Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

A linear-time algorithm to compute the triangular hull of a digital object

Apurba Sarkar [a], Arindam Biswas [b,*], Mousumi Dutt [c], Partha Bhowmick [d], Bhargab B. Bhattacharya [e]

[a] Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India
[b] Department of Information Technology, Indian Institute of Engineering Science and Technology, Shibpur, India
[c] Department of Computer Science and Engineering, International Institute of Information Technology, Naya Raipur, India
[d] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
[e] Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India

ARTICLE INFO

Article history:
Received 8 April 2015
Received in revised form 2 December 2015
Accepted 22 January 2016
Available online xxxx

Keywords:
Triangular hull
Triangular convex hull
Triangular polygon
Shape analysis
Combinatorics

ABSTRACT

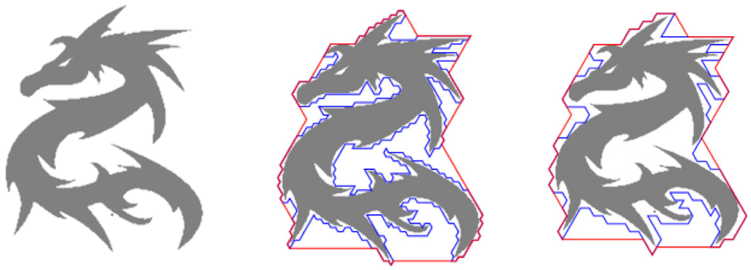
A linear-time algorithm for determining the triangular hull of a digital object that is digitized with a uniform triangular-grid scan, is presented in this paper. A triangular hull consists of a sequence of edges on the underlying triangular grid $\mathbb{T}$ consisting of three sets of parallel grid lines that are inclined at 0°, 60°, and 120° w.r.t. the x-axis. The proposed algorithm determines the triangular hull of a given object on the basis of certain geometrical properties of the edge-sequence observed along its boundary. The approach is purely combinatorial in nature as opposed to other conventional algorithms used for computing the convex hull such as those based on divide-and-conquer or line-sweep. The running time of the algorithm is linear on the number of pixels on the perimeter of the object. Also, by using a more sparse grid, i.e., by increasing the grid unit, the number of perimeter-pixels, and in turn, the running time of the algorithm can be reduced proportionately. The algorithm is tested extensively on several test cases and experimental results and analysis are presented.

1. Introduction

A polygon P is said to be convex if P is non-intersecting, and for any two points p and q on the boundary of P, the segment pq lies entirely inside P. The convex hull of an object A, denoted by $CH(A)$, is the smallest convex polygon that contains all the points of A. There exist a number of algorithms in the literature to find the convex hull of a point set or a polygonal object having arbitrary shape on the real plane. The running time of some of the existing algorithms are $O(n^3)$ (brute force), $O(n \log n)$ [6], $O(nh)$ [8], and $O(n \log h)$ [10], where, n is the number of points/vertices of A, and h is the number of vertices of $CH(A)$. Also, there are other algorithms for finding the convex hull, e.g., [2,4,15]. A detailed analytical study of convex hull algorithms may be seen in [1]. However, because of the inherent complexities, these algorithms do not perform well for sufficiently large digital objects and hence, they are unsuitable for real world applications. For example, in gift wrapping algorithm, from a point p_i on the hull, the next point p_{i+1} is determined by comparing polar angles of all points with respect to point p_i considering it as the center of polar coordinates. Computing and comparing these polar angles involve trigonometric operations. Graham scan, on the other hand, uses the sign of 3×3 determinants formed by three

* Corresponding author.
E-mail addresses: as.besu@gmail.com (A. Sarkar), barindam@gmail.com (A. Biswas), duttmousumi@gmail.com (M. Dutt), bhowmick@gmail.com (P. Bhowmick), bhargab@isical.ac.in (B.B. Bhattacharya).

http://dx.doi.org/10.1016/j.dam.2016.01.014



(a) (b) (c)

Fig. 1. (a) A digital image and its triangular hulls for (b) $g = 4$ and (c) $g = 8$.

consecutive points p_{i-1}, p_i, p_{i+1} to determine whether there is a left-turn, right-turn, or no turn at p_i. Computation of this determinant requires multiplication along with comparison and addition/subtraction. Biswas et al. [3] recently proposed a combinatorial algorithm to find the orthogonal hull of a digital object (Definition 1) in isothetic grid whose time complexity depends on the object perimeter instead of the object area. They were also able to control the precision and complexity of the hull by varying the grid spacing. However, there is no algorithm to find convex hull of a digital object in the triangular grid. Convex hull algorithms are widely used both in theory and application; in fact, convex hull algorithms are fundamental as many algorithms on computational and digital geometry are based on them. There are many variations of computing convex hull algorithm. The relative convex hull algorithm is proposed in [11], which is a special subject in computational geometry (shortest paths), in image analysis (calculation of features), in robotics (shortest path of a robot in a constrained environment), and so forth. Computation of convex hulls of spheres with a constant number of distinct radii, and convex hulls of a constant number of disjoint convex polytopes are discussed in [9]. The creation of polyhedral approximations for certain kinds of digital objects in a three-dimensional space is presented in [14], where objects are represented as sets of voxels. The proposed algorithm in this paper generates the convex hull of a given object and modifies the hull afterwards by recursive repetitions. The convex hull of a spherically symmetric sample is discussed in [7], where some new asymptotic results are presented. Convex hull algorithms in 2D or 3D depend on the coordinates of sample points. An approximation algorithm named FVDM is proposed in [17], which only utilizes the information of the distance matrix of samples to find the convex hull. A novel method of convex–concave hull is stated in [13] for support vector machine (SVM) classification. In [16], a hybrid convex hull algorithm is proposed to detect finger tips directly from a binary image without going through the edge detection process. The problem addressed here is the following: given a 2D digital object S imposed on a background triangular grid $\mathbb{T}$ (Definition 2) consisting of three sets of parallel *grid lines*, which are inclined at 0°, 60°, and 120° w.r.t x-axis, find a minimum area triangular convex polygon (i.e., triangular convex hull) containing the object S. A triangular grid, also known as isometric grid [5], is obtained by tiling the plane regularly with equilateral triangles. It may be noted here that hexagonal grid is the dual of the triangular grid and vice versa [5]. The resolution and complexity of the triangular hull of a given digital object can be controlled by changing the grid spacing, thus enabling it to meet the requirement of a specific application. It may be noted that the triangular hull depends on the registration of the digital object S with the grid $\mathbb{T}$, i.e., for the same object with different registration the resulting hull may be different. Fig. 1(a) shows a digital object and Fig. 1(b) and (c) show triangular hulls for two different grid sizes, $g = 4$ and $g = 8$, respectively.

2. Definitions and preliminaries

Definition 1 (*Digital Object*)**.** A *(digital) object* is a finite subset of $\mathbb{Z}^2$ consisting of one or more $k(= 4$ or $8)$-connected components [12].

In this work, the object is considered as a single 8-connected component.

Definition 2 (*Triangular Grid*)**.** A triangular grid (henceforth simply referred as grid) $\mathbb{T} := (\mathbb{L}_0, \mathbb{L}_{60}, \mathbb{L}_{120})$ consists of three sets of parallel grid lines, which are inclined at 0°, 60°, and 120° w.r.t. *x-axis*.

The grid lines in $\mathbb{L}_0, \mathbb{L}_{60}, \mathbb{L}_{120}$ correspond to three distinct coordinate axes, namely α, β, γ. Three grid lines, one each from $\mathbb{L}_0, \mathbb{L}_{60}, \mathbb{L}_{120}$, intersect at a (real) *grid point*. The distance between two consecutive grid points along a grid line is termed as *grid size*, g. A line segment of length g connecting two consecutive grid points on a grid line is called *grid edge*. The smallest-area triangle formed by three grid edges, one each from $\mathbb{L}_0, \mathbb{L}_{60}, \mathbb{L}_{120}$, is called *unit grid triangle* (*UGT*). A portion of the triangular grid is shown in Fig. 2. It has six distinct regions called sextants, each of which is well-defined by two rays starting from (0, 0, 0). For example, Sextant 1 is defined by the region lying between $\{\beta = \gamma = 0, \alpha \geq 0\}$ and $\{\alpha = \gamma = 0, \beta \geq 0\}$. One of α, β, γ is always 0 in a sextant. For example, $\gamma = 0$ in Sextant 1 and Sextant 4. For a given grid point, p, there are six neighboring *UGT*s, given by $\{T_i : i = 0, 1, \ldots, 5\}$. The three coordinates of p are given by the corresponding moves along a/the shortest path from (0, 0, 0) to p, measured in grid unit. For example, (1, 2, 0) means a unit move along 0° followed by two unit moves along 60°, starting from (0, 0, 0). The grid point p can have six neighbor grid points, whose direction codes are given by $\{d : i = 0, 1, \ldots, 5\}$.

Please cite this article in press as: A. Sarkar, et al., A linear-time algorithm to compute the triangular hull of a digital object, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.01.014