Indian Journal of Science and Technology, *Vol 9(28), DOI: 10.17485/ijst/2016/v9i28/97821, July 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Hybrid Convex Hull Algorithm for Fingertips Detection

Guat Yew Tan[1*], Bo Shen Woun[1] and Ya Ping Wong[2]

[1]School of Mathematical Sciences, Universiti Sains Malaysia, Penang, Malaysia; tan_gy@usm.my, wounboshen@hotmail.com
[2]Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia; ypwong@mmu.edu.my

Abstract

Objectives: This article presents a hybrid convex hull algorithm to reduce computational resources in fingertips detection from an image. **Methodology:** In this paper, we suggest to reduce the computational resources by leveraging on two proven algorithms and techniques in order to extract the convex hull vertices directly from a binary image without going through the edge detection process. This is done by embedding Bresenham algorithm within Jarvis March to replace most of the work required in the edge detection process. **Findings:** The hybrid convex hull algorithm which we have suggested requires only four global extreme points to begin with and thus the pre-processing step takes much less resources. The new algorithm yields time complexity of $O(N^2)$. **Novelty/Improvement:** The hybrid convex hull algorithm offers a direct way to detect the convex hull of the original image without edge detection process.

Keywords: Bresenham Algorithm, Convex Hull, Fingertips detection, Jarvis March

1. Introduction

Convex hull of a finite set of planar points S is defined as the smallest convex polygon P that encloses S[1]. It is a common structure which is widely used in many applications[2]. Various convex hull algorithms have been developed and the first algorithm was proposed in 1967 by Bass and Shubert[3]. 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[1,4]. 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[5]. The algorithm, though simple, fails on some self-intersecting polygons[6]. In 1983, Sklansky introduced a modified version[7] 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[8]. Despite these weaknesses, Open CV, which is a widely used image processing tool, uses Sklansky algorithm due to its simplicity[9]. 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. 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[10]. In 1985, Quickhull algorithm which has the complexity of O(n log n) was introduced. It is based on quick sort 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[11]. In 1977, divide-and-conquer convex hull was introduced by Preparata and Shamos with complexity O(n log n)[12]. 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.

**Author for correspondence*

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[13], the pre-processing scans 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[14], 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[15], the pre-processing 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[16].

Our project intends to bypass the edge detection process and apply the convex hull algorithm directly on a binary image to extract the fingertips vertices. In this paper, we present a hybrid algorithm to form convex hull by embedding Bresenham algorithm within Jarvis March algorithm, directly on an image with minimal pre-processing (**Figure 1**). The binary image we have used is considered to be free from noise after partitioning method is applied[17].

The rest of the paper is organized as follows. Section 2 describes the methodologies in pre-processing and convex hull formation. The time complexity analysis is discussed in Section 3 and conclusion is drawn in Section 4.

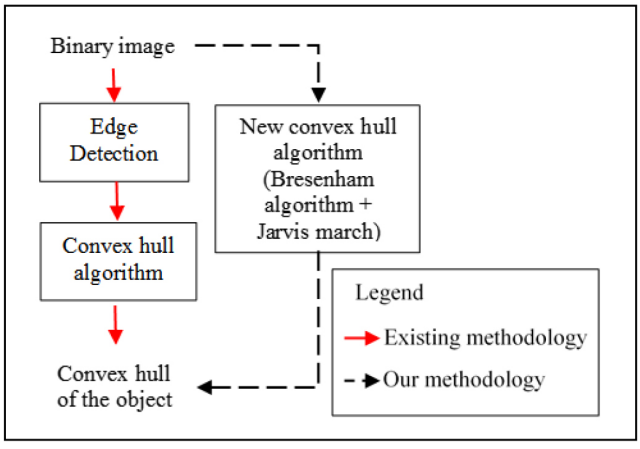

Figure 1. Theoretical framework of our convex hull algorithm.

2. Methodology

Our algorithm takes in a binary image produced in[17], in which de-noising procedures have been applied. Our pre-processing steps involve merely 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 following sections describe the two major steps involved: pre-processing and convex hull detection.

2.1 Pre-processing Step: Defining the Bounding Box

In this step, the white pixels of the image are scanned once to find the four global extreme values m_x, M_x, m_y, M_y and a global maximum point, ($hand_x$, M_y), . Let

$$H(x,y) = \begin{cases} 255, & hand\ pixel \\ 0, & background \end{cases}; m_x \le x \le M_x, m_y \le y \le M_y$$

where

m_x and M_x are global minimum and maximum horizontal values respectively,

m_y and M_y are global minimum and maximum vertical values respectively, and

$H(x, y)$ is the brightness of the pixel at coordinate (x, y).

Thus H is the bounding box of the image defined by the rectangular region confined within m_x, M_x, m_y and M_y; and $H(hand_x, M_y) = 255$ (**Figure 2**).

2.2 Proposition

Proposition 1. Let H be the bounding box containing a set of points, S. A point p, where $p \in S$, residing on the edge of bounding box is the vertex of the convex hull of S.

Proof. A convex hull of a finite set of points S is defined by the smallest convex polygon that encloses S[1]. The bounding box H is confined by the extreme values

of S, given by m_x, M_x, m_y, M_y. Thus, for a point p where $p \in S, p_x \in \{m_x, M_x\}$ or $p_y \in \{m_y, M_y\}$, p is the vertex of convex hull of S.

Proposition 2. Pick a convex hull vertex, name it p_0, and draw a straight line L from p_0 until L intersects with one edge of the bounding box. Ensure L is an exterior line of the convex polygon, i.e. L does not intersect with the convex polygon. By using p_0 as a pivot point, rotate L in clockwise direction. The first intersection point encountered by L with the hand pixel is a convex hull vertex (Figure 3).

Proof. Pin one end of a rubber band on a vertex, p_0, pull the other end of the rubber band horizontally towards an edge of the bounding box, and ensure the rubber band stays as an exterior line to the convex polygon. This can be done from Proposition 1 by selecting the rightmost top vertex of the convex hull as p_0, where $m_x \le p_{0x} \le M_x$, $p_{0y} = M_y$. The rubber band will form a horizontal line, L, intersecting with the right edge of the bounding box, H. As p_0 is the rightmost top vertex of the convex hull, let l be a point on L, $\forall l \in L$, $l \neq p_0$, $p_{0x} < l_x \le M_x$ and $l_y = M_y$, $l \notin S$. Thus, L is the exterior line of the convex polygon. While p_0 is pinned, L rotates around p_0 clockwise until it intersects with a point p_1 where $p_1 \in S$. In other words, L rotates θ angle with respect to its original position where θ is the smallest angle before it meets the first point p_1 where $p_1 \in S$. It is clear that p_1 is a vertex of the convex hull as mentioned in Jarvis March algorithm[10].

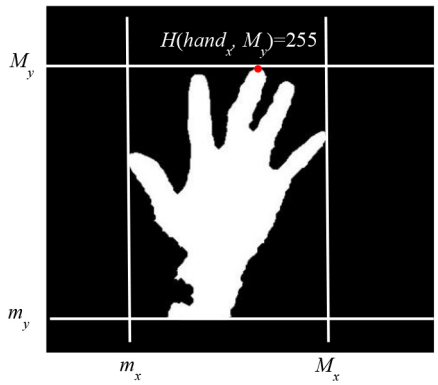

Figure 2. Extreme values, m_x, M_x, m_y, M_y; global maximum point, ($hand_x$, M_y); and bounding box, H.

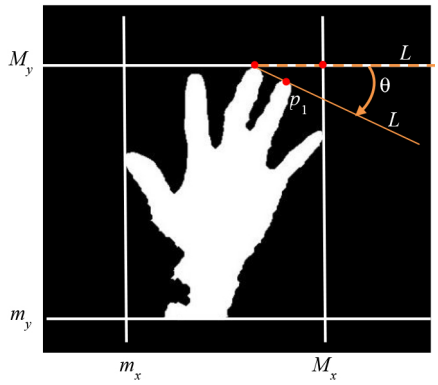

Figure 3. p_0 is a convex hull vertex and used as the pivot point, L is a straight line drawn from p_0 towards the right edge of the bounding box, H.p_1 is the first intersection point encountered while L rotatesq degree clockwise, thus $p_1$is a convex hull vertex.

2.3 Convex Hull Formation

From Proposition 1, as $p_0 = (hand_x, M_y)$ is the global maximum point, it is also the vertex of the convex hull. To search for subsequent vertex, see the following steps.

1. Take p_0 as the beginning pivot point, and let $r = (M_x, M_y - 1)$ as the intersection point of L and the right bounding box.
2. The brightness of all points from p_0 to r is checked. Bresenham algorithm is used to step through the points from p_0 to r.
3. Given a point p, where $p \in L$, such that $H(p) = 0, \forall p \in L$; move r one pixel down vertically, that is, let $r = (r_x, r_y - 1)$, and repeat step 2 until $H(p) = 255$ is encountered.
4. If $H(p) = 255$, p is the vertex of the convex hull (Proposition 2), brightness check stops.
5. p becomes the new p_0 and let $r = (r_x, r_y - 1)$, L connects the new p_0 to r.
6. Step 2 to Step 5 are repeated until $r_y = m_y$.
7. When $r_y = m_y$, r moves along the m_y line to the left, i.e. $(r_x - 1, m_y)$ becomes the new r and Step 2 to Step 5 are repeated by replacing $r = (r_x - 1, m_y)$.
8. The processes continue until all four edges of H have been examined.

Figure 4 shows the algorithm framework for the steps mentioned above for convex hull formation. In point detection of an image, there will be many convex hull vertices found. And as we are expecting hand image, the regions with high concentration of vertices are fingertips regions. Thus, any point within the group of the vertices can be used. For simplicity, we use the last point of a group as the fingertip point.

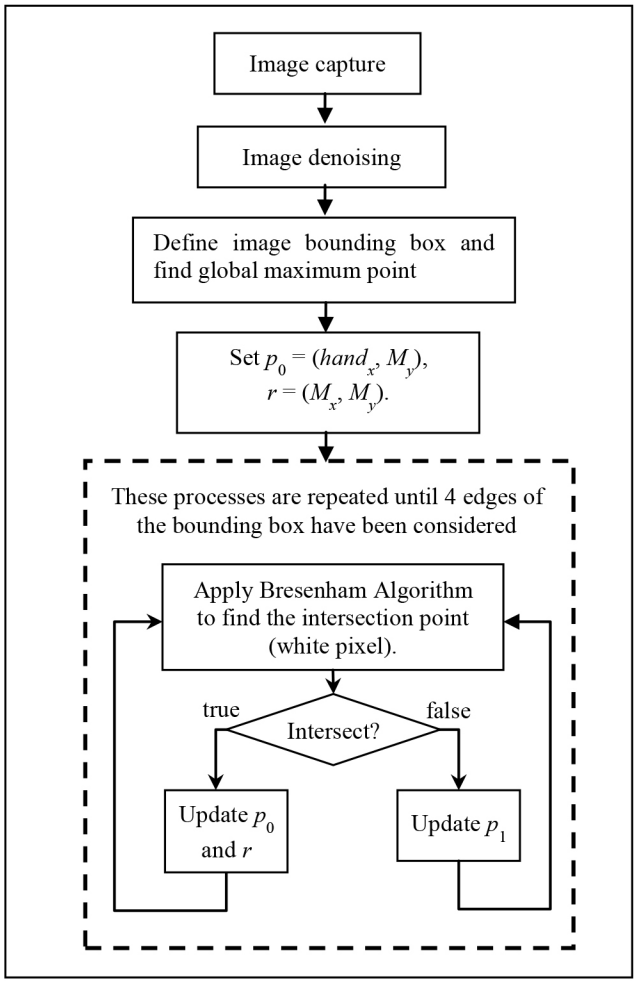

Figure 4. The algorithm framework.

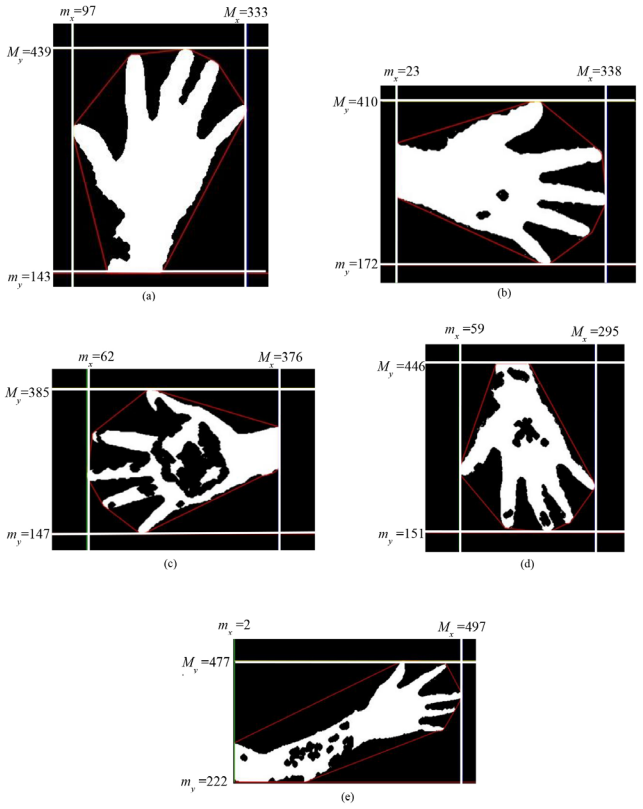

Figure 5. Various pose of hand images.

3. Time Complexity Analysis

In the pre-processing step, we are looking for the global extreme values of an *N*x*N* image. It scans through the whole image once to find the four extreme values, thus having complexity of O(N^2). The scanning activity involves only checking the brightness value of each pixel.

For convex hull formation, only some of the black pixels are checked, i.e. the black pixels in between the fingers will not be checked. Thus, it is only the black pixels outside of the convex hull but within the bounding box are being checked. We have captured a few hand images as examples in Figure 5 and the black pixel percentages are shown in Table 1.

From Table 1, the maximum percentage of total black pixels outside of convex hull in the bounding box in all five images is 46.58%. We make a reasonable conclusion that for any outstretched hand image, the total black pixels outside convex hull in bounding box is always less than 50%. Thus, in the worst case scenario, the number of black pixels to be processed are ½N^2. The time complexity is O(N^2). However, we must take into consideration that the checking algorithm in Bresenham involves only integers and the '+' operator, thus the processing time will be much faster.

Table 1. Percentage of black pixels outside of the convex hull in the bounding box

Figure	Number of black pixels outside of the convex hull (B)	Total bounding box pixels (A)	% of black pixels outside of the convex hull in the bounding box (% B in A)
5(a)	23,834	69,325	34.38
5(b)	26,413	74,418	35.49
5(c)	25,072	74,181	33.80
5(d)	25,337	69,090	36.67
5(e)	58,901	125,476	46.94

4. Conclusion

A hybrid convex hull algorithm by using Bresenham algorithm embedded in Jarvis March has been developed. The new algorithm is expected to reduce the resources allocated for edge detection and apply the convex hull algorithm directly on the pixels in binary image with minimal processes. Though the time complexity of our algorithm is $O(N^2)$, our algorithm uses only integers and the '+' operators and thus is expected to be efficient as far as computer processing is concerned.

5. Acknowledgements

This work is supported by RU-PRGS Universiti Sains Malaysia (1001/PMATHS/836026) and KPT-FRGS Malaysia (203/PMATHS/6711428).

6. References

1. o'Rourke J. Computational Geometry in C.2nd Edition. Cambridge University Press; 1998 Oct 13.
2. Rosenfeld A, Kak AC. Digital Picture Processing. Volume 1. 2nd Edition. Academic Press: New York, 1985.
3. Bass LJ, Schubert SR. On Finding the Disc of Minimum Radius Containing a Given Set of Points. Mathematics of Computation. 1967 Oct 1; 21(100): 712–4.
4. Graham RL. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. Information Processing Letters. 1972 Apr; 10(3):132–3.
5. Sklansky J. Measuring Concavity on a Rectangular Mosaic. IEEE transactions On Computers.1972 Dec 1; C-21(12):1355–64.
6. Bykat A. Convex Hull of a Finite Set of Points in Two Dimensions. Information Processing Letters. 1978 Oct 31; 7(6):296–8.
7. Sklansky J. Finding the Convex Hull of a Simple Polygon. Pattern Recognition Letters. 1982 Dec 31; 1(2):79–83.
8. Toussaint GT, ElGindy H. A Counterexample to an Algorithm for Computing Monotone Hulls of Simple Polygons. Pattern Recognition Letters. 1983 May 31; 1(4):219–22.
9. Team OD. Open CV Reference Manual. Version 2.2. 2010; 1073. Available from: http://docs.opencv.org/3.0-beta/opencv2refman.pdf. Date accessed: 17/06/2016.
10. Jarvis RA. On the Identification of the Convex Hull of a Finite Set of Points in the Plane. Information Processing Letters. 1973 Mar 31; 2(1):18–21.
11. Preparata FP, Hong SJ. Convex Hulls of Finite Sets of Points in Two and Three Dimensions. Communications of the ACM. 1977 Feb 1; 20(2):87–93.
12. Preparata FP, Shamos MI. Computational Geometry: An Introduction. Springer, New York, 1985.
13. Ye QZ. A Fast Algorithm for Convex Hull Extraction in 2D Images. Pattern Recognition Letters. 1995 May 31; 16(5): 531–7.
14. Yu MP, Lo KC. Object Recognition by Combining Viewpoint Invariant Fourier Descriptor and Convex Hull. IEEE Proceedings of the 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong. 2001; 401–4.
15. Zhang X, Tang Z, Yu J, Guo M. A Fast Convex Hull Algorithm for Binary Image. Informatica. 2010; 34(3):369–76.
16. Shrivakshan GT, Chandrasekar C. A Comparison of Various Edge Detection Techniques used in Image Processing. IJCSI International Journal of Computer Science Issues. 2012 Sep; 9(5):272–6.
17. Woun BS, Tan GY, Wong YP, Ng E. Partitioning Method in Noise Reduction to Improve Hand Detection, IJFCC *International Journal of Future Computer and Communication. 2013 Oct 1; 2(5):395–8.*