HAND TRACKING GUIDANCE FOR THE VISUALLY IMPAIRED

BY

TAN CHING SOON

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2013

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _____

_____

_____

Academic Session: _____

I _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:

_____

_____ _____

_____ Supervisor's name

Date: _____ **Date**: _____

HAND TRACKING GUIDANCE FOR THE VISUALLY IMPAIRED

BY

TAN CHING SOON

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2013

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Hand Tracking Guidance for the Visually Impaired**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEGDEMENTS

First of all, I would like to express my sincere thanks to my supervisor, Prof. Maylor Leung Kar Hang for giving me the opportunity to work on this project. Due to the challenging nature of this project, I felt lucky to have met Prof. Maylor as my advisor to assist me in this project. His guidance and wisdom allow me to expand my knowledge and widen my experience in computer vision. He always shares his knowledge and experience to guide me throughout the whole project.

Other than that, I won't forget the contributions from my friends. Their criticisms not only gave me to have more innovative ideas which can improve this project, but also let me feel that I won't get lonely being on my own throughout the whole project.

I had learnt a lot of computer vision knowledge by reading the relevant articles. I found that I was really interesting in computer vision. If it was given a chance to re-pick the project topic, I would like to participate in this project again. Lastly, I would like to thank everyone who had supported me in this project.

ABSTRACTS

Nowadays, assistive technology is advancing at a very rapid pace. Assistive technologies can help people with the disabilities to be more independent at work, in school and at home. In this paper, we demonstrate an assistive human computer interface (HCI) to aid the visually impaired to localize and pick up a specific colored object by hand tracking guidance. Our HCI is able to be worn by the user and carried freely. When the user would like to track the colored object in path of him, our HCI guidance system is able to give correct guidance in term of the direction via audio device. As the contribution of this paper, we strongly believe that the deliverables at the end of our HCI is realistic and practical to improve the quality of visually impaired life in real-time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AdaBoost	Adaptive Boosting
HCI	Human Computer Interface
OpenCV	Open Source Computer Vision Library
SIFT	Scale Invariant Feature Transform

CHAPTER 1 INTRODUCTION

1.1 Problem Description

In recent years, there has been an increasing concern on how to help the visually impaired more accessible to the environment. The visually impaired can refer as the person who suffers vision loss. They have numerous problems and inconveniences in their daily living due to their visual disability, such as information access, mobility, way finding, interaction with the environment. For instance, a person who is visually impaired would like to grab a cup in front of him. However, he couldn't guide his figure with the right direction. Thus it is a challenging task to the visually impaired, but being a normal human, it is incredibly difficult to imagine how hard to perform this action.

Traditionally, most of the visual impaired usually use a white cane to detect objects in the path of him. However, it is not a good alternative as mobility tool for visually impaired because it doesn't ensure its reliability and safety under some certain circumstances. For instance, when the visual impaired is attempting to find the object in front of him, using white cane doesn't help him to recognize the object which he is looking for. In fact, he perceives limited information about environment and even ignores some important clues in term of the object appearance and existence. Therefore it becomes unreliability through shortcoming of using the white cane.

On the other hand, in the new era of technology, assistive technology development has led to focus on helping people who have deficits in physical, mental and emotional functioning. We can find simple instances of assistive technology devices around us all the time. Assistive technology devices are items frequently used by

people with functional deficits as alternative ways of performing action. Assume if there is a kind of assistive device system which can be acted functionally like the human eye, it will be very useful to aid those who are visual impaired to eliminate their weakness and able to do anything which being a normal human could do. Thus in this paper, we demonstrate an assistive human compute interface (HCI) to aid the visual impaired to localize and pick up an specific colored object by hand tracking guidance. This paper is organized as follows. Project background and project objectives are introduced in chapter 1. Subsequently, chapter 2 briefly reviews some related works. In chapter 3, we will describe our proposed HCI system architecture. It is followed by chapter 4 to explain our proposed approaches in detail. Chapter 5 provides experiments results and discuss how well each module performance did. The future research direction is discussed to improve the current HCI system in chapter 6. Finally, we will conclude and reinforce the contribution of this project in chapter 7.

1.2 Motivation

Presently, over 285 million people in the world are visually impaired, of whom 39 million are blind and 246 million have moderate to severe visual impairment. It is predicted that without extra interventions, these numbers will rise to 75 million blind and 200 million visually impaired by the year 2020. As the population of the visual impaired continues to increase, global efforts have been getting aimed at discovering solution to help the visual impaired. Thus, it’s worth investing effort in our project.

In fact, lack of sight becomes a major barrier in daily living. For schooling and working-age visually impaired, they have very high unemployment rate. It means that most of the school and employers cannot accommodate the visual impaired due to their visual disability. In consequence, they face important socioeconomic constraints.

Contribution of our project leads to the alleviation of the visual impaired weakness and rebuilt their self-confidence. It gives the opportunity to the visual impaired to accomplish some routine daily tasks without deploying other people help. The end result of our project is to improve the quality of the visually impaired life.

1.3 Proposed Approach

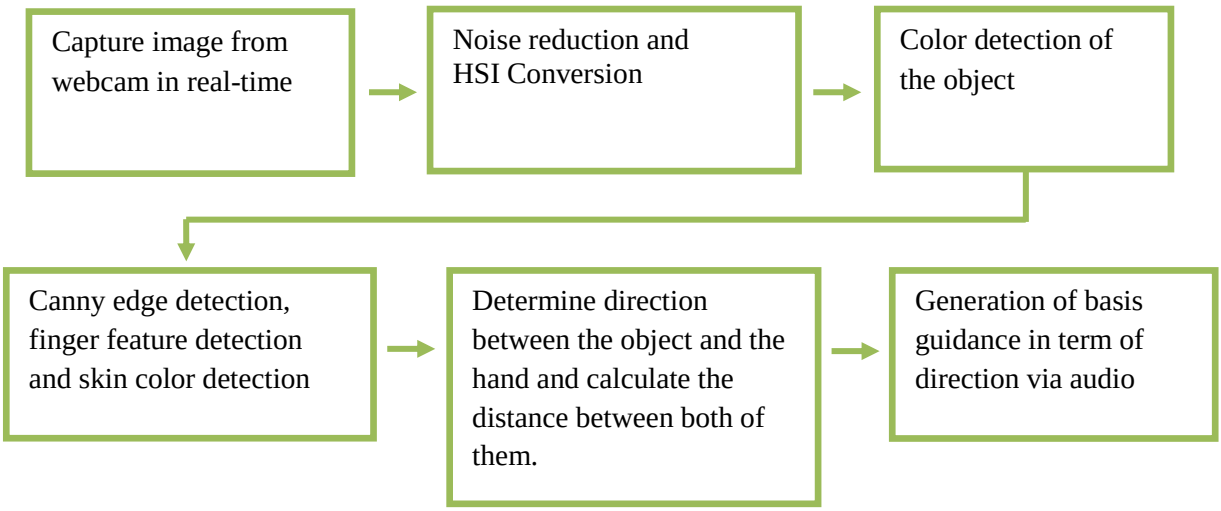


Figure 1-1: Block Diagram of HCI

At initial stage, the image is captured by using a webcam which is equipped along HCI. In pre-requisite, noise reduction is applied to smooth the image and then the image will be converted from RGB color space to HSI color space. Due to the time constraints, certain aspect will not be covered such as object pattern recognition. We simplified the object detection problem by just identify what constitutes as an "object" based on the tracking color instead of their shape and features. Hence all the objects are assumed to be in different colors. After the object is found inside the dynamic background, we subsequently determine if the hand is supposed to present inside the

dynamic background. To generate a robust hand detector, we proposed to apply composition of several scale invariant features for hand detection. In this paper, two primitive features which are fingertip and a pair of two nearly symmetrical parallel lines are used to form a finger feature. If more than two fingers are detected successfully in the scene, the hand is supposed to present inside the observed scene. Hence each finger will be marked as a candidate and delivered as a region of interest (ROI). For more details, our proposed approach for the hand detection will be further described in chapter 4. Since both the hand and the object had been found, HCI will provide the guidance in term of the direction via audio device to tell the user to perform the correct action. As noted that, the output of the detected object and the detected hand are represented as the region of interest (ROI). Therefore if both of their ROI are overlapping, HCI will stop the direction determination and tell the user to pick up the object immediately. Otherwise, it will keep determine the direction and tell the user to move his hand based on the given direction in order to localize the position of the object finally.

1.4 Project Objectives

The objectives of our project include as follow:

- Develop a user-friendly wearable human computer interface.
- Hand detection is implemented by identifying its fingers.
- Maximize the accuracy of hand detection under the cluttered background and good light conditions which are ambient light and indoor light.
- Maximize the accuracy of colored objects detection in different level of HSI.
- Implement an algorithm to calculate the direction and distance between the moving hand and targeted object in x-y plane.

- Place the hand inside the bounding region of the targeted object based on the given direction.
- Give correct guidance to the user via audio device.

1.5 Input Data

Input data for our HCI system is the image captured by a webcam device. In our implementation, bitmap image is used to represent the input image. Bitmap image are made of individual dots called pixels that are arranged and colored differently to form a pattern. To prevent high computational cost and increase the algorithm performance, the input image with high resolution is not recommended and supported considerably instead of 160 x 120 pixels. Apart from that, our HCI should work in any scene that the intensity of the environment is high enough so that the color of the object could be detected as well as its shape is clear. By doing that, it ensures that the input image is not distorted by light intensity. Figure 1-2 shows the examples of the images with different light intensities and its color is distorted by the change of the light intensity.

(a) High intensity **(b) Good intensity** **(c) Low intensity**

Figure 1-2: Same images with different light intensities.

CHAPTER 2 LITERATURE REVIEW

The ability to detect human's unconstrained hand in unknown environment is the most important in our HCI. The result of the hand detection usually affects how well our HCI can work in the real time. Therefore the present chapter focuses to study the relevant literatures. Since a lot of the different hand detection approaches had been proposed for both detecting a hand and its corresponding gesture efficiently and robustly without constraints upon either user or environment, different visual clues which are skin color, shape, unique features and other else, had been applied as a way to overcome the issues of the hand detection.

2.1 Viola-Jones Detection

Viola and Jones proposed a rapid object detection approach using boosted cascade of simple features (Paul Viola & Michael J. Jones 2001). Their concept "Integral Image" is introduced to allow the features used by the detector to be computed very quickly. Integral Image can achieve scale invariance by eliminating the need to compute a multi-scale image pyramid and significantly reduces the image processing time. Secondly, a simple and efficient classifier which is built by using the AdaBoost learning algorithm to select a small set of critical features from large majority of potential features. Thirdly, it is successfully combined in form of cascade to focus more attention on promising regions of the image. Their framework had been further extended by Lienhart and Maydt (Rainer Lienhart & Jochen Maydt 2002) which enriches the feature pool by adding a novel set of 45 degree rotated haar-like features (Paul Viola & Michael J. Jones 2004). These additional features significantly enhanced the expressional power of the learning framework and consequently improve the performance of the object detection system. On the other hand, Kolsh and

Turk (M. Kolsh & M. Turk 2004) exploited the limitation of hand detection by using Viola-Jones detector. Training with this approach significantly leads to expensive computational cost and complexity of the computation.

2.2 Hand Posture Recognition Using Adaboost with SIFT

Chieh and Ko (Chieh-Chih Wang & Ko-Chih Wang 2008) proposed to apply the discrete Adaboost learning algorithm with Lowe's scale invariant feature transform (SIFT) features (D. G. Lowe 1999) for hand posture recognition. SIFT features are invariant for their rotation and scale. This characteristic of SIFT allow accomplishing multi-view detection straightforwardly. For learning solution, Discrete Adaboost learning algorithm had been applied to combine many weak classifiers linearly to produce a strong classifier. In their approach, the SIFT features are extracted from the tested image at the initial stage. For each weak classifier, the distance between its associated SIFT features and the extracted SIFT features from the tested image are computed. The best match with the shortest distance which is shorter than a threshold is treated as the valid result from this weak classifier. Then the weight factors of all valid weak classifiers are summed. If this summed value is greater than the threshold of the strong classifier, the tested image is classified as a hand image. For the classification of the hand gesture, they apply a sharing feature concept proposed by Torralba et al. (A. Torrabla, K. Murphy & W. Freeman 2006) to increase the accuracy of multi-class hand gesture recognition. Based on their given result, their proposed approach is quite robust to deal with the background noise issues. However, it involves of the data learning mechanism. More the data it trains, more expensive computational cost it needs.

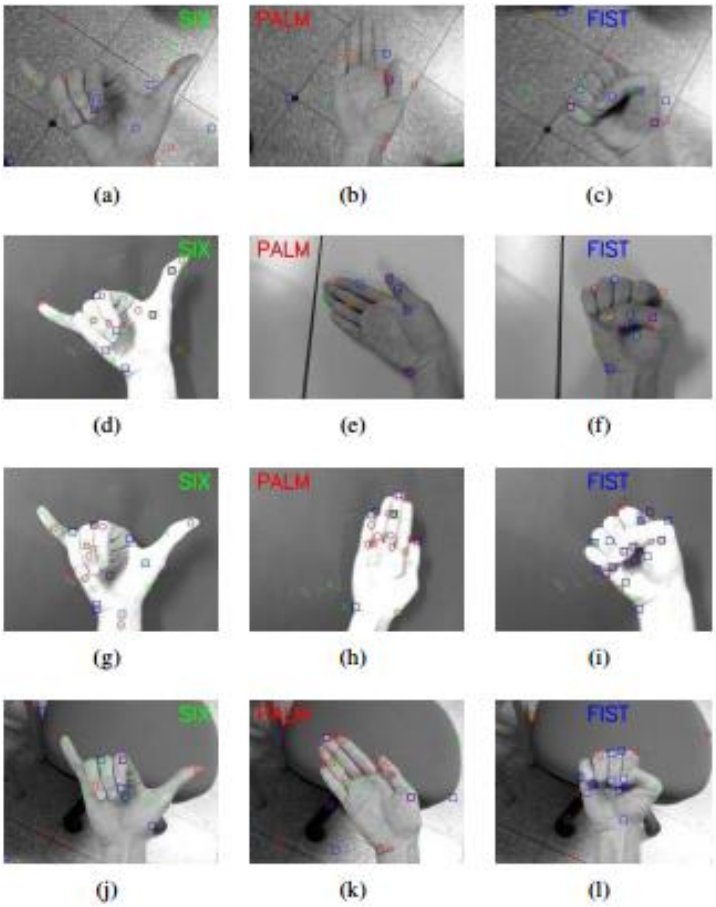


Figure 2-1: Correct hand gestures recognition results obtained by using Adaboost with SIFT. (Chieh-Chih Wang & Ko-Chih Wang 2008)

2.3 Simple Color Classification using Randomized Lists

A hand tracking method based on a simple color classification was presented in the paper "Robust hand tracking using a simple color classification technique" (Miaolong Yuan et al. 2008). This method involves of two procedures: training and tracking. L*a*b color space is used for the training and tracking. In the training procedure, the user specifies a region on a hand to obtain the training data to generate the color classifier. The classifier is stored into a randomized lists data structure. In the tracking procedure, the hand will be segmented from the background by using the randomized lists which had been done during the training. This method is based on clustering similar color pixels and defining a region of interest. Based on the tracking results which the paper presented, their proposed approach is robust against various luminance conditions but its shortage is hard to distinguish the hand from other skin-colored object in the background.

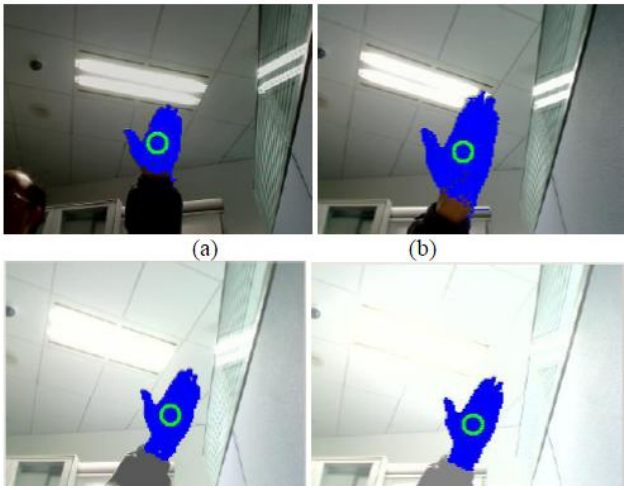
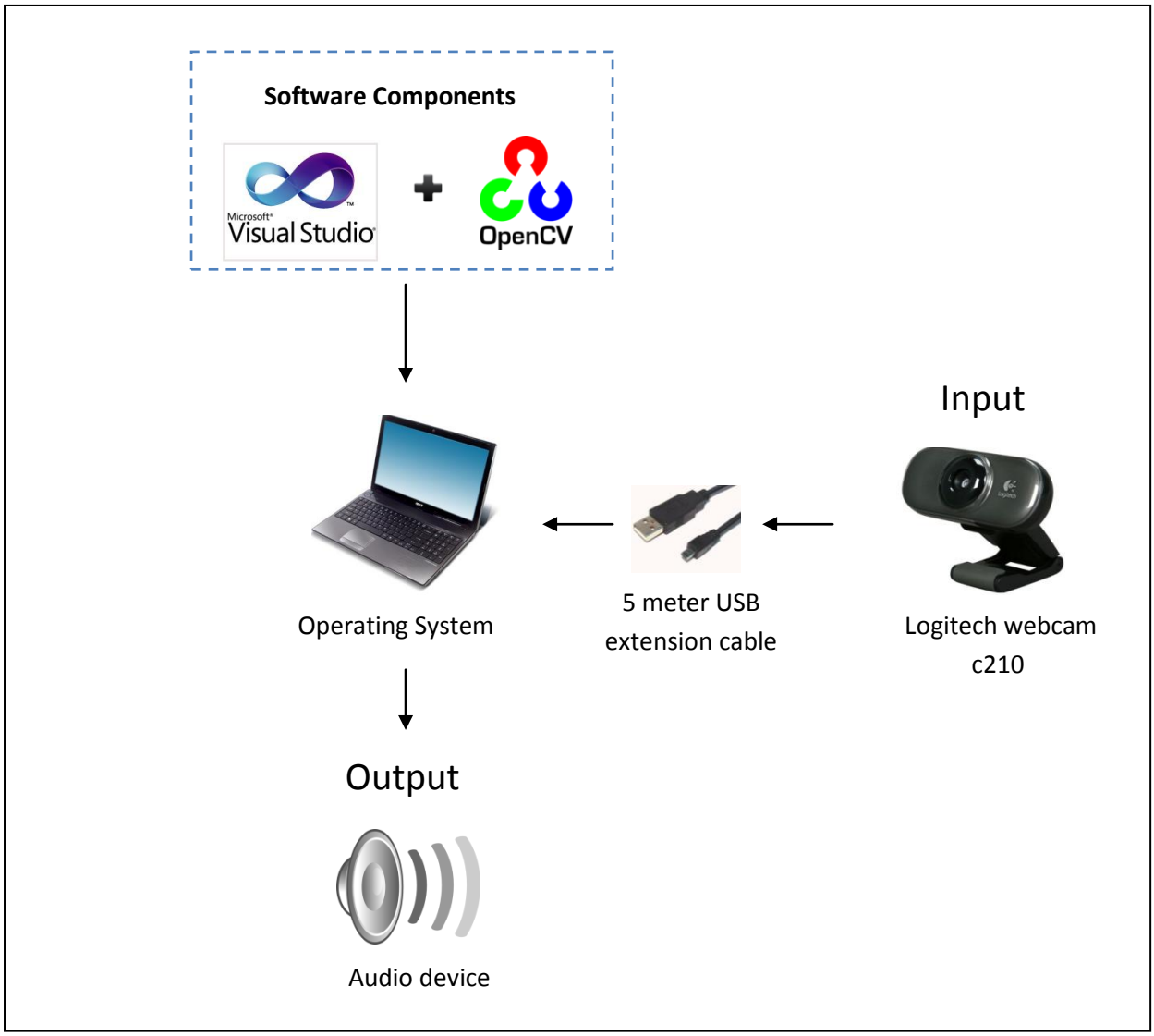


Figure 2-2: Hand tracking by clustering the similar color pixels. (Miaolong Yuan et al. 2008)

CHAPTER 3 SYSTEM OVERVIEW

This chapter introduces the architecture of HCI system. The overall system can be made up by several components which can be classified as the hardware and software as shown in figure 3-1.

Figure 3-1: Components diagram for the whole system.



3.1 Hardware Specification

The webcam used in HCI is Logitech webcam c210. The following table lists down the specification of this webcam.

Photo Quality	1.3 Megapixel
Interface	USB 2.0
Max Resolution	640 x 480
Supporting	Windows XP (SP2 or higher), Windows Vista or Windows 7 (32 or 64-bit)

The following is a list of specification of the laptop used throughout the whole project.

Processor	Intel® Core™ i3 CPU M330 @ 2.13GHz
RAM	2.00GB
Operating System	Windows 7 Home Premium 64-bit
Graphic card	NVIDIA® GeForce®310M
Display	14.0 inch 1366 x 786 pixels (Max)
Battery	6-cell Li-ion battery
USB port	Yes

There is a clear correlation between the hardware performance and the system processing speed. The higher performance the hardware components have, the faster the system can be executed.

3.2 Software Specification

The system built is running at window operating system. It is designed by using Microsoft Visual Studio with C++ programming Language. Microsoft Visual Studio is a powerful IDE (Integrated Development Environment). UI widget it provides is very convenient for programmers because the UI items from the UI widget can be generated into the code automatically once we had dragged them into the project. It gives a realistic and quick view to the current project UI interface. Besides that, the IDE contains keyword filtering function. It can filter code analysis result and give the warning whenever it finds out the syntax error.

OpenCV is well suited for the purpose of dealing with real-time computer vision project. OpenCV is a cross-platform library of programming functions mainly aims at real-time computer vision. The library was originally developed by Intel Corporation in 1999 as part of research initiative to advance CPU-intensive application, and is currently continued by Willow Garage, a robotics research lab and technology incubator as of 2006. It contains many features such as facial recognition, gesture recognition, image segmentation, motion tracking and more. Currently, it can support Windows, Linux, Android and Mac OS platform.

3.3 Assumptions

There are several assumptions which should be made up to ensure that the system can be operated properly. It covers:

- Should keep certain distance between the camera view and hand not so close.
- Our hand detector inside HCI currently works for opening palm gesture.
- Each of the user‘s hands should have five fingers.
- The HCI system should be running under the environment with the ambient light and good indoor light condition.
- The specific colored object in observed scene is only one and in single color, static and big enough.
- The HCI is able to be worn by the user but its mobility is limited by some activities. Running, jumping and other else can cause the failure of the hand tracking. These activities are avoided as well as possible.

3.4 System Flow Diagram

Figure 3-2: The flow chart of the proposed HCI system.

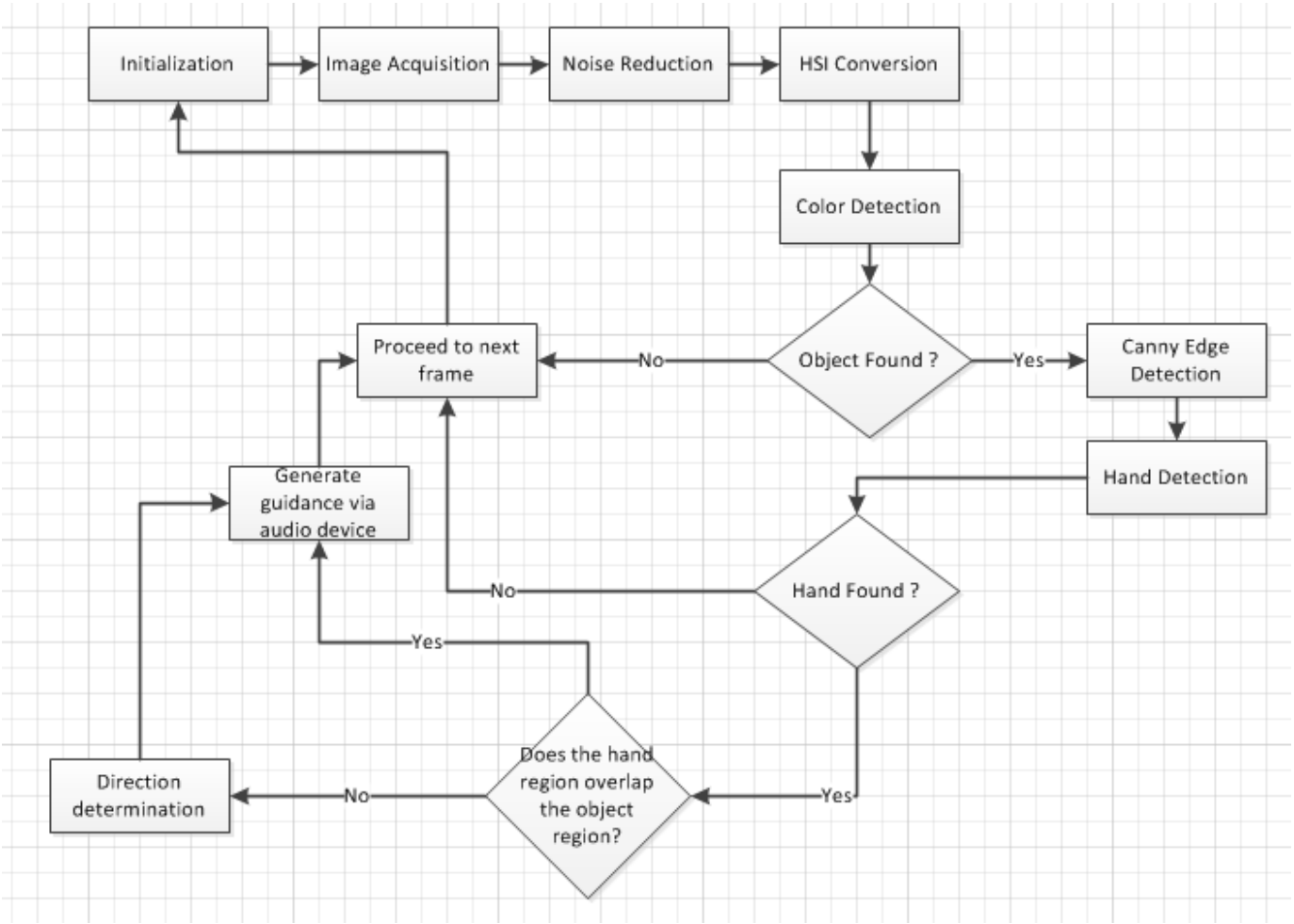
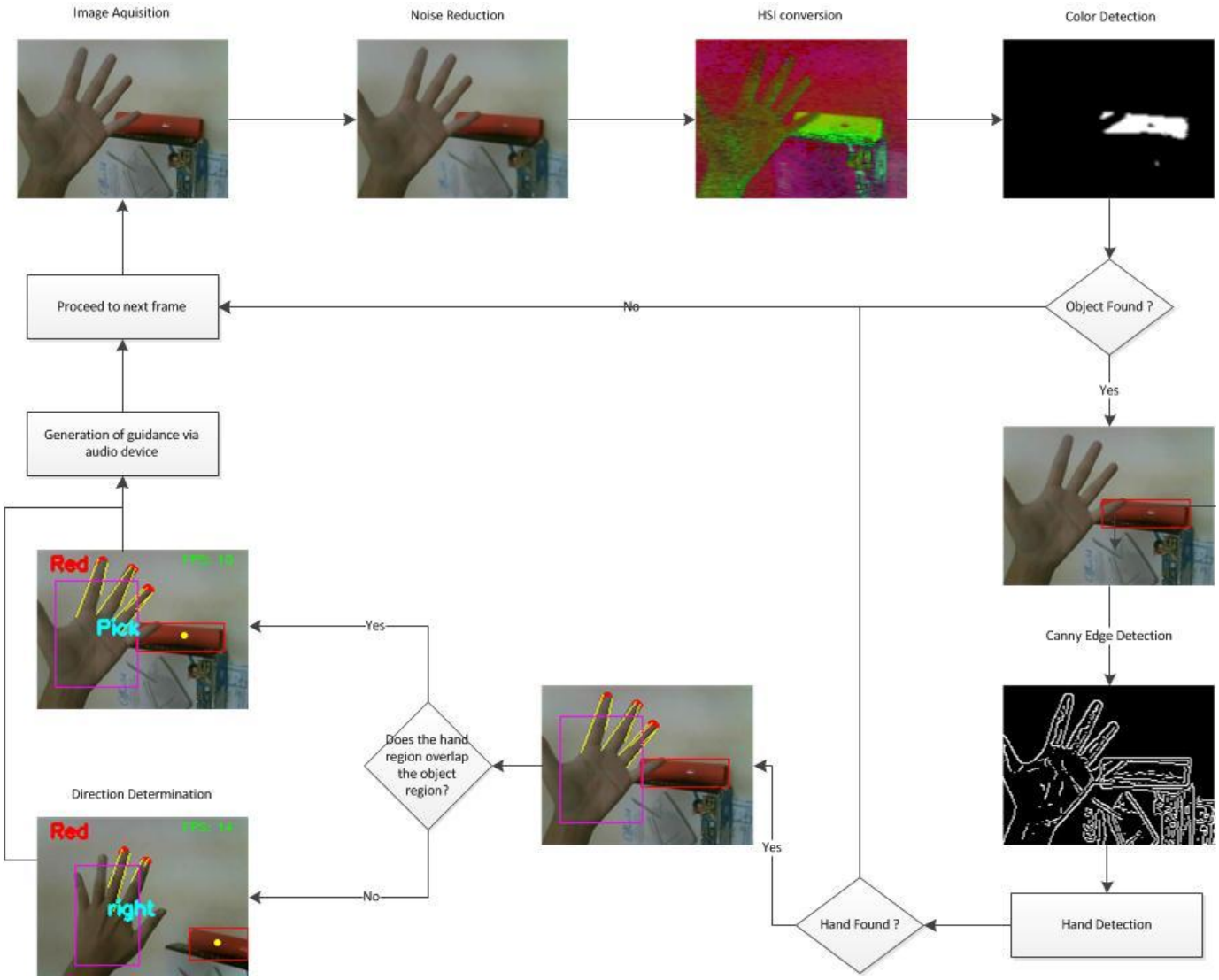


Figure 3-3 Example of input and output images for each step in HCI system



CHAPTER 4 METHODOLOGY

4.1 Hand Detection

Our hand detector is built by identifying a 5-finger pattern with scale-invariant features. This approach applies the analysis of the informative edge features of the fingers. The characteristic of the scale-invariant features allows us achieving successful multi-view detection in such a way that the different responses at different scales are same. The criteria to be a finger are:

1. A finger has a curved fingertip.
2. A finger is composed by two nearly symmetrical parallel lines.

4.1.1 Primitive Feature I: Two nearly symmetrical parallel lines

Our approach makes use of the Canny edge information to extract the scale invariant features from each of the contour in the binary image. In pre-requisite step, all the contours will be computed and queued after the transformation of the binary image by using Canny edge detection and then more image preprocessing techniques will be applied for contour refinement in order to achieve the elimination of the small blobs. Afterward, we examine each of the contours to find all the nearly straight lines within its bounding region by using Hough Transform. The Hough transform was first introduced as a useful method of detecting complex patterns of points in binary image. Firstly, we will discuss briefly how the Hough Transform does well in line detection. There are two general parameterizations which are the normal form and the slope-intercept form to be line representation. In the slope-intercept form, *m* can be large for nearly vertical line so there is no representation for vertical line. Hence we apply the

normal form to solve the issue above. The equation of normal form is expressed as below.

$$p = x_1 \cos\theta + y_1 \sin\theta$$

where (x_1, y_1) is a point along the line, θ is the angle this normal makes with the x axis, p is the length of the normal from the origin (0, 0) onto the line. In Hough Transform algorithm, it loops thought all the pixels inside the binary image. Each bright pixel is labeled as an edge candidate. For each edge candidate, we get a sinusoid by using θ = -90 and calculate the corresponding p value. Subsequently, we vote it in the accumulator cell in a plane θ, p by increasing the value of this cell by 1. Then we take the next θ value and calculate the next p value. This process is iterated until θ = +90. If the curves of two different points intersect in the plane θ, p, it means that both of the points are belong to a same line. Apart from that, all the shorter lines which length is less than Euclidean distance 5 are considered to be filtered out from the line collection. To achieve the in-plane rotation invariant characteristic, different orientations which start from -70° to 70° will be trained during the line detection process, as shown in figure 4-1.

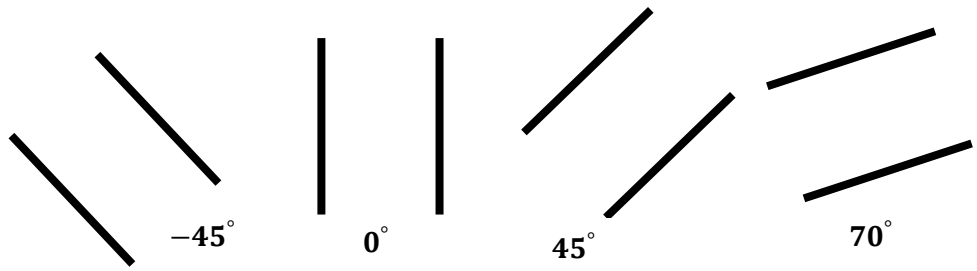


Figure 4-1: Line feature at different orientations

Basically, a pair of parallel lines is formed when both of their gradients are approximately same so we measure the gradient of each line and compare it with each others. If both of their gradients are same, we will proceed to next phase to determine

if both of the parallel lines are symmetric. To identify symmetrical parallel lines, each point at line A is drawn to construct a perpendicular line. Given a point $P(a, b)$ at line and known gradient of the line m, the equation of the perpendicular line can be formulated as below.

$$m_p = -1/m$$
$$c_p = b - m_p a$$
$$y = m_p x + c_p$$

where (x, y) is a point along the perpendicular line p, m_p is gradient of the perpendicular line p and c_p is a constant. If one of the points $P(x, y)$ at line B intersects at the perpendicular line which formed by one of the points $P(a, b)$ along line A, the weight w will be increased by one. This process is iterated until reaches the last point at line A. Since w exceeds three quarter of the length of the line A, both of the lines are treated as a pair of the symmetrical parallel lines and stored for the later grouping process. Figure 4-2 clearly illustrates how to identify a pair of the symmetrical parallel lines by using our proposed approach.

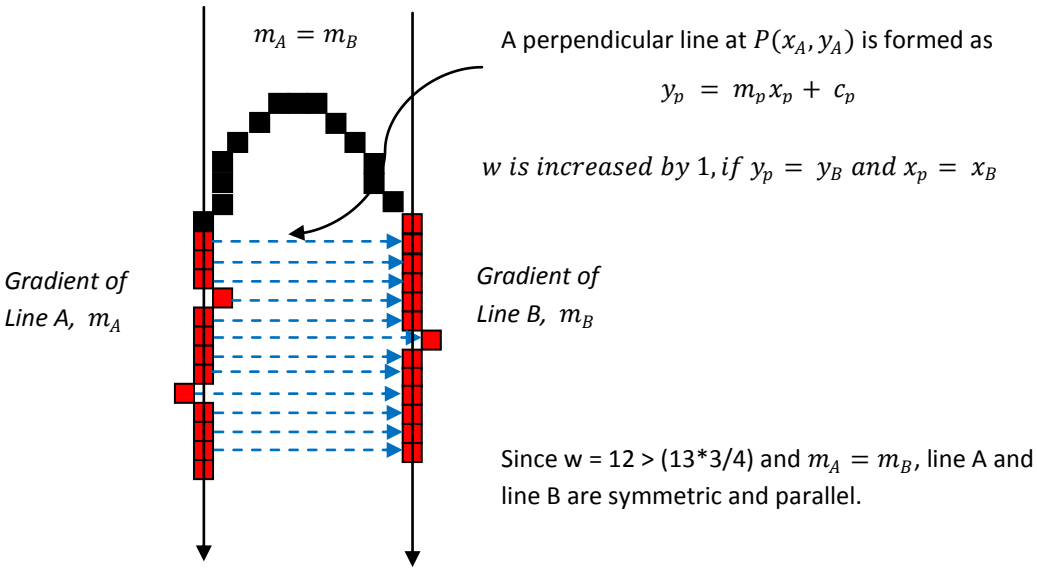


Figure 4-2: Extraction of a pair of symmetrical and parallel lines

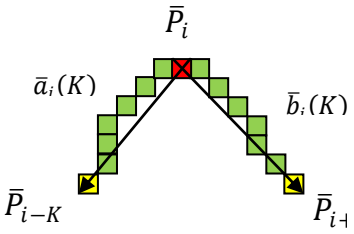
4.1.2 Primitive Feature II: Fingertip

After finding the symmetrical parallel lines of the finger, we implement an algorithm to detect the fingertip. In our algorithm, we only consider the curvature which orientation is upward in order to represent as a fingertip. Otherwise the other oriented curvature is not in our consideration. Suppose an edge contour is defined by C= $\{P_i \mid i = 1, 2, 3, \dots, m\}$, where P_i is the i^{th} boundary point of the edge contour, we use K control parameter to form a sequence of points to represent as a curvature. Firstly, we determine the suitable value to the K parameter. The definition of K respects to the arc length of the curvature. This varying value could work well in some situations, but it could also lead to the failure in some cases of scale variants. For instances, a small K makes the curvature noisy owing to digitization, whereas a large value loses the relevant information of points within the K-pixel area. Hence the K value is considered to be assigned based on the maximal width between two nearly parallel lines of the finger. Next, forming a sequence of points in such a way that $S_{P_i} = \{P_{K-i}, P_{K-i+1}, P_{K-i+2}, \dots, P_i, P_{i+1}, P_{i+2}, \dots, P_{K+i}\}$ where K is constant, P_i is i^{th} point of the edge contour C and S_{P_i} is the sequence formed at P_i. Subsequently, a formula which is used to check the each sequence S_{P_i} in order to detect the sharp turn of the curvature is defined as below.

$$c_i(K) = cos\theta_i = \frac{\bar{a}_i(K) \cdot \bar{b}_i(K)}{\|\bar{a}_i(K)\| \cdot \|\bar{b}_i(K)\|}$$

where $\bar{a}_i(K) = \bar{P}_{K-i} - \bar{P}_i$, $\bar{b}_i(K) = \bar{P}_{K+i} - \bar{P}_i$, θ denotes the angle between $\bar{a}_i(K)$ and $\bar{b}_i(K)$, $K \in N$. As $c_i(K)$ value is approximately 0, it indicates that a sharp turn between P_{K-i} and P_{K+i} is formed at P_i, implying that P_i is an indicator of the change in the curvature of the edge contour. After the successful detection of the curvature, we considerably determine if this curvature is upward-oriented. When a

sharp turn is formed at P_i, the y-coordinate of all the points in the sequence S_{P_i} except P_i will be checked and compared with the y-coordinate of P_i. If the y–coordinate of all the points in the sequence S_{P_i} are greater than the y-coordinate of P_i, this sequence will be treated as a upward-oriented curvature.



K = 7

$$\boldsymbol{S_{P_i} = \{ P_{K-i}, P_{K-i+1}, P_{K-i+2}, \dots, P_i, P_{i+1} \dots, P_{K+i}\}}$$

Figure 4-3: Upward oriented curvature measurement

4.1.3 Grouping the primitive features

In order to constitute a finger feature, we group them by connecting their boundary points of the contour in order. As mentioned above, an edge contour is defined by C= $\{P_i \mid i = 1, 2, 3, \dots, m\}$, assume there are N pairs of detected symmetrical parallel lines and M detected upward oriented curvature, we initially select one pair of symmetrical parallel lines and one of the upward oriented curvature to merge them together. If they are enclosed without have any gap, it obviously indicates that this pair of the symmetrical parallel lines intersects the fingertip. Therefore we mark this group as a finger candidate and store it in a linked list. The contour which contains at least two finger candidates is chosen for further analysis process. To further improve the accuracy of the detector, every candidate will be evaluated by employing the skin color to check its distribution probabilities of the skin pixel. Once they have past the skin color evaluation, they are treated as the promising region of the hand and will be

enclosed by a bounding rectangle, called as region of interest (ROI). The skin color detection will be discussed in next section.

4.1.4 Skin Color Detection

The previous work done by identifying the finger pattern doesn't ensure that all the results after the training had been satisfied by our expectation. To evaluate the accuracy for the hand detection, we eliminate the incorrect results by estimating the distribution probabilities of the skin pixel candidates within the promising region. The goal by doing that is to reduce the false positives rate and ensure that our HCI system can work successfully during the hand tracking. It is a challenging task while the human skin color is adapted under various illumination conditions.

Several color spaces had been proposed in the literature for skin-tones color detection. In our implementation, we proposed to use the HSI color space for skin-tones color detection. The reason behind it is HSI is more relative to the human perception. After the conversion of the color space from RGB to HSI, every pixel of the image is classified into skin-tone pixel or non skin-tone pixel by color threshold. Each skin-tone pixel is determined by measurement of its H channel value and S channel value. For Asian, the skin-tone in H channel is typically characterized with the value ranging from 0 to 50 and S channel is characterized with the value ranging from 0.23 to 0.63. If that pixel candidate is considered to be skin-tone pixel, the average gray value of that pixel will be linearly scaled to 255 and otherwise it will be scaled to 0.

Afterward we calculate distribution probability of the skin pixels to determine if the observed hand appears inside the scene. The distribution probability of the skin pixels is calculated by the parameterized equation below

$$D = \frac{\sum_{i=1}^{n} \mu(p_i)}{n} \quad if\ \mu(p_i) = \begin{cases} 1 & if\ p_i is\ considered\ as\ skin\ tone\ pixel \\ 0 & if\ p_i\ is\ considered\ as\ non\ skin\ tone\ pixel \end{cases}$$

where D is defined as the distribution probability of the skin-tones, n represents as the total number of the pixels within the boundary of the ROI, p_i is defined as the i^{th} pixel. If D probability exceeds threshold T, there has been an increasing true positives rate to verify that the observed hand is present inside the scene. Otherwise, it implies that the results obtained by previous work are considered to be invalid. To determine a suitable T value, we considerably assign T probability to be 0.5, implying that the amount of the skin pixels should reach over the half of the overall pixels we tested within the promising region.

4.2 Color-based Object Detection

The color of the object can vary over the time dependent on different illuminations. To handle this appearance, the color model must be adapted in a good lighting observed scene. For practical, our HCI should work with having greater result under the ambient light and indoor light. In our implementation, there are four colors (red, blue, yellow and green) which are initially set to be tested during the color detection. The purpose of performing color threshold is to simplify the process of identifying what constitutes as an "object". Color of the object is filtered based on adjusting the best tolerance ranging of each variable in the HSI color model. There are three types of variables which need to be tested, which are hue, saturation and intensity.

After performing color threshold to segment the object from the image, the segmentation process usually result in unwanted noise. The tiny pixels forming a

small blob normally arise after the segmentation. To approximately preserve area and eliminate the small holes, morphology close are used to fill the gaps found within the blobs. This kind of the morphology operation dilate the shape of the blob by using a 1x1 rectangular kernel and then erode the result by using a 1x1 rectangular kernel. This refinement should be done to ensure that those small noises are not considered to be a part of the object. Next, all the contours had been computed from the binary image by connecting and grouping the neighboring entries. The optimal contour is determined by selecting its maximal area among all the contours and another condition should be satisfied here is the ratio between the selected contour size and the whole image size cannot be less than 0.001. By doing that, only single object is selected for the later processing while there are multiple same colored objects appear simultaneously inside the observed scene.

4.3 Guidance

4.3.1 Direction Determination

Since the hand and targeted object are found, a set of the instructions has been given by the HCI system to aid the user with the correct direction to localize the object position. In this part, comparison between two angle α and β is introduced to determine the correct direction. The angle between two points is calculated by using the two-argument arctangent function.

$$\theta = atan\,2(x_{diff}, y_{diff}) \times \frac{180}{\pi}$$

where x_{diff} is the difference between the x-coordinate of object point and the x-coordinate of hand point , y_{diff} is the difference between the y-coordinate of object point and the y-coordinate of hand point, θ is defined as the angle between the hand

point and object point in the radian. As shown in figure 4-4, θ angle is obtained by using the two-argument arctangent function. Next, we determine the belonging sector to θ angle and then compute α angle and β angle respectively using the formula based on table 4-1. In figure 4-4, the value of α angle equals to the 30 degrees and we get β angle with 70 degrees based on the formula as shown in the table 4-1. In order to consider which axis (horizontal axis or vertical axis) is closer to the hand position, we compare α angle with β angle. The horizontal axis is represented as right or left direction and the vertical axis is represented as up or down direction. For the case in figure 4-4, as the value of α angle is smaller than the β angle, it means that the hand position is closer to the horizontal axis whereas the hand position is far away to the vertical axis. In our consideration, right direction is chosen instead of the down direction. Note that, right direction is defined as the user moves his hand to the right side. Table 4-1 lists down all the expected results obtained among different conditions and show the relationship between α angle and β angle. However, there are some problems which possibly arise during the object tracking in the real-time. If the size of object is quite small, the whole area of the object will be covered by the hand area. To prevent it, the HCI system is able to work properly on the assumption that the size of the object must be large enough to be navigated in the scene.

Figure 4-4: Direction determination.

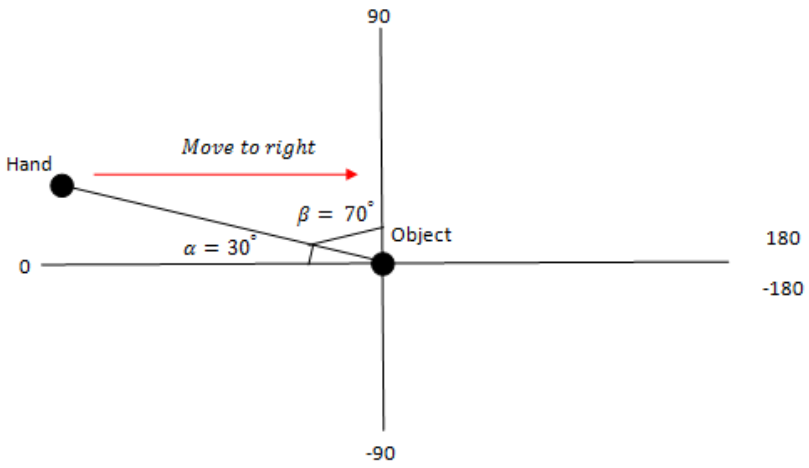


Table 4-1: Expected results among different conditions.

θ	α	β	Condition	Direction
$0 < \theta < 90$	θ	90 - α	$\alpha \geq \beta$	Down
$0 < \theta < 90$	θ	90 - α	$\alpha < \beta$	Right
$90 < \theta < 180$	θ-90	90 - α	$\alpha \geq \beta$	Left
$90 < \theta < 180$	θ-90	90 - α	$\alpha < \beta$	Down
$0 > \theta > -90$	$\lvert \theta \rvert$	90 - α	$\alpha \geq \beta$	Up
$0 > \theta > -90$	$\lvert \theta \rvert$	90 - α	$\alpha < \beta$	Right
$-90 > \theta > -180$	$\lvert \theta \rvert$-90	90 - α	$\alpha \geq \beta$	Left
$-90 > \theta > -180$	$\lvert \theta \rvert$-90	90 - α	$\alpha < \beta$	Up
$\theta = 0$	-	-	-	Right
$\theta = 90$	-	-	-	Down
$\theta = 180$	-	-	-	Left
$\theta = -90$	-	-	-	Up
$\theta = -180$	-	-	-	Left

4.3.2 Monitoring

The guidance process is monitored by estimating how closest the regions between the hand and the object. Given that the object region is defined as $R_{obj} = \{P_i | i = 1, 2, 3, \ldots \ldots\}$ and the hand region is defined as $R_{hand} = \{P_i | i = 1, 2, 3, \ldots \ldots\}$, as

$$R_{obj} \cap R_{hand} = \{\ \}, it\ means\ that\ they\ are\ separated.$$

$$R_{obj} \cap R_{hand} \neq \{\ \}, it\ means\ that\ they\ are\ intersected.$$

While both of them are located in such as way that some pixels of the hand region are also the pixel within the object region, it denotes that they are overlapping. Then our HCI system will terminate the guidance process and tell the user to pick up the object at that time. If they are not overlapping, our HCI system will keep giving the correct direction to the user.

CHAPTER 5 EXPERIMENT RESULT AND DISCUSSION

Our experiments were conducted in four aspects, which were hand detection performance, color-based object detection performance, guidance performance and system execution performance. In the practical, the wearable camera inside HCI is equipped by using c210 Logitech Webcam with a resolution 160 x 120.

5.1 Hand Detection Performance

A. Background

Figure 5-1 shows the results of hand detection by using our proposed approach. In figure 5-1(a, b), the results were obtained under the uniform light and uniform dark background. Against the uniform light and uniform dark background, our hand detector worked with a high performance, achieving a detection rate of 90.01%, with just 9.99% false positive rate according to the table 5-1. The detection rate is estimated with a collection of 25 samples which are randomly obtained upon different uniform background. As shown in table 5-1, we considered two cases for the false positive calculation in every sample. That is, suppose 5 fingers present in the scene, but there don't have any detected finger in the scene. For this case, false positive is directly assigned to be 1. Another case is after some fingers were successfully detected, we found that they also include some incorrect detected fingers. Hence the false positive in this sample is calculated as the number of incorrect detected fingers divided by the number of overall detected fingers in this sample. For the calculation of the total false positive rate, it is obtained as the sum of the false positive in every sample divided by the number of the samples and then multiplied by 100%.

The main reason which causes the false positive rate here is due to the edge distortion. As shown in figure 5-2, the results obtained under the cluttered background were not really encouraging. We found that a particular of edge information inside the contour are distorted by direction of the light intensity and projection. This made the pattern of the contour become not exactly similar to what‘s it supposed look like in the real-time. Therefore the edge distortion rapidly led to the failure of the hand tracking. As shown in figure 5-1(a, b) third column, having the high distribution probability of the detected skin blobs further eliminate the incorrect results as well as possible and usually implies that the hand was classified well and appeared correctly. Moreover, it obviously gave the fact that employing skin color, there had been a significantly increasing detection rate with low false positive rate. More experiment results are shown in Appendix A-3.

(a)

(b)

Figure 5-1: Correct hand detection results against (a) uniform light background and (b) uniform dark background.
(a, b 2th column) Edge result
(a, b 3th column) Skin color result

Figure 5-2: Unsuccessful hand detection due to the edge distortion.

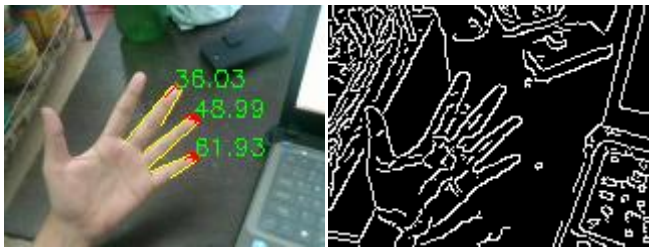
Sample	No. of finger in the scene	No. of detected finger with the skin color evaluation	No. of wrong detected finger	False Positive	Total time taken per frame (second)
1	5	2	0	0	0.03
2	5	2	0	0	0.03
3	5	3	0	0	0.06
4	5	3	0	0	0.04
5	5	2	0	0	0.03
6	5	2	0	0	0.10
7	5	3	0	0	0.12
8	5	2	0	0	0.03
9	5	2	0	0	0.05
10	5	0	0	1	0.12
11	5	3	1	0.3333	0.07
12	5	2	0	0	0.05
13	5	2	0	0	0.03
14	5	2	0	0	0.04
15	5	2	0	0	0.12
16	5	3	0	0	0.03
17	5	3	1	0.3333	0.07
18	5	2	0	0	0.09
19	5	4	2	0.5	0.03
20	5	2	0	0	0.03
21	5	2	0	0	0.06
22	5	3	1	0.3333	0.08
23	5	4	0	0	0.10
24	5	2	0	0	0.05
25	5	2	0	0	0.07
Detection Rate:		90.01%	False Positive Rate:		9.99%

Table 5-1: Detection rate and false positive rate obtained with a collection of 25 samples for the hand detection.

B. Scale and In-plane Rotation

The successful hand detections with the left hand and the right hand were demonstrated in figure 5-3. We respectively tested the hand detection at different orientations of in-plane rotations. In addition, we also tested the hand detection at different scales. It clearly shows that the characteristic of the scale invariant features which are upward-oriented curve and a pair of two nearly symmetrical parallel lines achieves the scale invariant and in-plane rotation invariant hand detection.

(a)



(b)

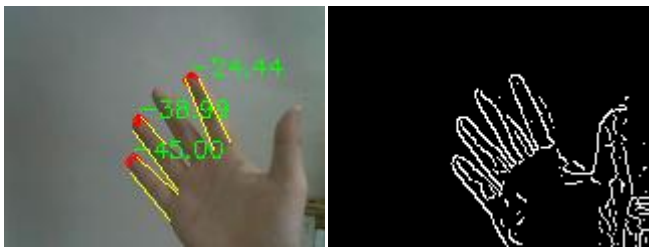


Figure 5-3: Scale invariant and in-plane rotation invariant hand detection.
(a) Left hand detection with successful detected fingers at orientation 36.03, 48.99, 61.93.
(b) Right hand detection with successful detected fingers at orientation -24.44, -38.99, -45.00.

5.2 Color-based Object Detection Performance

In color-based object detection, HSI color space was applied for the color threshold. Three variables which are hue, saturation and intensity were tested to determine their best tolerance ranging for the corresponding color. As shown in figure 5-4 (a), blue object in the observed scene was detected successfully by filtering off most of the non-blue pixels from the background. In binary image, most of the blue pixels were scaled to white and all the remaining pixels were scaled to black. We initially assigned the tolerance value of the hue variable by starting with 84 and slowly increase the tolerance value of the saturation with starting from 0 to determine if there is an increasing amount of the blue detected pixels. For the tolerance value of the intensity, we considerably assign its beginning value to be 50 instead of 0 because the object that may be distorted by low light intensity and become dark color is not in our consideration. In fact, various illuminations could largely affect the color appearance of the object, especially for the low light intensity and high light intensity. Thus our HCI system is designed on assumption that it could work fairly well as we predicted under the ambient light and a good indoor light which includes the situation when switch on the light and switch off the light in a room. Lastly, table 5-2 shows the entire information of each color which had been tested during the color classification.

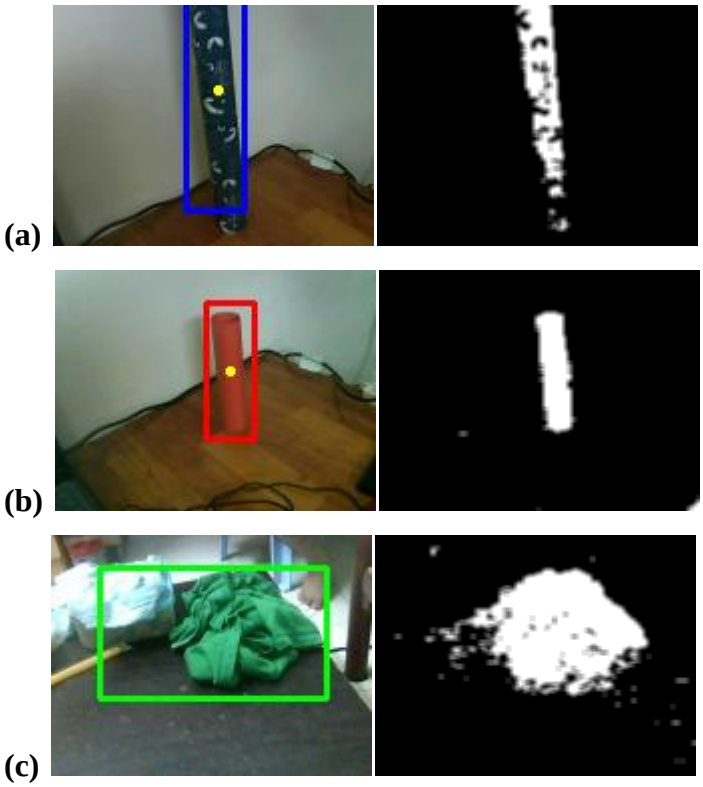


Figure 5-4: Correct color detection results for (a) blue colored object, (b) red colored object and (c) green colored object.

Table 5-2: Information for each tested color in the HSI color model

	Tolerance ranging of each variable in HSI color space.		
Color	**Hue**	**Saturation**	**Intensity**
Red	0 – 10	118 - 255	50 - 255
Blue	84 – 130	113 - 255	80 - 255
Green	53 - 75	74 - 255	80 - 255
Yellow	22 - 38	100 - 255	100 - 255

5.3 Guidance Performance

The HCI guidance process is the essential part of HCI. In order to ensure its reliability, several experiments were conducted to check the correctness of the direction with the given different hand locations and the different object locations. The experiment results were clearly shown in figure 5-5. In figure 5-5(a), as the hand appeared on the right side of the object in the scene, the guidance process correctly issued "right" command to tell the user to move his hand toward the right side, whereas in figure 5-5(b) both of the rectangles had been overlapped, HCI system displayed "Pick" and told the user to pick up it immediately.

(a)

(b)

(c)

Figure 5-5: The correct guidance was given based on different hand positions and different object positions.

5.4 System Execution Performance

To speed up the overall system performance and reduce the algorithm processing time, multi-threading technique is introduced and applied for the implementation of the HCI system. Multi-threading is an ability to manage the multiple tasks concurrently. In the hand detection algorithm, all the contours are needed to be checked for the scale invariant features extraction. As the number of edge detected in Canny increased, it linearly increases the amount of the contours especially for the hand detection under the cluttered background. In traditional way, each contour is checked one by one sequentially. It means that the next contour checking should be waiting as long as the current contour checking had not been accomplished, whereas by using multi-threading technique, both checking of the current contour and the next contour could be done simultaneously without the need of waiting. Table 5-3 obviously points out the effectiveness of using multithreading technique.

Table 5-3: The time taken for the hand detection algorithm with and without using the multi-threading technique.

Sample No	Contour size	Total pixel amount of all the contours	Time taken for Sequential checking (second)	Time taken for checking in parallelism with using the multithreading (second)
1	63	1460	0.03	0.03
2	165	4133	0.06	0.04
3	193	4793	0.07	0.05
4	216	5215	0.09	0.07
5	434	7277	0.13	0.10

CHAPTER 6 Future Works

To generate a well hand detector which is robust against the different environments, our future work will concentrate on improving the accuracy for detecting and classifying hand gestures in environments with more cluster and variability. According to our results, we found that variable illuminations, cluttered background both of the factors largely decrease the accuracy of the hand detector. In this project, two primitive classes which are upward-oriented curve and a pair of two nearly symmetrical parallel lines were grouped into a finger feature. However, it is still insufficient to detect the hand in the real-time scene. It will significantly increase the false positive rate if the scene contains so much the object which pattern is similar but doesn't correspond to the human finger. In order to yield satisfactory result, more scale invariant features are considerably extracted for more training as well.

On the other hand, more functionality can be added to our HCI system. Our HCI system currently works at two dimensions images without consider the vision-based depth cue. In order to measure the movement of the user's hand forward to the object in the 3D scene, depth estimation can be computed via stereo vision, in which images from two cameras are captured. Another improvement could be done is object recognition which enables user to track the desired object with multiple clues, not only depend on the color. Furthermore, an accident possibly occurs during object tracking due to the insufficient of perceiving the visual information surrounding environment. To prevent it, obstacle avoidance is considerably helpful to the user. The goal is to aid the user attempting to avoid any obstacles in front of the user and find a correct way until reach the object position.

Apart from that, another improvement could be done is the hand gesture recognition. It is useful tool for the user to communicate with the HCI system without the need of the interface control. Each hand gesture can be respectively expressed as a vision-based command to tell the HCI system to perform the corresponding task. For instance, the fist is used to call the HCI to stop the tracking process immediately. Compare to the hand gesture recognition, the speech recognition is designed to issues a voice command though microphone. Microsoft Windows Vista and Windows 7 provide Speech Development Kit for developer as well as a speech engine which is shipped with the windows. Employing the human voice, the user is able to change the current chosen color into any color of the color collection by just saying “red”, “blue” or other else.

CHAPTER 7 CONCLUSION

5.1 Conclusion

In this paper, an assistive human computer interface is demonstrated to localize the specific colored object by hand tracking guidance in the real-time. Our HCI was developed to work simultaneously with a normal camera, which is able to be worn by user and carried freely. The design of HCI centralized around the interest of the visual impaired. The HCI system provides the correct guidance to the user via audio device while they would like to pick up a particular colored object. As the major contribution of HCI, it eventually improves the quality of the visual impaired life.

To eliminate the visual disability of the visual impaired, a vision-based hand detection system is implemented to allow the visually impaired interacts with the HCI in order to perform the corresponding task. The hand detector is accomplished by identifying finger pattern with scale invariance features which the features we used in this project are upward-oriented curve and two nearly symmetrical parallel lines. The accuracy of the hand detector is further improved by employing the skin color of the hand. The results we discussed in chapter 5 obviously pointed out the effectiveness of our proposed approach for solving hand detection problem. Our proposed approach applies composition of several scale invariant features instead of large training data. Hence our approach achieves low cost computation comparing to the previous work done by Viola- Jones detector. Besides, our approach is robust against different orientations and scales. It means that no matter the left hand or the right hand the user employs, our hand detector still can work fairly well. There are some limitations which largely reduce reliability of the hand detector are needed to be covered in the future work. One of the limitations is poor illumination which greatly increases the

difficulty of the hand detection problem especially for low light intensity usually results in edge distortion with the impulsive noise. Our proposed approach is dependent on the edge information. If the hand contour extracted from the binary image doesn't similar to the original image due to the edge distortion, it will rapidly lead to the failure of the hand detection.

Due to the time constraint for this research work, a wide of relevant areas have not been explored. The first thing should be ensured here is our HCI system should work safely and with high reliability. In the future work, our HCI system is considered to be improved by extending more functionality to make HCI become more feasible in the real time. For instances, object recognition enables the user to determine what kinds of the object it is with multiple cues, not only depend on the color. Also, hand gesture recognition is very essential part in HCI because each hand gesture can be expressed as a meaningful order to interact with HCI for performing the specific task respectively, such as "switch color", "stop the tracking" and so on. Finally, our HCI system will be ideally implemented into a microcontroller along with the wireless signal device, and a wide angle hidden camera will be equipped inside the HCI for having the wide field of the view.

BIBLIOGRAPHY

A. Torrabla, K. Murphy & W. Freeman, 2006, 'Sharing features: efficient boosting procedures for multiclass object detection', in *IEEE Conference on Automatic Face and Gesture Recognition (AFGR)*.

Chieh-Chih Wang & Ko-Chih Wang, 2008, 'Hand posture recognition using Adaboost with SIFT for human robot interaction', in *Springer Lecture Notes in Control and Information Sciences*, vol. 370, pp. 317-329.

D. G. Lowe, 1999, 'Object recognition from local scale-invariant features', in *International Conference on Computer Vision*.

Miaolong Yuan, Farzam Farbiz, Corey Mason Manders & Ka Yin Tang, 2008, 'Robust hand tracking using a simple color classification technique', in *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, pages 6:1-6:5.

M. Kolsh & M. Turk, 2004, 'Robust hand detection', in *IEEE International Conference on Automatic Face and Gesture Recognition*.

Paul Viola & Michael J. Jones, 2001, 'Rapid object detection using a boosted cascade of simple features', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

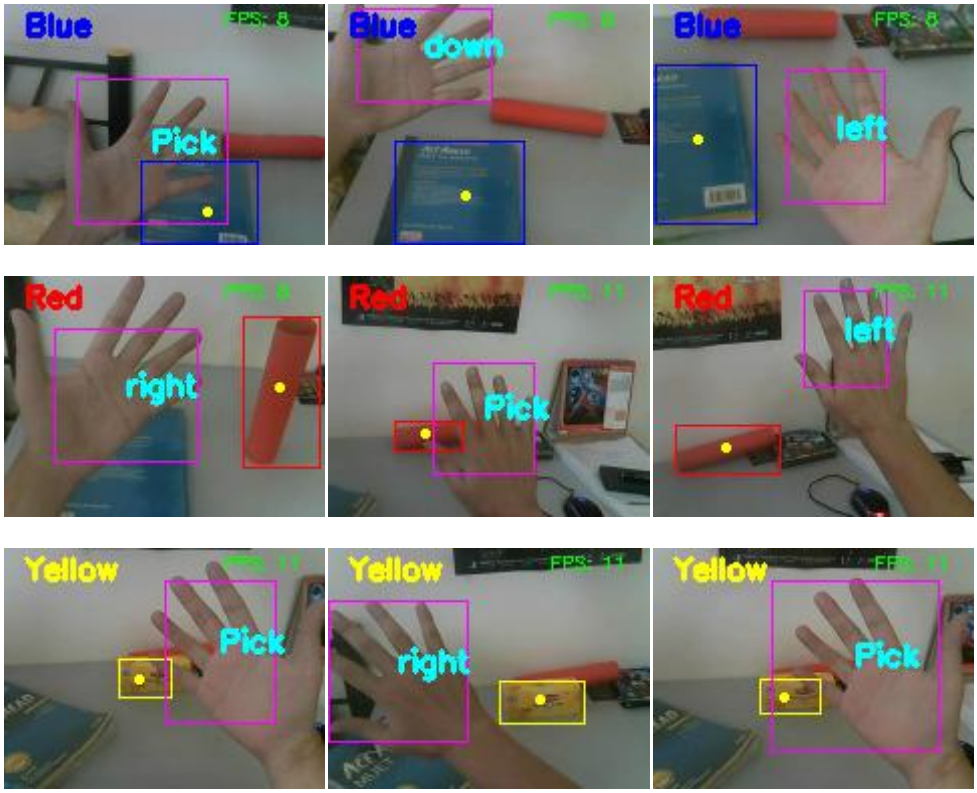
Paul Viola & Michael J.Jones, 2004, 'Roubust real-time face detection', in *International Journal of Computer Vision*, vol. 57(2), pp. 137-154.

Rainer Lienhart & Jochen Maydt, 2002, ‘An extended set of Haar-like features for rapid object detection’, in *IEEE International Conference on Image Processing (ICIP)*, Vol. 1, pp. 900-903.

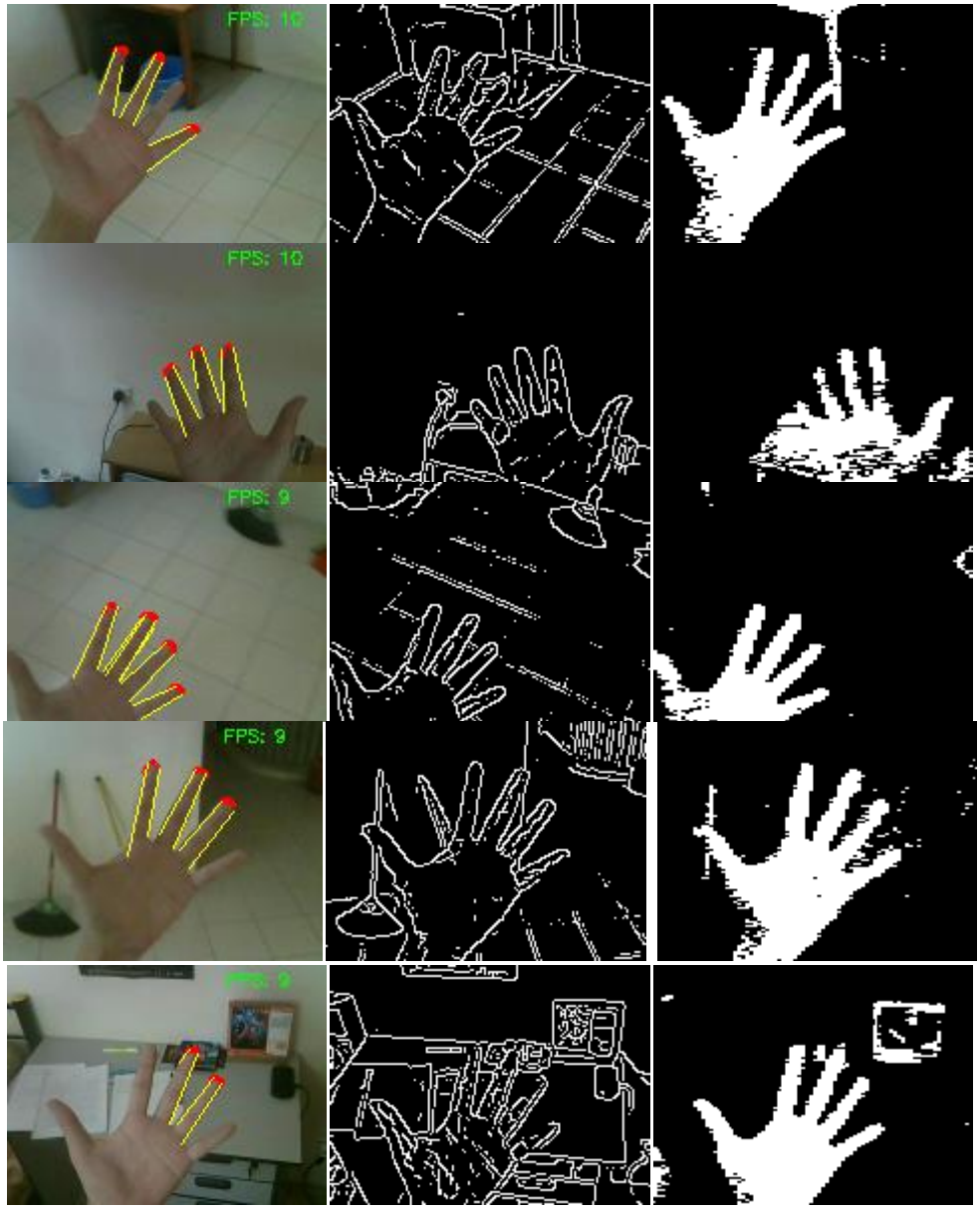
APPENDIX A SAMPLE IMAGES FOR HCI

A-1 Overall HCI performance under the Scene A

A-2 Overall HCI performance under the Scene B



A-3 Sample results of the hand detection



APPENDIX B POSTER