Advances in Engineering Software 53 (2012) 72–78

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

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

A fast algorithm for approximate surface reconstruction from sampled points

B. Repnik *, B. Žalik

University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, SI-2000 Maribor, Slovenia

ARTICLE INFO

Article history:
Received 7 November 2011
Received in revised form 11 July 2012
Accepted 10 August 2012
Available online 6 September 2012

Keywords:
Algorithms
Computational geometry
Point clouds
Surface reconstruction
Voxels
Error estimation

ABSTRACT

This paper presents a new algorithm for fast approximate surface reconstruction from sampled points. This algorithm works over three steps. Firstly, a raw rectangular surface is obtained, and then this surface is triangulated and smoothed in the second step. The surface vertices are fitted to the nearest input points at the end. The algorithm is very fast, numerically stable, easy to implement, and it constructs a watertight surface. In the experimental section, the algorithm is compared with other available algorithms (algorithm from CGAL library, Power Crust, Tight cocone, and Poisson reconstruction) in regards to the spent CPU time. Finally, an error of the obtained approximate surface is empirically estimated.

1. Introduction

Today, different technologies including laser scanners, structured light scanners, and optical position scanners exist to gather points from various 3D objects [1]. Unfortunately, obtained sets of structurally unrelated points are unsuitable for further processing, as they do not meet the conditions of well-formed surfaces necessary for the unambiguous representation of scanned objects [2]. An automatic process of connecting the scanned points to form the object's surface is known as a surface reconstruction.

This paper presents a new approximate algorithm for surface reconstruction from sampled points. The algorithm firstly arranges the input points into an uniform grid that enables the generation of an approximate rectangular surface. This surface is triangulated and smoothed. At the end, the surface vertices are fitted onto the nearest input points. This algorithm is robust, stable, simple to implement, and very fast. It always produces a watertight object and is therefore suitable for fast previewing and rapid prototyping [3]. Although some deviation in the points' distribution are tolerated, the proposed algorithm works the best with equally-spaced points.

The paper is organized as follows. Section 2 contains a brief overview of previous work. The approach is introduced in Section 3. The estimation of the algorithm's time complexity and the quality of the approximation are given in Section 4. Section 5 concludes the paper.

2. Previous work

The problem of surface reconstruction from sampled points has attracted huge interest in the past, which started with the work of Hoppe et al. [4], who classified each point with a signed distance to the surface. A similar algorithm was later proposed by Curless and Levoy [5]. One of the earliest overviews of the surface reconstruction methods was done in 1998 by Mencl and Müller [6]. Many algorithms initially construct Delaunay tetrahedronization from the sampled points. For example, the α-shapes algorithm [7] removes all tetrahedrals that are outside a sphere with the radius $1/\alpha$. The main drawback is when selecting of the optimal value for parameter α (too small a value leads to non-connected surfaces, too large a one neglects details). The approach proposed by Boissonat [8] also employs Delaunay tetrahedronization. Using simple rules, any unnecessary tetrahedra being located on the convex shell of the input points are eliminated. Unfortunately, this approach is only suitable for uniformly-sampled points. Veltkamp [9] uses γ-indicator to adopt the elimination criterion to non-uniformly sampled points. γ-indicator is defined as $1 - r/R$, where r is the radius of the circumcircle of the considered tetrahedron's face and R the radius of circumcircle of the tetrahedron sphere. An automated approach for determining the α value was proposed by Xu and Harada [10]. They developed an approach based on the points' density to scale the variable α on the fly. A more well-known approach, named Crust, was presented by Amenta et al. [11]. This method has a strong theoretical background, that guarantees the correct reconstruction. Unfortunately, Crust is computationally demanding, since it has to build a 3D Voronoi diagram and

* Corresponding author.
E-mail addresses: blaz.repnik1@uni-mb.si, blazrepnik@gmail.com (B. Repnik).

0965-9978/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2012.08.003

a 3D Delaunay tessellation, which also increases the possibility of numerical instability. The improved version of the basic algorithm, which also copes well with the noise, is Power Crust [12]. A generalization of Crust was later proposed by Dey and Goswami [13]. A noise resistant algorithm for reconstructing a watertight surface, also based on Delaunay tetrahedronization, was introduced by Kolluri et al. [14]. They used a variant of the spectral graph partitioning to decide whether the considered tetrahedron is inside or outside of the original object. The spectral-graph is a powerful tool for the global viewing of a model for eliminating noise, outliers and to efficiently deal with undersampled regions. A ball-pivoting algorithm was presented by Bernardini et al. [15]. Starting from the seed triangle, the ball pivots around an edge until it touches another point, which then forms a new triangle. The major advantage of this algorithm is that it does not need to construct the Delaunay tetrahedralization. A greedy algorithm for surface reconstruction was presented by Cohen-Steiner and Da [16]. The algorithm starts from a seed triangle and adds Delaunay triangle one by one to the piecewise surface. A similar approach with competing evolving fronts was proposed by Sharf et al. [17]. These fronts adapt to the local feature size of the target shape in a coarse-to-fine manner. The method is fast and guarantees a watertight surface. An interesting topological approach for surface reconstruction was presented by Biscaro et al. [18]. This approach strongly relies on topological information that reduces the computational time.

The more related approaches proposed in this paper are those works introduced by Hornung and Kobbelt [19], Mullen et al. [20], Bolitho et al. [21], and Volodine et al. [22]. Hornung and Kobbelt [19] proposed a method for the watertight reconstruction of non-uniformly sampled points without normal information. At first, they derive a confidence map in the vicinity of the input points. The confidence map represents the probability that the unknown surface passes through a certain part of the 3D space. A closed surface with maximum confidence is then extracted by transforming their combinatorial optimization problem into a max-flow/min-cut problem of an embedded graph. Mullen et al. [20] proposed a method for robust surface reconstruction from point sets. An unsigned distance function to the input data is calculated at first. An ϵ-band to eliminate the outliers and noise is then determined. A global stochastic sign estimation of the distance first outside then inside the ϵ-band is computed. Finally, the estimation is smoothed to obtain a smooth and closed surface. Any possible unwanted holes on the surface are repaired during this step, too. Bolitho et al. [21] showed that the Poisson surface reconstruction can be performed as a sequence of streaming passes over and out-of-core octree representation. This reconstruction includes several steps: preprocessing (aligning the dominant axis of the point cloud covariance matrix with the x-axis), octree construction, vector field construction, divergence computation, Poisson system solving, isovalue computation, and finally an isosurface extraction. Volodine et al. [22] proposed an algorithm that constructs an interpolating triangular mesh. A structure called the Delaunay cover is constructed at first. It represents a barrier between the interior and exterior of the object. Using the Delaunay cover, a Boolean voxel grid is constructed in the next step. In this way, the empty and non-empty voxels are determined. The outer surface of the voxel grid is then snapped to the point cloud.

3. The method for approximate surface reconstruction from sampled points

The proposed approximate surface reconstruction algorithm (ASRA) works over the following steps:

- obtaining a rough approximate rectangular surface of the model,

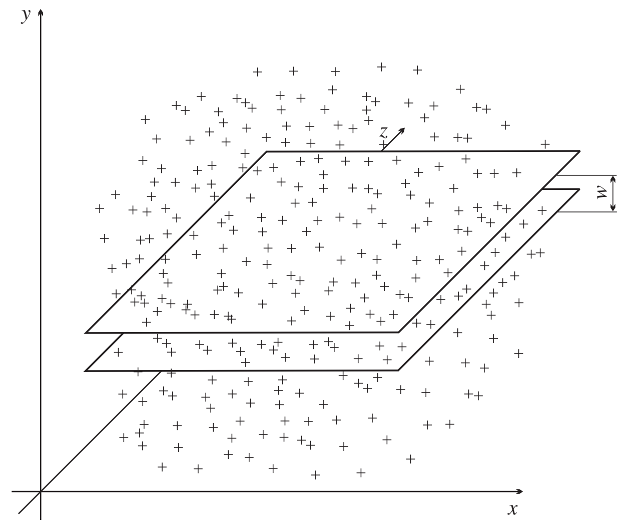

Fig. 1. Bounding box of input points is divided into slices of the width w.

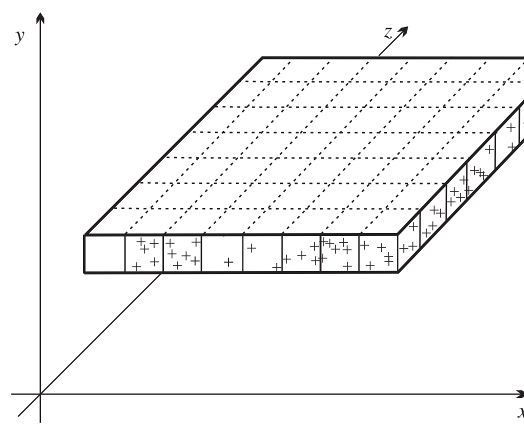

Fig. 2. Slice is divided into equally-sized cells.

- smoothing and triangulating the approximate surface, and
- fitting the surface vertices to the nearest input points.

3.1. Obtaining a rough approximate rectangular surface of the model

Firstly, the number n of input points p_i, $i = 1,\ldots,n$, and their bounding box B is determined. The bounding box is then translated into the origin and normalized according to:

$$B = \left(\frac{x_{max}}{d}, \frac{y_{max}}{d}, \frac{z_{max}}{d}\right) \tag{1}$$

where $d = max\{x_{max}, y_{max}, z_{max}\}$. B is finally aligned with its longest side with the Y coordinate axis. After that, B is divided into slices, parallel to the XZ-plane (in Fig. 1, only one slice is plotted).

The number of slices s is determined by the following heuristics

$$s = k\sqrt[3]{n} \tag{2}$$

where k is determined experimentally (see Section 4.4). The slice width w is trivially obtained as

$$w = \frac{1}{s} \tag{3}$$

Each slice is divided into cells in the form of a cube with the size w (Fig. 2). The slice can be considered as a 2D raster of $x \times y$, (x, $y \leqslant s$) pixels, where those cells that contains points represent black pixels. As this is an approximative surface reconstruction