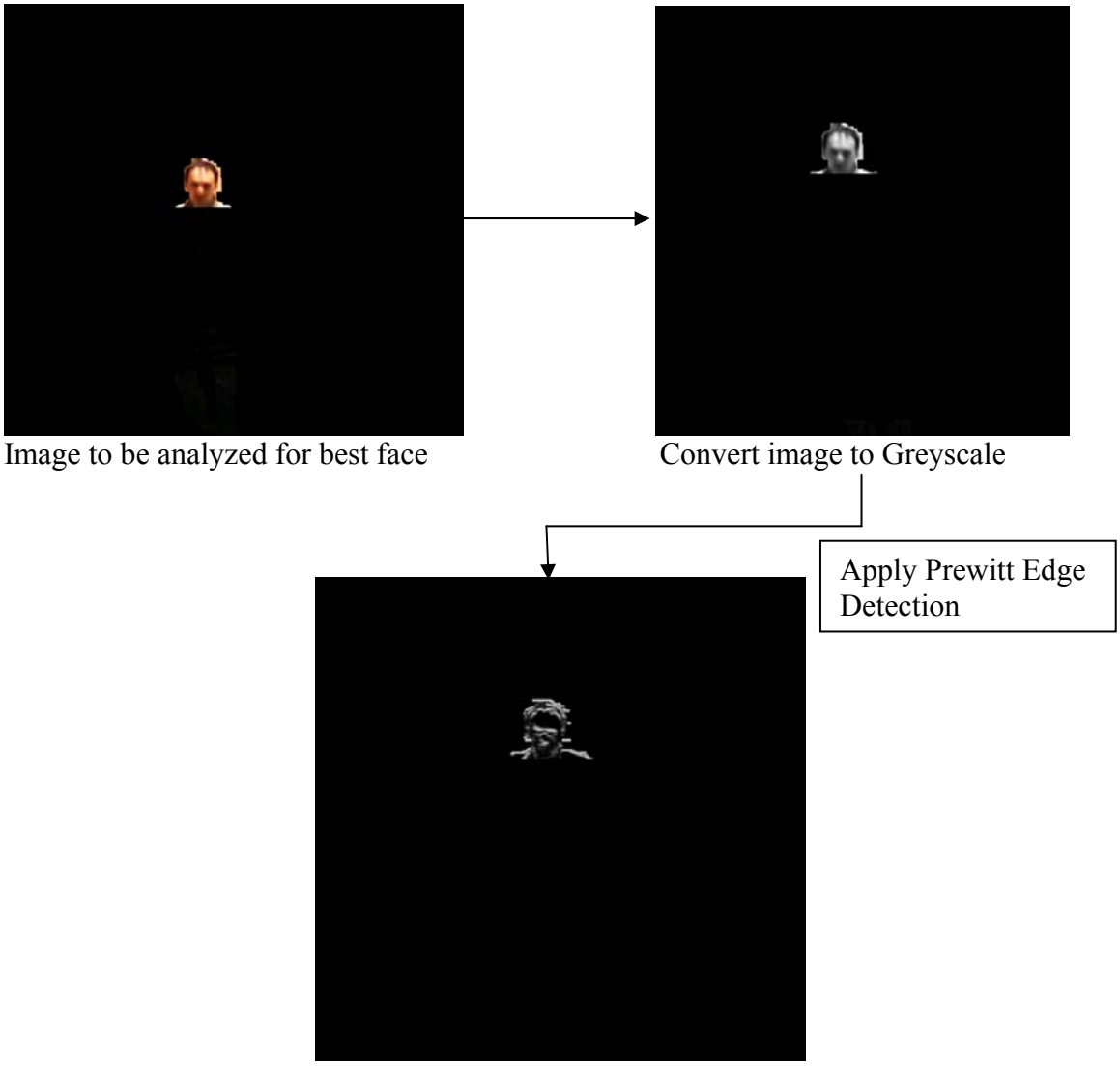
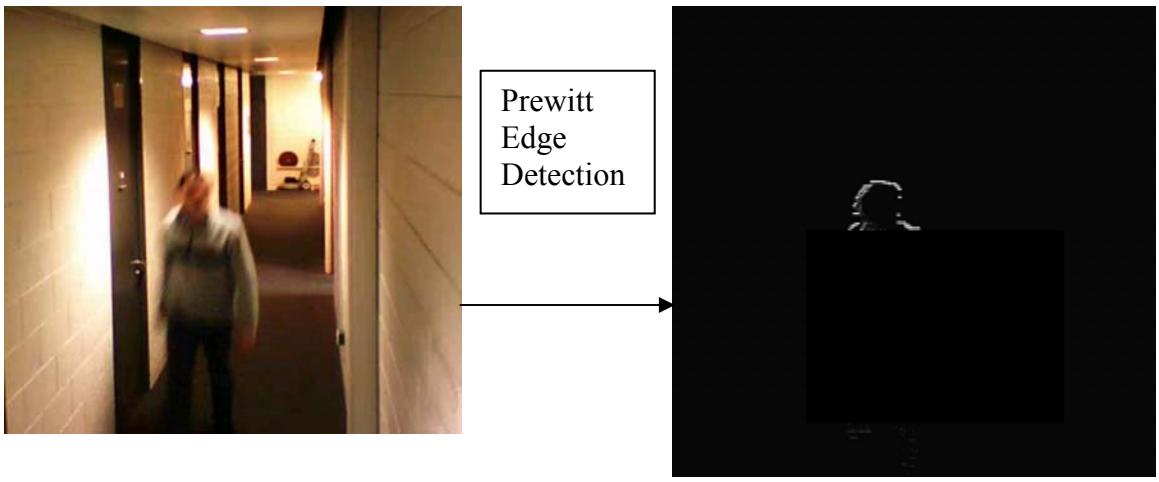


Figure 42: Prewitt edge detection

Figure 43 shows the result of doing Prewitt edge detection on a blurred face image. It is obvious from the output that the amount of high grey levels in the output are much less than that of Figure 42. A calculation is completed to get the average grey level of the image. The calculation takes into account that border elements are not counted. The image with the highest average grey level is determined to be the *best face image*.



A blurred image of person in corridor

Edge Detection on blurred image

Figure 43: Example of Prewitt edge detection on a blurred image

Chapter 6

6 Communications

In this chapter we will discuss the communication process between the Image Processing Application and the display application on the office machine.

The Image Processing Application will result in a *best face image* of the missed knocker, the time of the knock and a door number that the person knocked on. The *best face image* is the best image of the missed knocker from the time that they enter the corridor until they leave. This best image needs to be transported over to the correct office to be displayed. The Image Processing Application will have to be able to send this best image to a number of different offices. The method used to send this image to the different offices and the communication channel used is discussed below.

6.1 The Client/ Server Model

For this project to work it requires that the Image Processing Application be able to provide the *best face image* to a number of different applications all running on different machines in various locations. In this case the various locations are different offices in the corridor. It was decided that a Client Server model [16] would be used. Most inter-process communication uses the client server model. Client server terms refer to the two processes, which will communicate with each other. This is the type of communication model used in this project. Figure 44 shows an outline of what the Client Server Model looks like for this project.

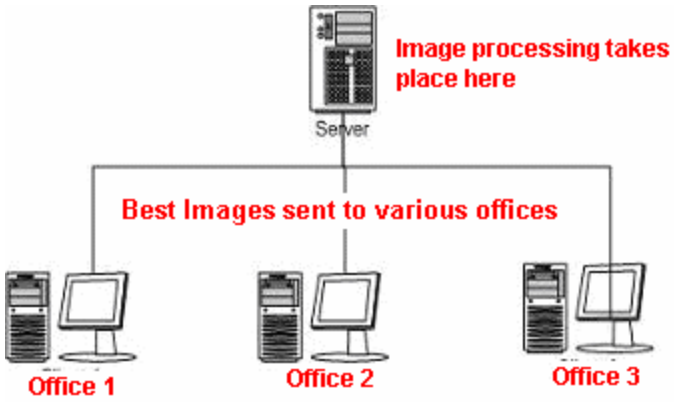


Figure 44: Client / Server Model

For the workings of this project the roles for the client and server are switched around. Instead of having the Image Processing Application listening for connections from the office computers, the office computers are listening for connection from the Image Processing Application. The Image Processing Application will only connect to a client when it has a *best face image* to send to it. All of these clients are listening on a certain port number and they are all accessed using their unique IP addresses.

6.2 Sockets

The Image Processing Application was developed in C++. As a GUI was required on the office computer to display the best image it was decided that this GUI would be developed in Java. GUIs in Java are platform independent. As we now have a server in C++ and a client in Java it was decided that sockets would be used as the communication channel between them. The type of socket used is a TCP socket [10]. With TCP sockets once a connection is established, both sides can send and receive information. The Image Processing Application will only

make a connection once it has detected a knock and has a *best face image* of the missed visitor. This office application will be listening on a certain port and once a connection is made it prepares to accept an image file.

Chapter 7

7 User Interface

Java is used on the client side to develop a GUI to display the missed visitor. The office computer receives the best image from the Image Processing Application via sockets. It then must display this image to the person in the office. Below is the GUI that displays these missed visitors. The user will have the option to select from a number of missed visitors along with being able to see the person that knocked on their door and the time of the missed guest is also displayed.

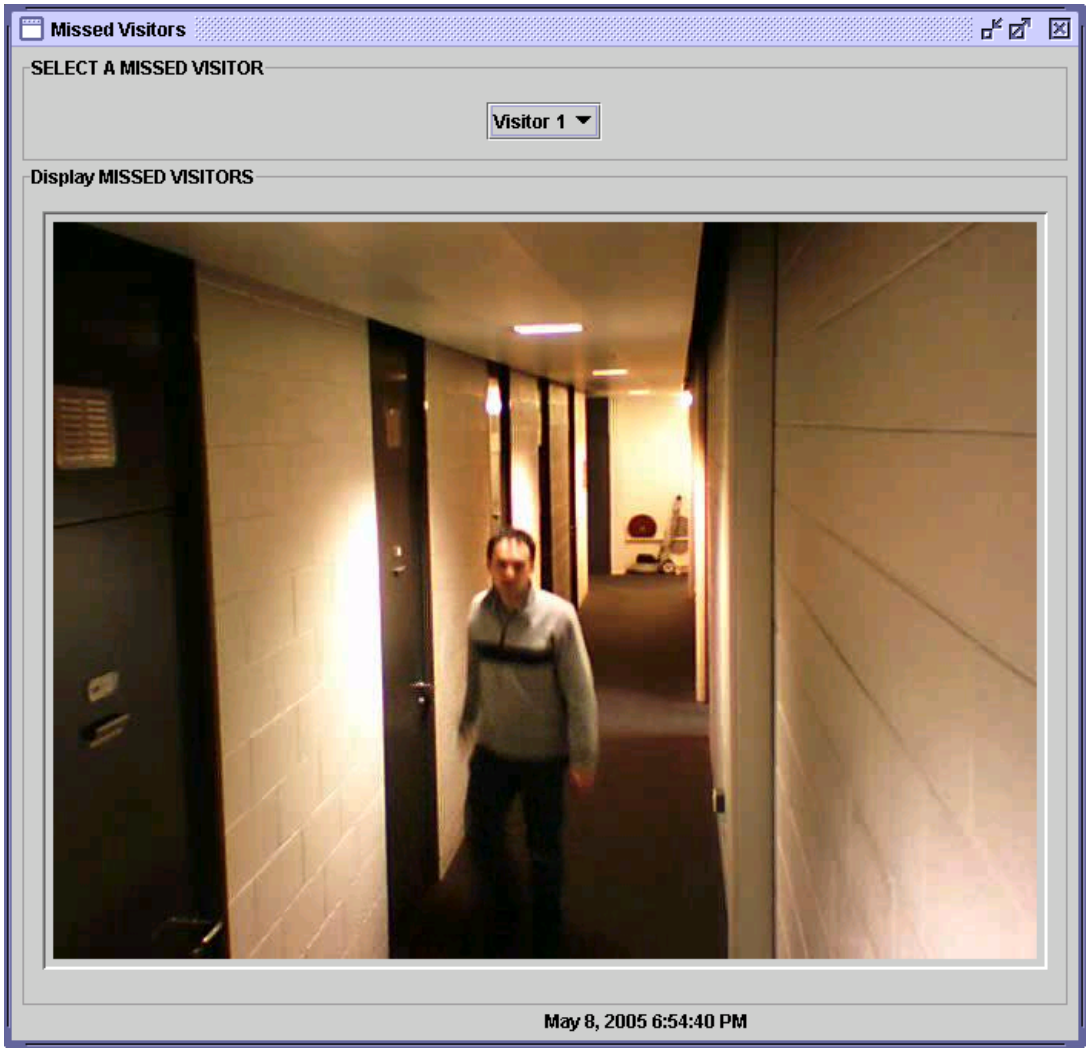


Figure 45: Output that the User will see in the office

Chapter 8

8 Evaluation

This chapter takes a look back at the project and the results of the project are shown. Elements of the project that were successful are discussed as well as areas where difficulties were encountered. The project was felt to be a success. In the end the system was able to detect a knocking movement, detect a *best face image* and display this missed visitor *best face image* in the correct office. The main goals of the project were reached. An example of the full working system can be viewed on the CD provided with this Thesis.

8.1 Results

This project shows that it is possible to create a fully automated real-time Missed Visitor Detection System. This section will show some sample output from the system and also discusses some of the problems that occurred.

The images on the next few pages are examples of the visual outputs that appear in the TIPS system. These images are being taken every three to four seconds. In Figure 46 you can see that there is no one in the corridor. As there is no one in the corridor, no image analysis of the hall is being completed. In the bottom-right corner of the screen the best picture of the person in the corridor is displayed. As there is no person in the corridor there is no best picture to be displayed and the bottom-right corner is blank.

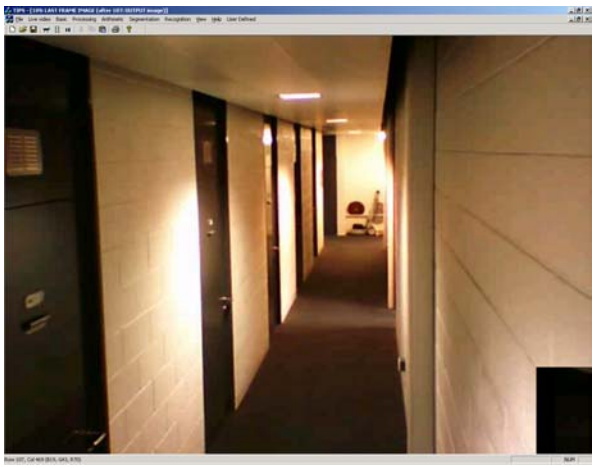


Figure 46: Empty corridor

Figure 47 shows a person enter at the bottom of the corridor and their motion is up the corridor away from the camera. The arrow on the left pointing up indicates the direction that the person is moving. As the person is moving up the corridor no *best image analysis* occurs as a picture of the back of some ones head is of no use. The red line around the person is the boundary that is required for the *knock detection*.

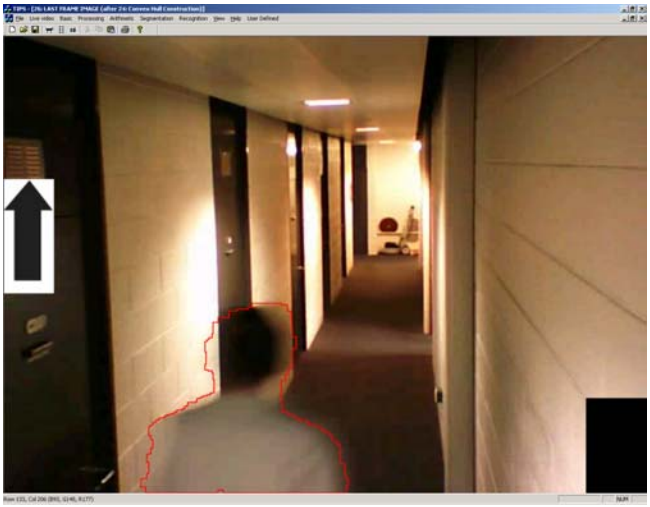


Figure 47: Person enters corridor and is moving up the corridor

You can see in Figure 48 that the Person has come to a stand still outside the office door. The movement is now neutral and the

horizontal arrows on the left side of Figure 48 indicate this. The best picture is also now taken, as movement is no longer up the corridor. The best picture is updated in the bottom right hand corner of the output

Figure 48: Person standing at the door

Next you can see a person knocking on the door. The "knock knock" that appears in the top left of the screen indicated that a knock has been detected. Also you can see the concavity formed by the knocking motion is outlined in green. This is the area that is evaluated to decide if a knock has occurred.

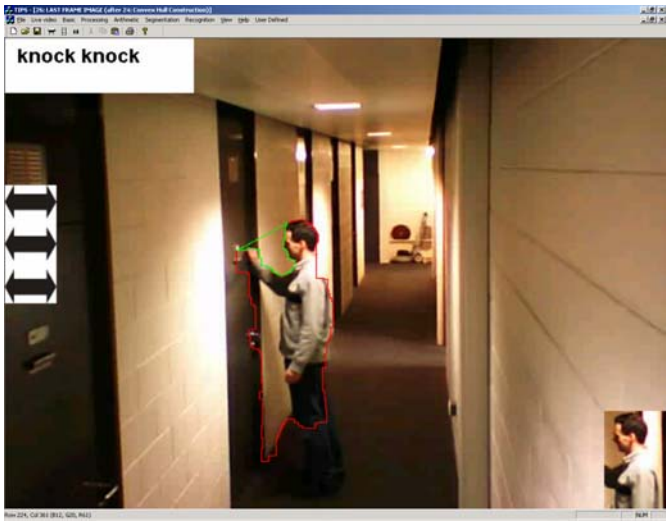


Figure 49: Person knocking on an office door

After the person knocks on the door they leave as there is no one in. The person begins to walk towards the camera. The direction of their movement changes from neutral to down. The arrow on the left of the screen now changes to down. The system has still not detected a better face image so it does not update the best image in the bottom-right hand corner.

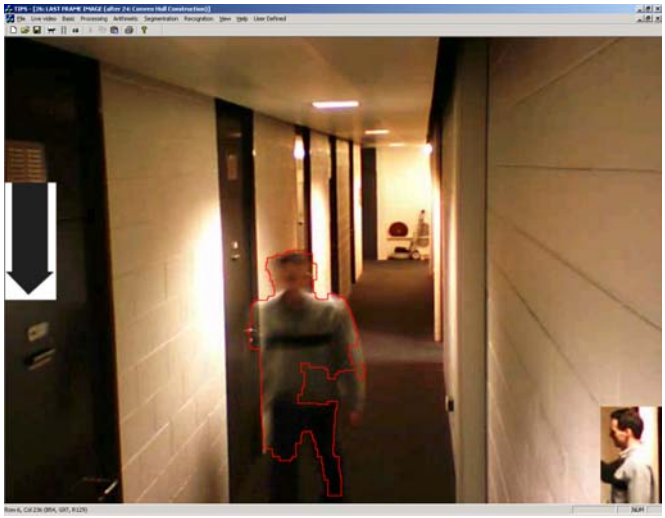


Figure 50: Person Walking away from office door and down the corridor

After the person leaves the corridor again the image processing stops. As can be seen in Figure 46 the corridor is empty and there is no best image in the bottom-right hand corner.

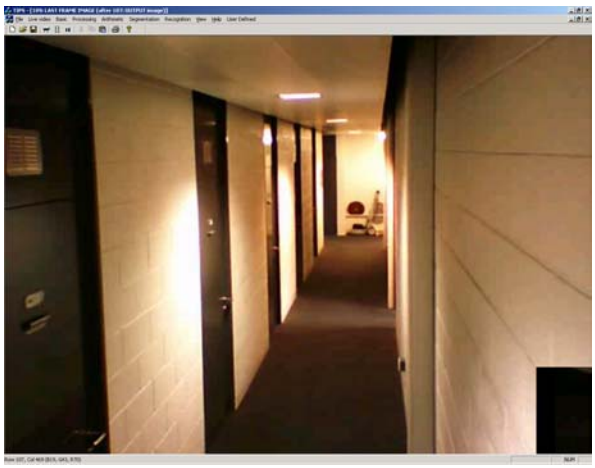


Figure 51: Person leaves the corridor. Processing stops, as corridor is empty

Once the person leaves the corridor the best image is sent to the display application in the office. On the next page Figure 52 shows the display application that will be displayed on the office occupants computer. The best image displayed on the office computer is the same best image that appears in the bottom right corner of Figure 50 above.

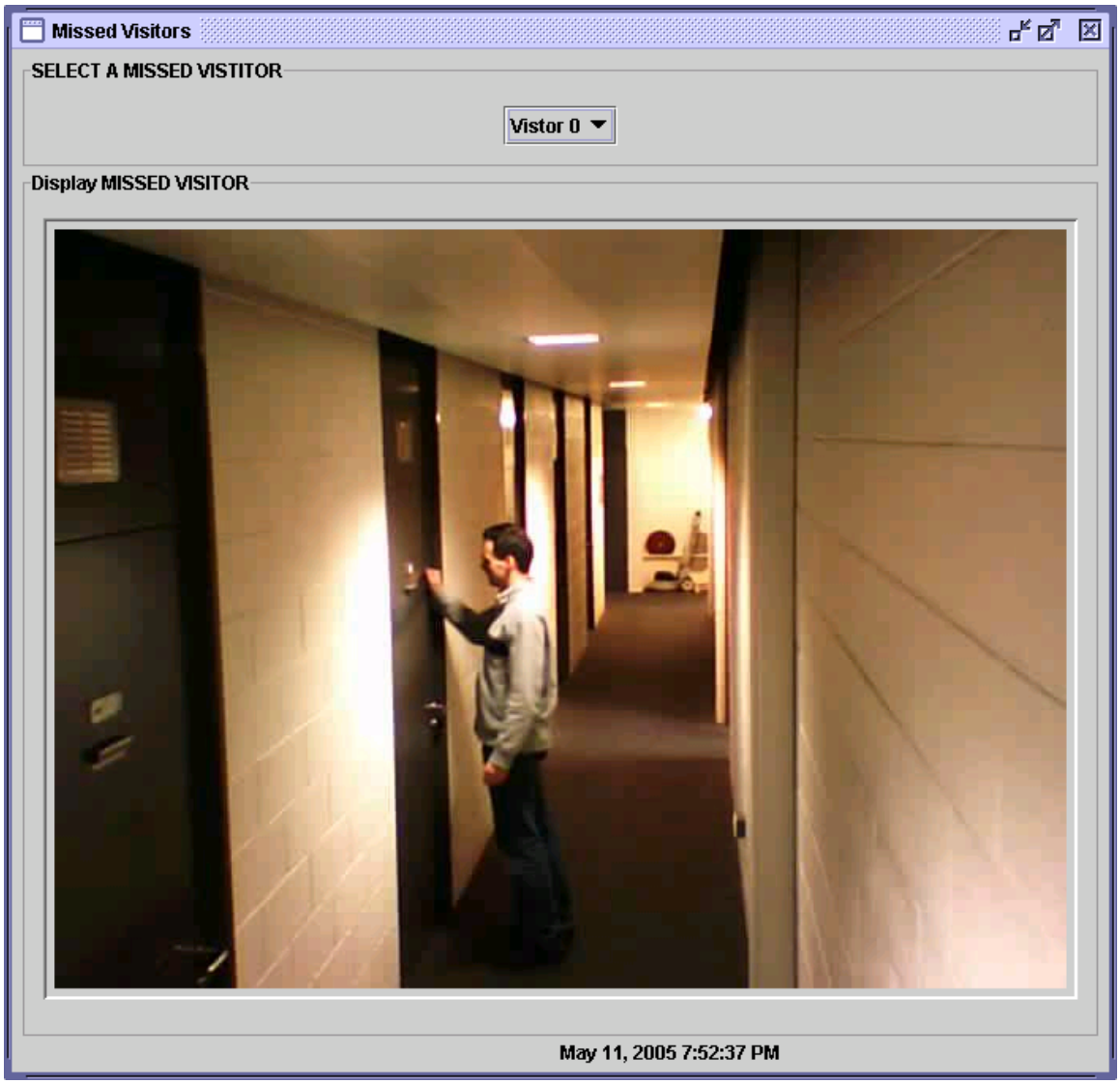


Figure 52: Display GUI that appears on the office computer

For a more detailed sample please see the .avi movie file on the CD that is provided at the back of this book.

8.2 Difficulty's Encountered

8.2.1 Shadows

Shadows were a major problem in this project. Due to the brightness in the corridor, shadows created are very strong. This can be seen in Figure 53 where the shadows that are created are very prominent. Shadows can be attached to a person and after Image Subtraction

they are treated as part of the person. As can be seen in the diagram below, the arm extending to knock on a door casts a shadow under it, which is connected to the arm.

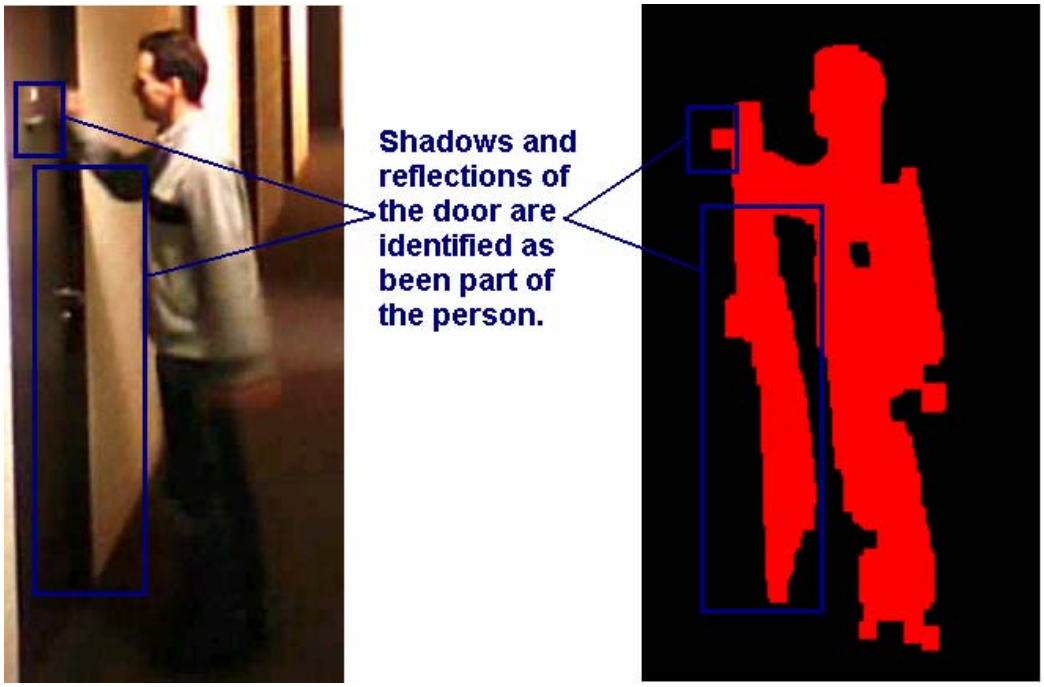


Figure 53: Example of shadows problems

A method was implemented to detect shadows and attempt to remove them. Figure 54 shows the results of this implementation.

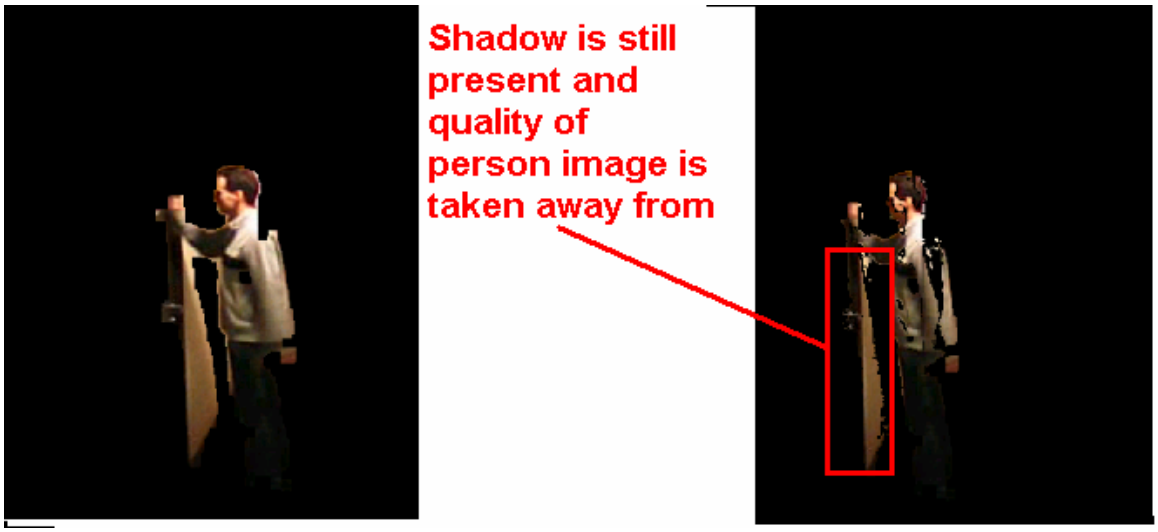


Figure 54: Shadow removal

As can be seen above the shadow removal did not function correctly. It did remove some of the shadow but stronger regions of the shadows remained. The shadow removal process also removed some valid pixels belonging to the person's body. The algorithm to implement shadow removal was also computationally very expensive and it slowed the system down to an unacceptable level. For these reasons it was decided that shadow removal would not be implemented in this project.

8.2.2 Colour Collisions

The system has problems when the moving pixel has the same colour as the static background. The moving pixel will be classified as part of the background. In Figure 55 the person in the corridor has their head cut off, as it is the same colour as the wall in the background. This is not a straightforward problem to solve as how can a background pixel and a moving pixel be distinguished between if they are exactly the same colour.

Figure 55: Colour collision

8.2.3 Problem with Tracking System

Problems with Tracking in the corridor were also encountered. Due to the fact that shadows remained in the systems this meant that a Centroid of the person in the corridor could not be used to track the person. A Centroid is the centre point of an object. Due to the presence of connected shadows, this centre point of the object moved around a lot. Figure 56 is an example of the Centroid point moving due to the presences of shadows. The Blue cross represents the centre points of the red objects.

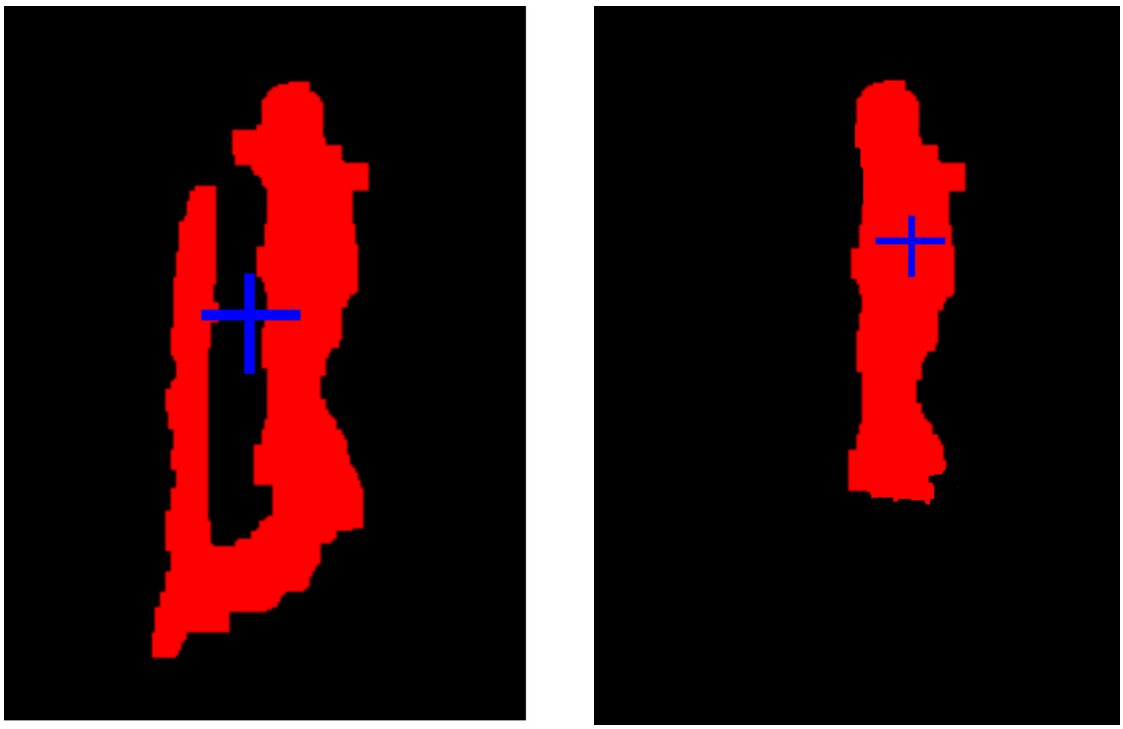


Figure 56: Center point movement due to shadows

Due to these reasons it was decided that to track a person the pixel with the lowest row value would be used to track the person. This pixel is situated on the top of the person's head. The problem with using the top of the head to track a person is that there is a very small movement in the top of the head from when a person enters at the bottom of the corridor until they leave at the top.

Figure 57 shows that there is only a movement of 60 pixels from when they enter the bottom of the corridor until they exit at the top.

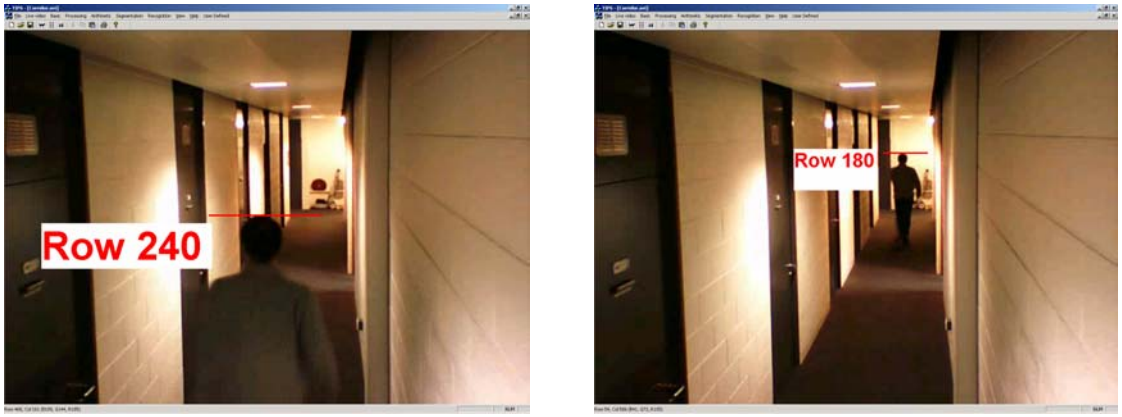


Figure 57: Row Movement in corridor

8.4 Future Work

Due to the time constraints, there are several improvements that could have been applied to this project, which were left out. Primarily, the speed at which the whole system runs is quite poor. Algorithms need to be optimised so that the overall speed of the project runs much faster.

Also the current project only processes one person in the corridor at any one time. The system should be able to track several people in the corridor and detect several knocking motions at once. This would be relatively straightforward to do as the hard part of identifying a person knocking in a corridor is implemented.

Another improvement that could be implemented is a changing background that would accommodate objects being left in the corridor. If an object, for example a bag is left in the corridor this bag is added to the background scene for background subtraction. This could be implemented by detecting if an object does not move for ten frames it becomes part of the background. This would not have been very difficult to implement but due to time constraints it was not implemented.

An improvement in the quality of the *best face image* could be made if a high-resolution camera is used.

8.5 Conclusion

I felt that the project was a success, as all the requirements were met. The person in the office does receive a best image of the missed visitor. This was the main goal of the project and it was achieved. The detection of the knock using Convex Hull Construction worked very well and was one of the resounding successes of the project. The Best Face Detection using Prewitt Edge Detection worked very well also.

The project works well within certain constraints and with further work it could be scaled to a fully working implementation. The only problem that such a system would have to overcome before it could be extended into a commercial prospect is the problem of privacy. Even trying to get a camera installed in this college caused many problems as every person in the building had to be asked for permission to install the camera.

Personally I really enjoyed working on this project and have learned a lot about Image Processing that I was unaware of prior to commencing this project. What I found very interesting is that you can take a basic Vision technique such as Edge Detection and use this to detect a best face, for example. The number of possible applications using these relatively simple vision methods is countless. I was very happy with the final outcome. At the start of the project I had miscalculated just how much work would be required to implement a working system. But thanks to some late nights and excellent direction by my supervisor Kenneth Dawson-Howe the project was a resounding success.

Chapter 9

9 References

[1] Computer Vision - A Modern Approach. Forsyth and Ponce Prentice Hall, 2003.

[2] "Snakes: Active contour models." International Journal of Computer Vision M. Kass, A. Witkin, and D. Terzopoulos, 1987.

[3] Background subtraction based on logarithmic intensities Bor-Shenn Jeng, Quen-Zong

[4] Noise reduction Quantitative Imaging Group http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-_Noise.html

[5] A programmable logic based implementation of binary images morphological operators Songpol Ongwattanakul, Phaisit Chewputtanagul, David J. Jackson,

[6] Hypermedia image processing reference Robert Fisher, Simon Perkins, Ashley Walker and Erik Wolfart

[7] Digital Color Imaging connected component analysis a study C. A. Bouman:

[8] Computer Vision and Template Matching, Roberto Brunelli, University of Trento

[9] Human Face Detection and Tracking using Skin Color PREM KUCHI, PRASAD GABBUR, P. SUBBANNA BHAT, SUMAM DAVID S. SMIEEE, MIETE

[10] Beej's Guide to Network Programming http://www.ecst.csuchico.edu/~beej/guide/net/

[11] James Crowley, Grenoble University, Grenoble, France

[12] Philips Research Laboratories Peter B. L. Meijer http://www.seeingwithsound.com/

[13] Vision based system for quality control of food products O. Alhusain a,b, *, Z. Tóth a, Á. Rakusz a, L. Almási a, B. Farkas c

[14] University of Technology in Sydney http://www.uts.edu.au/

[15] Prewitt Edge Magnitude Detection http://www.cs.colostate.edu/cameron/prewitt.html

[16] Client / Server model http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci211796,00.html

Books:

[Sonka]: Milan Sonka, Vaclac Hlavac, Roger Boyle – "Image Processing, Analysis, and Machine Vision".