Neurocomputing 77 (2012) 212–221

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A fast convex hull algorithm with maximum inscribed circle affine transformation

Runzong Liu, Bin Fang*, Yuan Yan Tang, Jing Wen, Jiye Qian

College of Computer Science, Chongqing University, Chongqing 400030, PR China

ARTICLE INFO

Article history:
Received 5 May 2011
Received in revised form
31 August 2011
Accepted 5 September 2011
Communicated by Y. Yuan
Available online 22 September 2011

Keywords:
Convex hull
Computational geometry
Affine transformation
Point pattern
Shape analysis

ABSTRACT

This paper presents a fast convex hull algorithm for a large point set. The algorithm imitates the procedure of human visual attention derived in a psychological experiment. The merit of human visual attention is to neglect most inner points directly. The proposed algorithm achieves a significant saving in time and space in comparison with the two best convex hull algorithms mentioned in a latest review proposed by Chadnov and Skvortsov in 2004. Furthermore, we propose to use an affine transformation to solve the narrow shape problem for computing the convex hull faster.

1. Introduction

Convex hull is a fundamental construction for computational geometry [1]. It makes possible solving many other problems, for example, half-space intersection, Delaunay triangulation, linearly separability analysis [2], image registration [3], natural computation [4], and pattern recognition [5,6]. Convex hull algorithms are widely used in geographic information systems, and they are also used in astrophysics data system [7].

The notion of convex hull is defined in the following ways [1]:

- Definition 1. Region D, which belongs to space E^2, is called convex if for any two points d_1 and d_2, which belong to D, segment $d_1 d_2$ entirely belongs to D.
- Definition 2. A convex hull of a point set S, which belongs to space E^2, is a boundary of the smallest convex region in E^2, which surrounds S.

There are two variants of the convex hull problem [8]:

- Problem 1. A set S is given in E^2, which contains N points. The task is to pick out those points, which are the vertexes of the convex hull.
- Problem 2. A set S is given in E^2, which contains N points. The task is to build a convex hull of these points (i.e. to find the boundary $CH(S)$).

Problem 2 can be easily solved by solving Problem 1. Here we only consider Problem 1.

There are many different convex hull algorithms, for example, Granham scan algorithm [9], Andrew algorithm [10], Quickhull algorithm [11], and Datta and Parui's algorithm [12]. All of the above algorithms are widespread, while Quickhull algorithm and Andrew algorithm are the two best algorithms in a latest review of convex hull algorithms [8]. Recently, there are also some studies on convex hull algorithm [13–16]. Each of the studies made a valuable contribution to a particular field of convex hull algorithm. A latest literature [17] proposed an algorithm composed of two existing algorithms, i.e. Quickhull and Granham scan algorithm. We call it as QuiGran algorithm below. The study reported that QuiGran performed better than Quickhull as an improved version of the Graham scan algorithm. The common architecture (see Fig. 1) of convex hull algorithms can be deduced by a more common one (see Fig. 2), with each component defined as follows:

- Initial convex hull: The initial convex hull can be a starting point (Granham scan algorithm), or points for partitioning (Andrew algorithm, Quickhull algorithm). The initialization is easy but important for the algorithm efficiency.
- Point location: Point Location can be sorting the points (Graham scan algorithm), or locating the points to specific partitions (Andrew algorithm, Quickhull algorithm).
- Point comparison: Point comparison is to compare the query point with some threshold to decide whether the point is a vertex of the convex hull. The thresholds might be different for different algorithms, for instance, whether the angle is greater

* Corresponding author. Tel.: +86 23 65112784; fax: +86 23 65102502.
E-mail addresses: fb@cqu.edu.cn, newmaybetrue@163.com (B. Fang).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.09.011

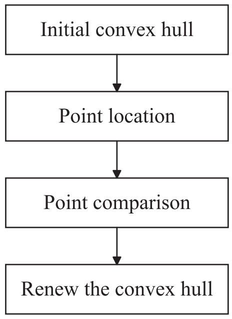

Fig. 1. The common architecture of a convex hull algorithm.

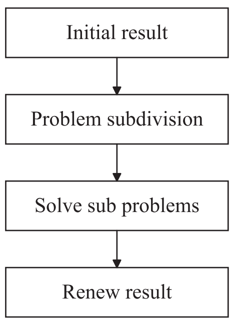

Fig. 2. The common architecture of solving a problem.

than or equal to π (Graham scan algorithm, Andrew algorithm) or whether the point is inside a triangle (Quickhull algorithm).

- Renew the convex hull: The renewal of a convex hull can be made by adding a new vertex to the original convex hull vertex set (Graham scan algorithm), or by merging several convex hull vertex subsets into a bigger convex hull vertex subset or the whole convex hull vertex set (Andrew algorithm, Quickhull algorithm).

The initialization of the convex hull can be achieved by some simple calculations for the extreme points of the point set, e.g. point with max x coordinate. Besides, only several points need to be processed to renew the convex hull. Therefore, the main spent of a convex hull algorithm is the spent of point location and point comparison which are two basic query processes for every convex hull algorithm. One merit of human recognition procedure is visual attention [18]. When human tries to find the convex hull of a point set, he or she only pays attention to the points near the boundary. Most inner points are neglected by him or her using the information of an initial estimation of the boundary of the point set. The initial estimation of the boundary is achieved by the observation of some extreme points of the point set, which also belong to the vertex set of the convex hull of the point set. On this account, we should use the information of the extreme points of a point set to reduce the computation of point location and point comparison as much as possible. However, most of the current convex hull algorithms have not fully or efficiently utilized the information of the extreme points of a point set. In this paper, the proposed algorithm, i.e. the visual-attention-imitation convex hull algorithm (VAICH), makes a big advance on the usage of information of the extreme points of a point set.

The rest of the paper is organized as follows. Before proposing visual-attention-imitation convex hull algorithm (VAICH), some mathematical lemmas about convex hull are discussed in Section 2. In Section 3, our approach VAICH is proposed. In Section 4, we establish Maximum Inscribed Circle Affine Transformation (MICAT) to solve the narrow shape problem for computing the convex hull faster. The experimental results about VAICH and MICAT are presented in Section 5. Section 6 finally provides the conclusions.

2. Mathematical basis for visual-attention-imitation convex hull algorithm

Lemma 1. *Given a point set S, the centroid O of S is inside the convex hull of S.*

Proof. $S=[S_1,S_2,\ldots,S_n]$ is a point set. $[A_1,A_2,\ldots,A_h]$ is the vertex set of the convex hull of S. Then, the necessary and sufficient condition for a point A inside the convex hull is

$$A=r_1A_1+r_2A_2+\cdots+r_hA_h,$$
$$\sum_{i=1}^{h} r_i=1, \quad r_i\ge 0. \tag{1}$$

From the definition of convex hull, we know that any point from point set S is inside the convex hull, that is

$$S_i=r_{i_1}A_1+r_{i_2}A_2+\cdots+r_{i_h}A_h,$$
$$\sum_{j=1}^{h} r_{i_j}=1, \quad r_{i_j}\ge 0,\ i=1,2,\ldots,n. \tag{2}$$

If O is the centroid of the point set S

$$O=(S_1+S_2+\cdots+S_n)/n. \tag{3}$$

$$(2)\&(3)\Rightarrow$$
$$O=r_{o_1}A_1+r_{o_2}A_2+\cdots+r_{o_h}A_h,$$
$$r_{o_i}=1/n\sum_{j=1}^{n} r_{j_i}, \quad i=1,2,\ldots h,$$
$$r_{j_i}\ge 0\Rightarrow r_{o_i}\ge 0,$$
$$\sum_{i=1}^{h} r_{o_i}=\sum_{i=1}^{h}\left(1/n\sum_{j=1}^{n} r_{j_i}\right)=1/n\sum_{j=1}^{n}\sum_{i=1}^{h} r_{j_i}=1. \qquad \square \tag{4}$$

Since (4) satisfies (1), O is inside the convex hull.

Lemma 2. *Let E be a point of a point set S, and O be the centroid of S, A and B be two vertexes of the convex hull of S, and the direction from O to E intersect with the segment AB at point D. If segment OE is shorter than the distance from O to AB then E is strictly inside the convex hull of S.*

Proof. From the definition of convex hull, the segment AB entirely belongs to the convex region. Since D is a point of the segment AB, D is inside the convex hull. From Lemma 1, we also know that the centroid O is inside the convex hull. Therefore, the segment OD entirely belongs to the convex region (see Fig. 3). If segment OE is shorter than the distance from O to AB, OE is shorter than the segment OD. Because D is a point on the direction from O to E and OE is shorter than the segment OD, E is inside the

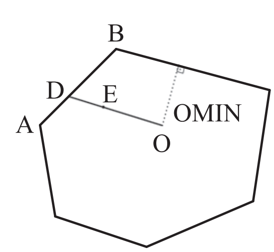

Fig. 3. The locations of points in Lemmas 2 and 3.