A PARTITIONING METHOD IN NOISE REDUCTION AND A HYBRID CONVEX HULL ALGORITHM FOR FINGERTIPS DETECTION ON AN OUTSTRETCHED HAND

WOUN BO SHEN

UNIVERSITI SAINS MALAYSIA

2016

A PARTITIONING METHOD IN NOISE REDUCTION AND A HYBRID CONVEX HULL ALGORITHM FOR FINGERTIPS DETECTION ON AN OUTSTRETCHED HAND

by

WOUN BO SHEN

**Thesis submitted in fulfillment of the requirements
for the degree of
Master of Science**

NOVEMBER 2016

ACKNOWLEDGEMENT

First for all, I would like to thank my main supervisor, Ms Tan Guat Yew for the valuable guidance, advice and support during the research. Besides, her motivation and patience are appreciated. Dr. Wong Ya Ping and Dr. Edmund Ng are my advisers and gave technical support and also spiritual encouragement. Dr Wong is lecture in MMU and he always gave me some concept of Computer Vision and Computer Graphic. When I started my Master degree's life, I had visited Dr Edmund's workshop in UNIMAS, exposed and experienced the application of AR. And again, I felt appreciate and thankful to all my supervisors.

Next, I would like to thank the Dean, lecturers and staffs of School of Mathematical Sciences and Institute of Postgraduate Studies, USM for providing the materials and information needed to finish the project. With the provided infrastructure, it helped to speed up the research.

Lastly, I am very thankful to all my friends as well as my course-mate. They gave me encouragement when I was facing some difficulties along the journey of research. Furthermore, I need to thank to my parents as they had provided me a stress-less environment to go on to my Master study.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLE

Page

PENGURANGAN HINGAR IMEJ DENGAN KAEDAH PEMBAHAGIAN DAN PENGESANAN HUJUNG JARI PADA TANGAN TERULUR DENGAN ALGORITMA HIMPUNAN CEMBUNG YANG HIBRID PADA TANGAN TERULUR

ABSTRAK

Pengesanan hujung jari merupakan salah satu alat input yang penting dalam Penglihatan Komputer. Hasil pengesanan ini digunakan dalam pelbagai bidang, sama ada dalam Augmentasi Realiti Saling-tindak, pengenalian tangan isyarat, dan juga industri permainan. Sarung tangan data dan Kinect adalah peranti biasa yang digunakan untuk pengesanan. Namun, ketepatannya perlu dipertingkatkan. Masalah yang perlu diselesaikan termasuk peningkatan kecekapan pengurangan hingar dan ketepatan pengesanan hujung jari. Kajian ini menumpu kepada pengesanan hujung-hujung jari dalam satu imej tangan terulur. Projek ini dibahagi kepada 3 peringkat, iaitu pengurangan hingar pada imej, himpunan cembung pada tangan terulur, dan pengesanan hujung-hujung jari. Dalam peringkat 1, kaedah pembahagian dijalankan untuk mengurangkan hingar imej. Strategi kaedah ini adalah membahagikan imej tangan terulur supaya dijadikan imej-imej yang lebih kecil dan menganalisa setiap imej kecil tersebut. Imej-imej kecil yang mengandungi bilangan pixel berwarna kulit yang rendah akan dianggap sebagai hingar imej dan dibuang. Selepas proses ini, imej yang "bersih" akan dihantar ke peringkat 2 untuk pembentukan himpunan cembung. Dalam peringkat 2, Algoritma himpunan cembung berhibrid telah dibangunkan untuk mengekstrak bucu himpunan cembung daripada imej tangan terulur tanpa melalui proses pengesanan tepi. Algoritma Jarvis March dan Bresenham digunakan untuk

mengekstrak 7 bucu daripada himpunan cembung. Himpunan cembung yang diekstrak akan diproses pada peringkat seterusnya, iaitu pengesanan hujung jari. Dalam peringkat terakhir, hujung-hujug jari dikesan daripada himpuan cembung di peringkat 2. Dalam peringkat ini, setiap sisi 7-bucu himpuan cembung akan dikira dan dianalisa untuk meramalkan lokasi hujung jari. Dua sisi dari himpuan cembung yang terpanjang akan diambil sebagai sisi yang menyambungkan pergelangan tangan dengan ibu jari dan pergelangan tangan dengan jari kelengkeng. Dalam kes ini, kita dapat menentukan 2 bucu himpuan cembung yang menyambungkan pergelangan tangan, dengan itu 5 bucu yang lain akan merupakan hujung-hujung jari. Kesimpulannya, dalam Peringkat 1, Kaedah Pembahagian bukan sahaja dapat menapiskan hingar imej, ia juga dapat mengekalkan bentuk tangan, jika berbanding dengan Algoritma Matematik Morfologi. Algoritma Himpunan Cembung Berhybrid dapat memintas proses pengesanan tepi serta mengurangkan pra-pemprosesan, secara tidak langsung mempercepatkan pengiraan hujung jari.

A PARTITIONING METHOD IN NOISE REDUCTION AND A HYBRID CONVEX HULL ALGORITHM FOR FINGERTIPS DETECTION ON AN OUTSTRETCHED HAND

ABSTRACT

Fingertips detection is one of the important input tools in Computer Vision. The results of the detections are used in many areas for example Interactive Augmented Reality, hand-gesture recognitions, and gaming industries. The data glove and the Kinect are commonly used for detection. However the accuracy of detection still needs to be improved. The problems to be solved include improvements in the efficiency of noise reduction and the accuracy of the fingertips detection. This project focuses on fingertips detections of an outstretched hand in an image. The project is divided into 3 stages, which are image noise reduction, convex hull finding of the outstretched hand and finally fingertips detection. In stage 1, a partitioning method is developed to perform image noise reduction. The main idea for this method is to divide the image into small pieces and analyze number of skin pixels in the small pieces. Image pieces which contain less than half skin pixels are considered as image noise and discarded. After this process, the "clean" image is then proceeded to stage 2 for convex hull formation. In stage 2, a Hybrid Convex Hull Algorithm is developed to extract the convex hull vertices directly from the outstretched hand image, without going through the edge detection processes. Jarvis March and Bresenham Algorithm are used to perform the extraction of 7 vertices of the convex hull. The convex hull extracted will be processed in next stage for fingertips detection. In the final stage, every edge of the 7-vertex convex hull is calculated and analyzed

for prediction of the fingertips location. The 2 longest edges of the convex hull are taken as the edges connecting wrist-thumb and wrist-pinky fingers. In that case, we can find out the 2 convex hull vertices which connect the wrist, thus the other 5 vertices are fingertips. In the conclusion for Stage 1, partitioning method not only can filter the noise, but it also preserves the shape of the hand better, as compared with the Mathematical Morphology algorithm. In Stage 2, the Hybrid Convex Hull Algorithm which is introduced in this thesis intends to bypass the edge detection to minimal pre-processing, so that the speed of forming the convex hull can be increased indirectly. In Stage 3, i.e. fingertips detection stage, we avoid histogram analyzing and this will increases the speed of calculation the fingertips.

CHAPTER 1

INTRODUCTION

1.1. Background

Computer input tools are necessary to make any computer application work. The input tools come in various styles and format, in terms of physical and non-physical. Conventionally, mouse and keyboard are the most popular and "longest survival" physical input tools over the years if compare to light-pens, joysticks, trackball and pointing stick [1]-[3]. Nevertheless, the latest overwhelming trend for physical input tool is touch-screen input for tablets and mobile phone devices. In general, physical input tools constraint the users' movements, i.e. users need to sit in front of the screen and have to be "within an arms' length" to the devices.

On the other hand, non-physical input technology such as recognition-based user interface (RBUI) that involves digitally input of gestures and speech is getting more and more attention amongst the researchers [4]-[6]. Though RBUI is not new in discussions amongst researchers, this technology gained its popularity around the globe when Nintendo released its *wii* version in 2006, to control the avatar remotely via accelerometer and optical sensing technology [7]. It is anticipated that the next interactive application which will stir a craze in the gaming industry is Microsft's Kinect for Xbox 360, which is based on PrimeSense light-coding technology for motion and depth detection. Unlike Nintendo *wii*, Kinect for Xbox requires no external controller device, i.e. the player is the controller [8]. Kinect uses hardware algorithm to perform skeleton detection and motion tracking. It is able to detect

human head, body, limbs, and hands and even down to fingers. However, the accuracy of Kinect's built-in hardware algorithm needs improvements especially when detecting the hands which are far from the device; those hands are usually detected mistakenly as background [9]-[10].

Hand detection can also be done without the pricy Kinect. With only a low-cost Personal Computer camera, we can detect the hand and even fingertips of the hand from the image captured by the Personal Computer camera. Nevertheless, it is not a one-step process but a series of processes must be carried out for the detection. These processes involve software algorithms development and thus can keep the project cost lower if compare to Kinect development.

Hand detection is also one of the important features in interactive Augmented Reality (AR). Comparatively, AR is a better environment as far as gaming industry is concerned, as AR can help to achieve true 3D immersive experiences and provide the user with six degree-of-freedom when interacting with the 3D virtual objects. AR uses pattern recognition technology to display virtual objects in the real environment. Hand detection algorithms can be embedded in AR to enable interactions between the hand and the overlay virtual objects.

In the normal practice for hand detection, the classification of series of processes starts from binary image conversion, convex hull finding and finally edge detection to filter some interior points, for more efficient time complexity during convex hull [11]-[12].

1.2. Motivation

As discussed in previous section, hand detection is one of the key input tools for user interface. The Kinect, which is getting popular amongst the game users, is used as game console for Xbox360. In this case, the joystick is not required to control the game, but the user can control the game by providing body/hand gestures input to Kinect. The input features which are recognizable by the Kinect including skeleton detection, motion tracking and hand gestures recognition, thus the user can use body language to play the game without joystick. However, the accuracy of Kinect in hand detection needs improvement especially when the hand is far from the device. In which case, the hand is considered as part of the background and not being able to be detected.

Hand detection also serves as an input tool in Augmented Reality (AR) which can be used to interact with the virtual object once detected.

In view of the above features mentioned, this project aims to develop an efficient hand detection algorithm and further into fingertips detections, which can be used for Kinect and Augmented Reality.

1.3. Overview of the Project

In a broad concept, hand detection may refer to the input detected on touch-screen monitors by physically touching the screen, or hand gesture input and fingertips detection from a digital image. To avoid ambiguity, the discussion on "hand detection" in this thesis refers to hand gesture or fingertips input which detected from a digital image as in Computer Vision.

This project group series of processes into three stages, namely noise reduction, convex hull formation and fingertips detection.

3

In noise reduction stage, as the name self-explained, unwanted noise occurred automatically during image-capture is reduced to increase the accuracy of hand detection. In addition, by eliminating noise pixels in the image, computer CPU efficiency in speed will be increased. In general, morphology algorithm is the commonly used algorithm in noise reduction. However, after several experiments, we have found that the result of the hand-shape image may be reshaped. Thus, a new methodology named "Partitioning Method" is introduced in this project. Refer to the following chapters for more detailed discussions on morphological noise reduction and the new Partitioning Method. Meanwhile, in this stage, binary image conversion is also performed. The "clean" binary image is then used in the next stage for convex hull detection.

In convex hull detection stage, a new Hybrid Convex Hull Algorithm which adopts Jarvis March and Bresenham algorithm is introduced to perform the convex hull detection on the outstretched hand of the binary image. It is done by first defining a bounding box based on four global extreme values and a global maximum point of the hand image under detection. To find the convex hull, the global maximum point is used as the first convex hull vertex. Bresenham lines are then drawn from the first vertex to the right edge of the bounding box with the purpose of looking for intersection point with the hand. By using Jarvis March algorithm, the intersection point is the second convex hull vertex found and the process is repeated until all four bounding box edges have been processed. The convex hull vertices are then passed on to the next stage for fingertips detection.

In fingertips detection stage, the strategies of calculating and analyzing the length of convex hull have been used. In this stage, we will filter the vertices of convex hull so that seven vertices are accepted. Next, each length of edge of the

convex hull will be calculated. The two longest edges will assume join to the wrist end. Hence, the other edges are connected to fingertips and the coordinate of fingertips are found.

1.4.   Objectives

- To develop a simpler methodology for fingertips detection from an image.

- To proposed and apply a new methodology namely Partitioning Method in noise reduction.

- To develop a new algorithm namely Hybrid Convex Hull Algorithm for convex hull finding.

1.5.   Problem Statement

The main purpose of this thesis is to detect the fingertips from the image of an outstretched hand. In order to perform the detection, there are three major stages to be carried out,

1) To perform noise reduction from the image;

2) To form an convex hull of the outstretched hand from the de-noised image resulted from stage 1;

3) To detect the fingertips from the convex hull formed in stage 2.

In Stage 1, some methodologies, such as Morphology, reshapes and distorts the hand image after the noise reduction. In this thesis, a Partitioning Method is developed to divide the image into small pieces. Analysis of skin pixels' quantities is then performed within each piece in order to keep the pieces which contain high quantities in skin pixels. In Stage 2, a Hybrid Convex Hull Algorithm which is

developed (based on Jarvis March and Bresenham Algorithm) to by-pass the edge detection process in a convex hull formation from a hand image. By forming the convex hull directly from the hand image without going through the edge detection process, a reduction in calculation resources is expected. In Stage 3, the cloud of points at the convex hull vertices are analyzed and filtered, so that the most appropriate point among the cloud of points is selected as the vertex and thus becomes the fingertips. The edge lengths of the convex hull are calculated to conclude that the two longest edges will be the joints to the wrist end. Hence, the other edges are connected to fingertips and the coordinate of fingertips are found.

There are assumptions and scopes in the project. In terms of hand gesture, this project focuses on the outstretched hand only. The project is also designed to detect one hand only, two or more hands detection remain the extension of the project at this stage. Further, the camera must be static to avoid calculation error. We used background subtraction as the first step of noise reduction. So, the camera is assumed to be static and fixed at one point. It is also assumed that the images used for hand detection are taken in a bright environment. So that the hand and the background are distinguishable. More detailed discussion will be made in the following chapter.

1.6. Definition

I. Hand detection.

In this thesis, hand detection refers to the process to detect position or coordinate of hand from an image. In this project, we use convex hull for hand detection.

II. Fingertips detection.

In this thesis, fingertips detection refers to the process to detect position of fingers from an image after processing the hand detection. In this project, we analyze the vertices of convex hull from hand detection to determine which vertices are fingertips.

III. Simple background

It is a term to describe the situation of an image background. In this project, simple background refers to the background of that image which is monochrome and not same or close to skin color.

IV. Complex background

It is a term to describe the situation of an image background. In this project, complex background refers to the background of that image which is not monochrome and it may be close to skin color.

1.7. Scope of Data and Assumptions

We focus on an outstretched hand in this project, and some assumptions had been made.

1. There must be only 1 hand in the image. The program is only design for finding 1 user's hand. If there are 2 or more hand in the image, the result of convex hull may marge those hand become 1 convex hull, and get the wrong result.

2. The only hand gesture considered is a hand which all fingers are outstretched, with no overlapping amongst the fingers. In analyzing the convex hull to find the fingertips, the ideal case of outstretched hand will give us 7 vertices of convex hull. Hence, other gesture of hand may not

get 7 vertices of convex hull, and fail the calculation in fingertips detection.

3   The camera must be static to avoid calculation error. We used background subtraction as the first step of noise reduction. So, the camera is assuming static and fix at 1 point.

4   Lighting is in good condition. The images used for hand detection are taken in a bright environment. So that the hand and the background are distinguishable.

1.8.   Thesis Organization

The thesis consists of six chapters, in which three of the six chapters discuss in detail three new methodologies introduced in this project, noise reduction, convex hull finding, and finder tips detection.

Chapter 1 discusses the background of hand and fingertips detection. The motivation and objectives of this project is presented.

Chapter 2 reviews related work in the literature. Related work in image noise reduction, convex hull detection and fingertips detection are discussed in separate sections. For noise reduction, the review focuses on various techniques from binary image conversion to removing the unwanted pixels which do not belong to the part of a hand. For convex hull detection, the discussion touches on normal practices in multiple pre-processing steps, which include edge detection algorithm, before the actual convex hull finding algorithm is applied. While for fingertips detection, discussion is mainly focused on vision-based detection and histogram analysis.

Chapter 3 focuses on image noise reduction methodology in detail. In this chapter, advantages of Partitioning Method and the comparison with other noise reduction methodology are also discussed.

Chapter 4 introduces Hybrid Convex Hull Algorithm developed in this project. It describes the way to bypass the edge detection which simplified the pre-processing step and shows time complexity analysis of the Hybrid Convex Hull Algorithm.

Chapter 5 discusses fingertips detection methodology which applies on the convex hull vertices. Assumptions are also stated in this chapter.

Chapter 6 concludes the work done for the entire project and discusses the future work.

CHAPTER 2

LITERATURE REVIEW

2.1. Noise reduction

Image noise is a major issue yet unavoidable in image processing. Noisy images are produced mainly during image sampling or transmission. Various types of noise such as photo electronic noise, impulse and structured noise present challenging problems in image restoration, particularly in object detection. Several noise reduction techniques have been developed to overcome different types of noise and for different aspects of image restoration.

Adaptive noise reduction algorithms such as median filtering, K-nearest neighbor averaging, gradient inverse weighted smoothing, etc. were introduced in [13]-[15]. Median filtering is particularly effective for reducing impulse noise. It is performed by running through the entire image from one region to another, and replaces each pixel in question with the median of its neighboring entries. However the median based filtering method often cannot differentiate the thin lines and impulse noises and thus mistakenly removes the thin lines [15]. Zoican introduced adaptive noise reduction scheme as a method of noise reduction for salt and pepper image noise [16]. Salt and pepper noise can be found on digital image. The formation of this image noise is during the camera capturing the image. The strategy in [16]-[17] is detect-and-fix. The algorithm will detect the image noise by using the predefined threshold and calculated values. After the pixels were detected as image noise, the

algorithm will calculate and estimate the new pixel value to replace the noise pixel. This can restore and reduce the image noise.

There are other methods for image noise reduction but normally the methods used are suitable for specific purpose. Thresholding is one method to eliminate image noise or unused pixel. By using thresholding, it separates image pixels into two parts and convert into binary image [18], where the white pixel are the object as wanted part and black pixel are image noise as unwanted part.

Lu *et al.* performed human figure detection (or human detection in his paper) by using model based approach and segmentation scheme [19]. By using Kinect, it can obtain 2D head contour model and 3D head surface model of a detected human figure. The obtained model is segmented by applying image segmentation after human figure detected. In other word, after detected the human figure based on 2D head contour model and 3D head surface, the project separates human and the background, and tracking the movement of the detected human in the video. In [19], another noise reduction methodology is used. They assume the points that cannot be detected by the depth sensor are image noise. They offset the image noise to 0 in the output array, and using nearest neighbor interpolation algorithm to fill all the image noise. Figure 2.1 shows the algorithm of noise reduction used in [19].

```
┌─────────────────────────────────────────────────────────────┐
│         ┌──────────────────────┐                            │
│         │     Depth image      │                            │
│         └──────────┬───────────┘                            │
│     ┌ ─ ─ ─ ─ ─ ┼ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐             │
│     │            ▼          ┌──────────────────┐ │          │
│     │  ┌──────────────────────┐   │ Check next pixel │ │     │
│     │  │ Depth pixel can be measure? │──┘             │ │     │
│     │  └──────────┬───────────┘       TRUE      ▲   │      │
│     │        │ FALSE                          │   │        │
│     │        ▼                                │   │        │
│     │  ┌──────────────────────┐               │   │        │
│     │  │     Offset to 0      │        Repeat  │   │        │
│     │  └──────────┬───────────┘               │   │        │
│     └ ─ ─ ─ ─ ─ ─ ┼ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘          │
│   ┌───────────────▼────────────────────────┐               │
│   │ Apply the nearest neighbor interpolation │               │
│   │    algorithm, to fix the image noise.    │               │
│   └──────────────────────────────────────────┘               │
└─────────────────────────────────────────────────────────────┘
```
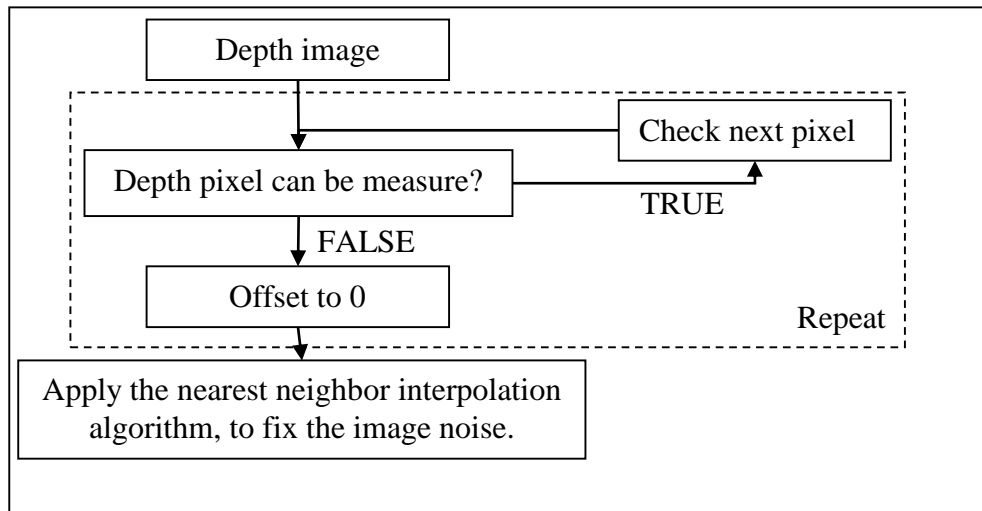
Figure 2.1   Flow of the algorithm in noise reduction.

Mathematical Morphology is another well-known algorithm in image processing for image noise reduction [20]. Morphology is based on set theory of mathematic or simple logic operation. Basically there are 4 operations in processing the binary image, which are erosion, dilation, opening and closing. The operation of the erosion will make the object become smaller and thin, but the object will become larger after operation of dilation. Opening is an operation that processes the erosion and followed by dilation. Closing is defined as dilation followed by erosion. Morphological Image-Cleaning (MIC) algorithm [21] by Peter was introduced as the Morphology method to reduce image noise. The goal for MIC is to make the lines or edges of the image sharper and clearer. However, Morphology method is not suitable in hand detection. By applying the Morphology method, the hand image tends to be reshaped [22]. After several experiments have been done, we found that the hand shape polygons are reshaped and this will be further explained in the next chapter.

In this research, partitioning method has been proposed in image noise reduction. The main idea of this methodology is to divide the image into blocks of the same size and the density of skin color will decide whether the blocks are a part

of the hand or not. If the blocks contain high density or the numbers of pixels of skin color are more a constant amount, they will be assumed as part of hand (named as hand-block). The blocks which are surrounding the hand-block will be repeated and processed with lower parameter. A comparison between partitioning method and morphology algorithm is made, particularly on image noise reduction in hand detection. The result produced by partitioning method displays a cleaner hand image with better shape preservation.

2.2.   Convex hull detection from an Image

Convex hull of a finite set of planar points S is defined as the smallest convex polygon P that encloses S [23]. It is a common structure which is widely used in many applications [24]. Various convex hull algorithms have been developed and the first algorithm was proposed in 1967 by Bass and Shubert [25]. Later in 1972, Graham was the first to introduce O(n log n) convex hull algorithm which is considered as an important algorithm in both accuracy and efficiency [26]. Graham uses radial sort on the points and checks repeatedly for convexity of every three subsequent points along the polygon perimeter. In the same year, Sklansky introduced the O(n) convex hull algorithm by 8-connect concavity tree technique [27]. The algorithm, though simple, fails on some self-intersecting polygons [28]. In 1982, Sklansky introduced a modified version [29] to add an additional process to create a polygon monotonically in both horizontal and vertical directions prior to the concavity tree technique as in his previous algorithm. However this modified algorithm does not always work and sometimes even yields non-simple polygons [30]. Despite these weaknesses, OpenCV, which is a widely used image processing tool, uses Sklansky algorithm due to its simplicity [31]. In 1973, a simple gift

wrapping algorithm (Jarvis March) with O(*nh*), where *h* is the number of convex hull edges, was introduced by Jarvis [32]. The algorithm measures the angle of the line rotating about an ankle extreme point, and takes the point which forms the smallest angle, as another extreme point. In 1985, Quickhull algorithm which has the complexity of O(*n* log *n*) was introduced. It is based on quicksort methodology to process the set of planar points by dividing the points according to two left and right extreme points and discard points strictly inside the upper and lower-hulls recursively [33]. In 1977, divide-and-conquer convex hull was introduced by Preparata and Hong with complexity O(*n* log *n*) [34]. This algorithm uses divide-and-conquer technique to divide the *x*-sorted points into two nearly equal halves repeatedly and then it finds the lower tangent for each side to merge the two sides together to form a polygon.

In binary image processing, for example fingertips detection, pre-processing in terms of key/feature points extraction, is always required before convex hull algorithm can be performed on the points extracted. In [35], the pre-processing step involves scanning the 2D image clockwise to check for the extreme points in order to form a polygon. The proposed convex hull algorithm is then performed on the extracted polygon to check for the convexity of the polygon and to make necessary adjustments. In [36], feature points of the object are extracted for generating convex hull before viewpoint invariant Fourier descriptor is used to calculate the set of invariants for three dimension planar object recognition. In [37], the pre-processing step scans the image for eight extreme points, and then divides the region within the extreme points into 5 regions, further scans are carried out on every region to find the boundary pixels for convex hull. In other projects, the typical way is to apply edge detection on an image prior to convex hull algorithm. In general, edge detection

involves filtering techniques such as Laplacian or gradient, which requires a great amount of processing work [38].

Our project intends to bypass the edge detection process and apply a Hybrid Convex Hull Algorithm directly on a binary image to extract the fingertips vertices. In this thesis, we present a hybrid algorithm to form convex hull by embedding Bresenham algorithm within Jarvis March algorithm, directly on an image with minimal preprocessing (Figure 2.2). The binary image we have used is considered to be free from noise after partitioning method is applied [39].
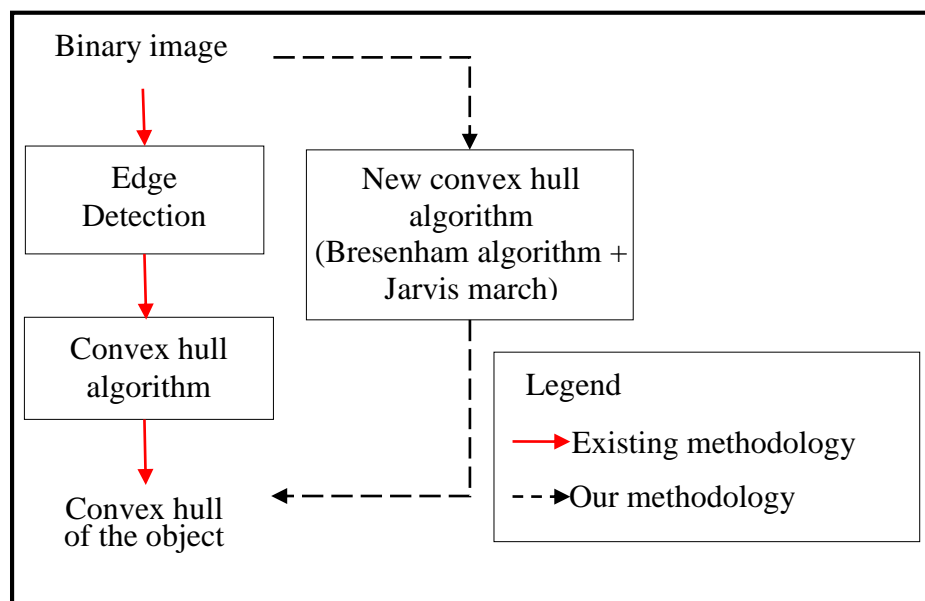
Figure 2.2 Theoretical framework of our convex hull algorithm.

2.2.1. Bresenham algorithm.

Bresenham algorithm is used to draw a straight line on a raster screen. The idea of the Bresenham algorithm is to choose the pixels with close approximation to a straight line while moving from the starting point to the ending point of a line. As the Bresenham algorithm only involves addition/subtraction arithmetic calculations in integers, it eliminates rounding errors, thus more efficient and accurate. The

15

following section describes the steps to draw a straight line in Bresenham algorithm. Note that the algorithm only involves integer calculations.

Given a straight line with 2 end points, (x_0, y_0) and (x_{end}, y_{end}), where the slope m, is $0 < m < 1$,

1. User input 2 end points of the line. The left most of the point stored in (x_0, y_0), and the right most point will be (x_{end}, y_{end})

2. The constants will be calculated:

   $d_x = x_{end} - x_0$

   $d_y = y_{end} - y_0$

   $p_0 = 2(d_y) - d_x$

3. Starting at $k=0$ and compare the p_k:

   If $p_k < 0$, plot the point (x_k+1, y_k) and calculate $p_{k+1} = p_k + 2 (d_y)$,

   else plot the point $(x_k+1, y_k +1)$ and calculated $p_{k+1} = p_k + 2 (d_y) - 2 (d_x)$.

4. Steps 3 will repeat from (x_0, y_0) to (x_{end}, y_{end}) along the line.

2.2.2. Jarvis March algorithm.

Jarvis March algorithm also known as gift wrapping algorithm in two-dimensional case. The Jarvis March is used to calculate the convex hull from finite set of points in the planer. The idea of Jarvis March algorithm is start from an outermost point and start wrapping to form convex hull. The algorithm will choose the leftmost point and wrapping in counterclockwise direction. The following sections describe the steps of Jarvis March algorithm.

1. Find the leftmost point and this point also known as 1 of the vertices of the convex hull. Assume the leftmost point is (x_0, y_0) and k as the number of points.

2. By comparing polar angles respect to point (x_0, y_0), the point clockwise furthest to the right relative to the (x_0, y_0) will be set as next vertices of convex hull.

3. Step 2 will be repeated with new vertices and process will be terminated when meet the point (x_0, y_0).

2.3. Fingertips detection

Fingertips detection is one of the key input methods in Human-Computer Interaction (HCI) nowadays. The conventional physical input devices such as mouse and keyboard are more suitable for 2D- environment and they have limited functions. For example, mouse can only perform "click", "drag" and "drop"; while keyboard can be used for text input or some simple cursor. In view of the physical input devices' limitations, vision-based hand/fingertips detection gains more attention due to its nature in 6 degree-of-freedom and also hand gesture recognition is able to perform texting, moving cursor, etc., including many functions that the physical input devices can perform. Thus, hand/finger detection is getting more important as far as HCI is concerned.

For virtual objects interaction in Augmented Reality (AR) system, free hand for interaction enables the user to obtain six-degree of freedom and the interaction to "touch" the virtual objects is more realistic visually.

Vision-Based hand detection technique was introduced in [40]. In the paper, stereo camera is used to construct a 3D environment. The finger detection processes

are divided into two parts in [40], which are hand region segmentation and hand feature extraction. After the image is captured, the color space of the image is converted from RGB to YCbCr. Skin filter and binary image conversion is applied where the skin color is changed to white color, else become black. The binary image will be processed by morphological operations to reduce the image noise. In addition, the large region (white pixel) will be considered as possible hand-region. After the hand region segmentation is done, the next process is to find the fingers from the image. By using distance transforms, the palm center is estimated. The reason is because the palm center is assumed as a point that is furthest from the contour of the hand region. A curvature based algorithm is applied to find the fingers. There are 2 assumptions which have been made, where the candidates' finger points are not close to the boundary image and the candidates finger points are not too close to each other. If the candidates' finger points are too close to the next candidate, a filtration will be applied by only 1 finger point will be chosen in the same finger. The candidates' points will be ignored if those points are close to boundary image. Next, the progress is to index the finger points. To standardize the sequent of the finger point, the thumb which contains a longer distance to other will index as 1 and follow by other fingers. Lastly, the valley points will be obtained. The valley points are said that a point which is between 2 subsequent fingertips and furthest away from those two fingertips. Figure 2.3 shows the algorithm of noise reduction which is applied in [40].

```
                    ┌──────────────────┐
                    │   Image Input.   │
                    └──────────────────┘
                             │
        ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┼ ─ ─ ─ ─ ─ ─ ─ ─ ┐
                             ▼
        │   ┌──────────────────────────────┐   │
            │     RGB convert to YCbCr.     │
        │   └──────────────────────────────┘   │
                             │
        │                    ▼                  │
            ┌──────────────────────────────┐
        │   │  Skin filter and convert to  │   │
            │  Binary image.               │
        │   └──────────────────────────────┘   │
                             │
        │                    ▼                  │
            ┌──────────────────────────────┐
        │   │  Noise reduction:            │   │
            │  Eliminate small regions.    │
        │   └──────────────────────────────┘   │
        │                               Hand Region Segmentation │
        └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ─┘
        ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─│─ ─ ─ ─ ─ ─ ─ ─ ─┐
                             ▼
        │   ┌──────────────────────────────┐   │
            │  Find the palm centre        │
        │   │  (Distance transforms).      │   │
            └──────────────────────────────┘
        │                    │                  │
                             ▼
        │   ┌──────────────────────────────┐   │
            │  Find the fingers (A curvature│
        │   │  based algorithm)            │   │
            └──────────────────────────────┘
        │                    │                  │
                             ▼
        │   ┌──────────────────────────────┐   │
            │  Index and sort the index of │
        │   │  the fingers.                │   │
            └──────────────────────────────┘
        │                    │                  │
                             ▼
        │   ┌──────────────────────────────┐   │
            │  Find valley points between  │
        │   │  fingers.                    │   │
            └──────────────────────────────┘
        │                              Hand Features Extraction │
        └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─┘
```
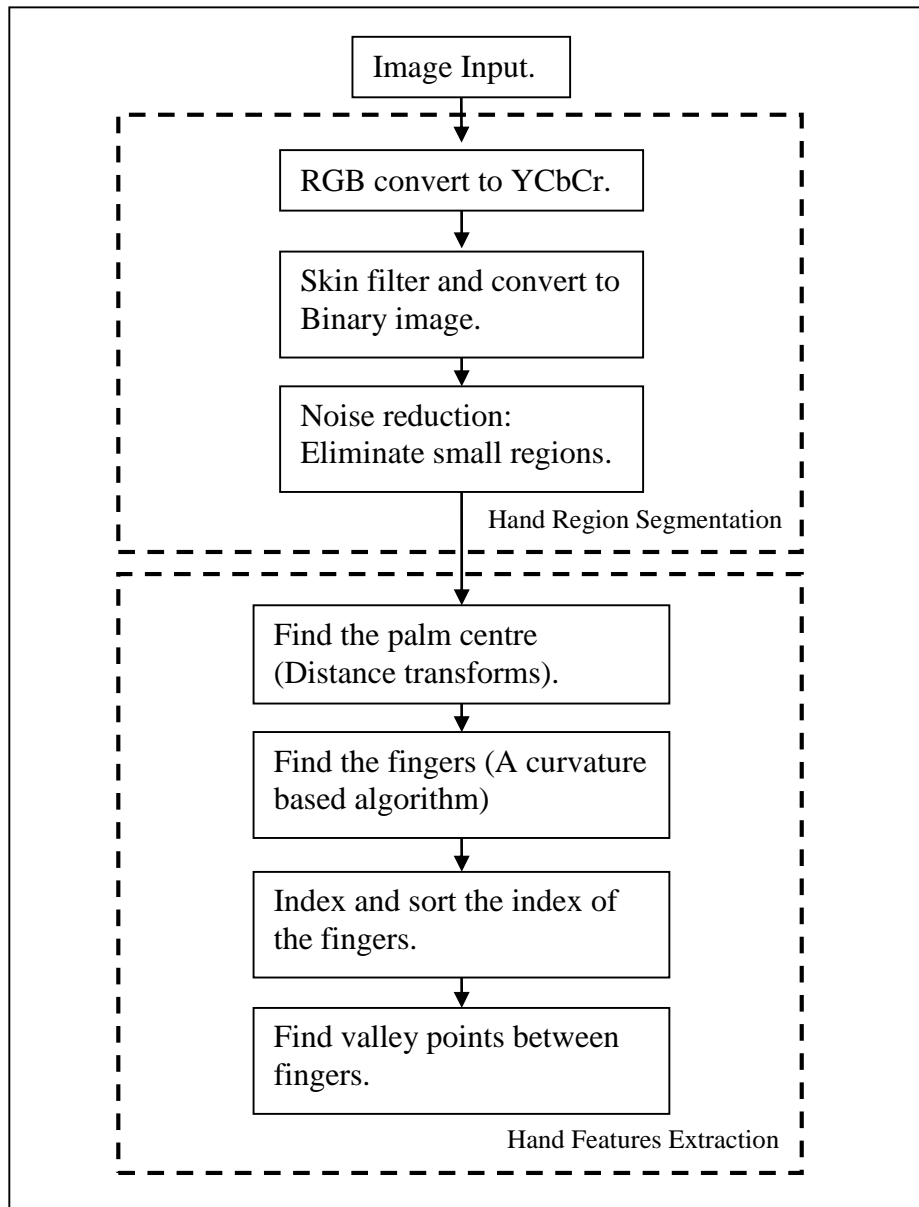
Figure 2.3   Theoretical framework in [40].

Kinect is a depth camera which based on light-coding technology for motion
tracking and detection. It is accepted by many researchers nowadays. In [41], an
analysis depth map method is used to find finger from [41]. The general idea is
assuming that, the user's fingers are close to the camera. Hence, the lowest depth
value is possible to be user's fingers. The hardware has been used are Kinect, so they
can get the depth image easily. After obtaining the depth image and user's hand
detected, image segmentation is carried out. The segmentation uses depth-

19

thresholding, which evaluates the distance between the hand and the camera. After that, a big circular filter on the image is applied and all the fingers are removed. Next, the result of palm of the hand will be subtracted from original hand image to get fingers image. It is define as user's hand if the pixels of the depth value are lowest and fulfill the assumptions made as mentioned above. Hence, the user's fingertips are obtained. Next step is palm center detection. The distance transform is applied to find the palm center. This method is better than other, where this method can be used in multidirectional because it detects finger via depth image. Unlike other method, some need to consider if the hand is always pointing upwards. However, if the user's hand is positioned upright, it may cause an error in finger detection. This project assumes that the user's hand is always facing the camera. Besides, if the hand is not the closest object to camera, or there is another object which has smaller depth value than the user's hand, it may obtain detection error. Figure 2.4 shows the algorithm of Jagdish's project.