Neurocomputing 133 (2014) 231–236

Contents lists available at ScienceDirect

Neurocomputing

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

Letters

An enhanced version and an incremental learning version of visual-attention-imitation convex hull algorithm

Runzong Liu [a,b], Yuan Yan Tang [a,b,*], Bin Fang [b], Jingrui Pi [b]

[a] Faculty of Science and Technology, University of Macau, Av. Padre Toms Pereira, Taipa, Macau, China
[b] College of Computer Science, Chongqing University, Chongqing 400030, PR China

ARTICLE INFO

Article history:
Received 26 December 2012
Received in revised form
5 September 2013
Accepted 11 November 2013
Communicated by Tao Mei
Available online 10 January 2014

Keywords:
Convex hull
Computational geometry
Algorithm
Incremental learning

ABSTRACT

This paper presents an enhanced version and an incremental learning version of the visual-attention-imitation convex hull algorithm reported in our latest paper in Liu et al. (2012) [3]. The enhanced algorithm merges the virtue of point comparison of the Graham scan algorithm into the visual-attention-imitation convex hull algorithm. In comparison with its previous edition, the proposed algorithm achieved a significant time saving. In view of machine learning, there are interesting situations where training data acquisition must take place over time. An incremental learning version is also proposed in this paper in order to compute convex hulls of point sets whose points are acquired over time. The incremental learning version reuses the prior results and computes the new convex hull without processing of previous points. Experimental results show that the incremental learning version is more flexible and more efficient for incremental learning tasks.

1. Introduction

Convex hull is a fundamental construction for computational geometry [1]. Graham scan algorithm [2] is often called [1] the first algorithm of computational geometry. The visual-attention-imitation convex hull algorithm (VAICH) [3] is one of the fastest convex hull algorithms for large point sets. VAICH achieves a significant saving in time and space, in comparison with two of the best convex hull algorithms [4], i.e., the Andrew algorithm [5] and the Quickhull algorithm [6]. It is also superior to a latest reported algorithm [7] which combines the Quickhull algorithm and the Graham scan algorithm. In this paper, VAICH and the Graham scan algorithm are merged into a more powerful convex hull algorithm for large point sets.

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_1d_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 [4]:

- 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.

The common architecture of convex hull algorithms in our latest paper [3] is shown in Fig. 1. For the original edition of VAICH [3], the point comparison of Quickhull algorithm is used as a significant part of the point comparison. The point comparison of Quickhull algorithm needs to analyze three edges of a triangle. However, the point comparison of Graham scan algorithm is much simpler. It just needs the judgment on the sign of the cross product of two vectors. In this paper, we merge the virtue of point comparison of the Graham scan algorithm into VAICH.

Furthermore, we can treat the computation of convex hull as a learning task. In view of machine learning, the original edition of VAICH assumes that the input point set is available a priori and that learning ceases once this set has been duly processed. However, there are a number of interesting situations where learning must take place over time, in a kind of continuous fashion rather than as a one shot experience. It is argued that incremental tasks are rather ubiquitous in learning and that the most natural

* Corresponding author at: Faculty of Science and Technology, University of Macau, Av. Padre Toms Pereira, Taipa, Macau, China. Tel.: +853 83978473; fax: +853 28838314.
E-mail address: yytang@umac.mo (Y. Yan Tang).

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.11.013

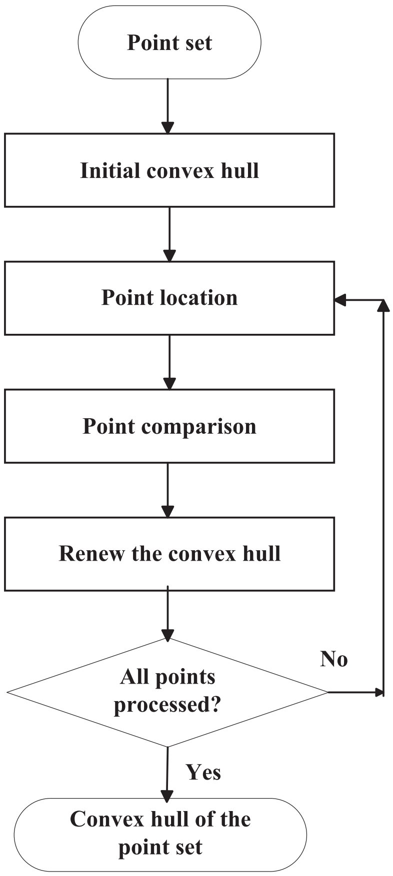

Fig. 1. The common architecture of a convex hull algorithm.

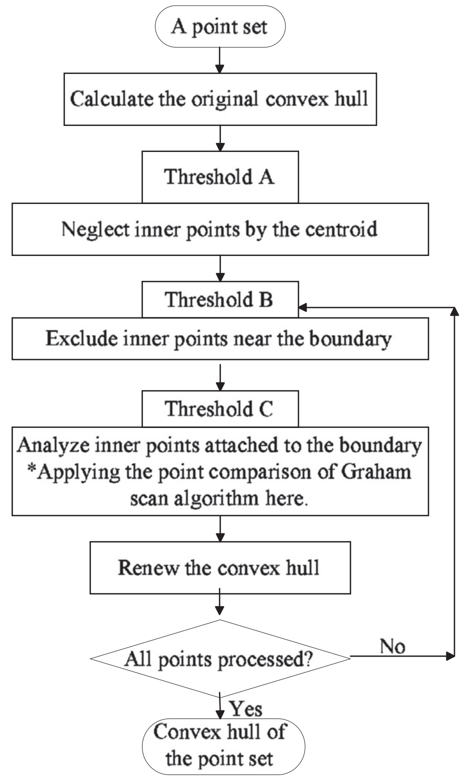

Fig. 2. Flow chart of the combined algorithm of visual-attention-imitation convex hull algorithm and Graham scan algorithm.

and flexible way to tackle incremental learning tasks is with incremental learning algorithms [8–10]. An incremental learning version of VAICH is proposed in this paper in order to tackle incremental learning tasks.

2. The combined algorithm of the visual-attention-imitation convex hull algorithm and the Graham scan algorithm

The detail procedure of visual-attention-imitation convex hull algorithm [3] is as follows (see Fig. 2):

Step 1: *Calculate the initial convex hull*. The following eight extreme points of the point set can be found out by simple calculations: point with maximum x coordinate, point with minimum x coordinate, point with maximum y coordinate, point with minimum y coordinate, point that maximizes the function $f_-(x,y)=x-y$, point that minimizes the function $f_-(x,y)=x-y$, point that maximizes the function $f_+(x,y)=x+y$, point that minimizes the function $f_+(x,y)=x+y$. x and y in the above functions are respectively the x coordinate and the y coordinate. These eight extreme points constitute the vertexes of the initial convex hull.

Step 2: *Threshold A: Neglect inner points by the centroid*. The distance between the centroid of the point set and an edge of the initial convex hull is called partition distance. If the distance from a point to the centroid is shorter than the shortest partition distance *OMIN*, the point is inside the convex hull and it will be neglected (see Fig. 3).

Step 3: *Threshold B: Exclude inner points near the boundary of the convex hull*. If the distance from a point to O is shorter than the partition distance of the partition which the point locates in, the point is inside the convex hull, and it will be excluded (see Fig. 3).

Step 4: *Threshold C: Analyze inner points attached to the boundary*. O is the centroid, A, B are two neighboring vertexes of the convex hull, we call the partition between the direction from O to A and the direction from O to B as partition *ABO*. If a point locates at the partition *ABO*, it will be checked if it is inside the triangle *ABO*. If it is inside the triangle *ABO*, it will be excluded (see Fig. 3). Otherwise, it will be put into a dataset corresponding to its partition for the following process.

Step 5: *Renew the convex hull*. Two new datasets are derived from the original dataset in the corresponding partition. The new datasets are processed by Steps 3, 4 and 5 recursively until no new dataset is derived.

The point comparison of step 4 of VAICH is the same as that of Quickhull. However, the point comparison of Quickhull is much more complex than that of Graham scan algorithm.

The point comparison of Graham scan algorithm is as follows: After point location, let A, B be two neighboring convex hull vertexes on the order of counter-clockwise (see Fig. 4), point C be a point which locates between the direction from the centroid O to A and the direction from O to B. If the angle *ACB* (edge AC and BC are on the order of clockwise) is greater than π, the angle *ACB* is concave, point C is inside the convex hull and it should be excluded. Otherwise, point C is a candidate of the vertexes of the convex hull.

It just needs the judgment on the sign of the cross product of $\overrightarrow{CA}$ and $\overrightarrow{CB}$ to decide whether point C is inside the convex hull. In another way, if the equation of the edge *AB* is $ax+by+c=0$, and the coordinate of point C is (x_i, y_i), we can judge whether point C is inside the convex hull by the sign of ax_i+by_i+c. Thus, we simplify the point comparison from the analysis of three edges of a triangle to the analysis of one edge.