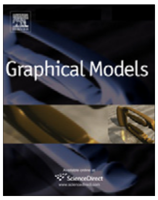
Graphical Models 72 (2010) 35–45

Contents lists available at ScienceDirect

Graphical Models

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



Sparse grid distance transforms

Takashi Michikawa *, Hiromasa Suzuki

RCAST, The University of Tokyo, Japan

ARTICLE INFO

Article history:
Received 9 October 2008
Received in revised form 13 April 2010
Accepted 2 May 2010
Available online 12 May 2010

Keywords:
Distance fields
Volume graphics
Divide-and-conquer algorithm

ABSTRACT

We present a Sparse Grid Distance Transform (SGDT), an algorithm for computing and storing large distance fields. Although SGDT is based on a divide-and-conquer algorithm for distance transforms, its data structure is quite simplified. Our observations revealed that distance fields can be recovered from distance fields of sub-block cluster boundaries and the binary information of the cluster through a one-time distance transform. This means that it is sufficient to consider only the cluster boundaries and to represent clusters as binary volumes. As a result, memory usage is less than 0.5% the size of raw files, and it works in-core.

1. Introduction

Distance transform (DT) [1,2] is known as a method of computing distance fields from volume models. Distance field is a scalar field defined by the distance function $d(\mathbf{p})$ for each cell $\mathbf{p} \in R^3$ in Eq. (1)

$$d(\mathbf{p}) = \min \|\mathbf{p} - \mathbf{q}\|, \tag{1}$$

where $\mathbf{q}$ denotes a cell on the boundary of an object. Applications for distance fields can be found in areas such as medial surface extraction, solid modeling and morphology operations. This paper focuses in particular on computing distance fields from large volume models.

Our work is motivated by convergence engineering based on X-ray CT images [3]. X-ray CT scanners can measure mechanical objects with high resolution, and the images produced are used in a range of applications such as simulation and comparison with CAD models. Constructing distance fields from such CT images is also important, as there are various applications for them. For instance, the topological features of complex objects can be extracted by computing the medial axis of images [4]. However, as the resolution of CT images increases and it becomes harder to apply conventional methods to them, there is a need for discussion of large distance field computation.

There are many methods of computing approximate and exact distance fields from binary volumes, and these methods are known as distance transforms (DT) [1,2]. However, it is hard to compute distance fields from large volumetric models using these methods due to the limitations of RAM size, with common 32-bit computers being able to allocate only up to 2 GB.[1] Distance fields are represented by uniform grid sampling, and each sampling cell contains a distance value. If the distance is represented by a floating-point number of four bytes, the limitation becomes 0.5 G cells (approx. 800^3), so conventional methods cannot compute distance fields from a volume of data with over a billion cells. Although 64-bit computers extend this limitation somewhat,[2] the difference is in fact small as the volume size increases by the cubic order of the resolution.

---

* Corresponding author. Address: 4-6-1 Komaba, Meguro, Tokyo 153-8904, Japan.
E-mail address: michi@den.rcast.u-tokyo.ac.jp (T. Michikawa).

[1] Some Linux OS supports up to 4GB, however memory issue still remains.

[2] In theory, 64-bit CPU can use memory addresses up to 2^{64} bytes. However, with current PCs it is impossible to support such large amounts RAM.

1524-0703/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.gmod.2010.05.001

This paper proposes a Sparse Grid Distance Transform (SGDT), an in-core distance transform for large volumetric models. SGDT consists of the two new ideas of Cluster-wise DT (CWDT) and sparse grid data structure. CWDT is a DT algorithm to compute a distance field by splitting a large volume into clusters (subvolumes), applying DT to each one and integrating the distance fields of those clusters. As SGDT needs to store only one cluster at a time, it can drastically reduce the amount of memory usage. It was used to construct an out-of-core distance transform algorithm (OoCDT) in our previous work [5].

The sparse grid is a data structure to represent a distance field split into clusters by distance fields on the boundaries between the clusters. This idea came from our observation that distance fields can be recovered from the distance fields of cluster boundaries and the binary information of the cluster through a one-time distance transform (Fig. 1). This means that it is sufficient to consider only cluster boundaries and to represent clusters as binary volumes. Indeed, its size is as little as 0.5% that of raw files and it can be worked without bulk memory.

Thus, by combining those two ideas of CWDT and sparse grid, SGDT can compute a distance field for a large volume with a very small amount of memory.

The main contribution of this paper is to provide an in-core algorithm for distance transforms to compute large distance fields. The advantages of our method include:

- It does not require large amounts of memory and large distance fields can be computed using common 32-bit computers.
- Since CWDT is independent of other clusters, speed can be increased using a multi-core processor or distributed environments.
- It can integrate any other distance transform schemes not only wavefront algorithm but also sequential algorithms, fast marching methods and level set methods [6,7].
- Our framework can be applied to n-D volumes.

This paper consists of seven sections: Section 2 reviews related work, Section 3 describes cluster-wise distance transform (CWDT) algorithm, Section 4 describes the Sparse Grid Distance Transform (SGDT) algorithm, an extension of CWDT, Section 5 describes implementation issues in multi-thread environments, Section 6 reports the experimental results and evaluates our methods, and Section 7 concludes the paper.

2. Related work

Distance field computation is a widely studied topic in the area of computer graphics and image processing. The brute-force method is simple; for all cells $\mathbf{v}$, $d(\mathbf{v})$ is computed by finding the minimum distance to all object boundary cells. Although the $d(\mathbf{v})$ value obtained is accurate and easy to make out-of-core, the cost of computation is very high (roughly $O(mn)$, where n and m denote the number of cells and the number of object boundary cells, respectively). The related work therefore seeks to reduce the costs of computation while maintaining accuracy.

Excellent surveys can be found in the work of [1] and [2]. There are many methods to compute distance fields from polygonal meshes, implicit surfaces, parametric surfaces, volumes etc. Voxelization can convert these to volumetric models, so we focus in particular on distance transforms and distance field computation from binary volumetric models.

Chamfer Distance Transform (CDT) computes distance fields using distance templates, giving a set of distances to neighboring cells. Given binary volumes, CDT can compute distance fields in two ways. One of these is by sweeping[8–11], which estimates from the top to the bottom of volumes and vice versa. The computation cost of the sweeping method is $O(n)$. The other technique is the wavefront method [12,13], which estimates from the object boundary to the front. This is similar to Dijkstra's shortest-path algorithm. The computation cost is $O(n\log n)$, because a priority queue is required to find the minimum distance in unestimated cells. Vector Distance Transform (VDT) [14–16] improves issues related to accuracy. VDT estimates the difference vector to the closest vector instead of the distance, and evaluates the distance in the final steps. VDT can therefore compute precise distance fields, and also supports sweeping and wavefront schemes. The sweeping scheme in VDT [16] requires eight passes for estimation.

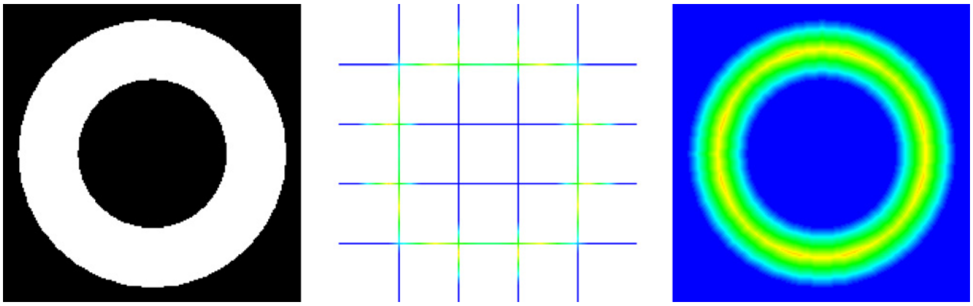

(a) Binary (b) Sparse grid (c) Result

Fig. 1. The concept of sparse grid distance transforms. With a binary volume of volume data and distance fields of cluster interfaces (cluster boundaries), distance fields can be recovered through a one-time distance transform for each cluster.