Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2018-0130

Vol. **26**, No. 3, pp. 654-680
September 2019

Fast Distance Fields for Fluid Dynamics Mesh Generation on Graphics Hardware

Alo Roosing[1,*], Oliver Strickson[1] and Nikos Nikiforakis[1]

[1] *Laboratory for Scientific Computing, Cavendish Laboratory, Department of Physics, University of Cambridge, Cambridge CB3 0HE, United Kingdom.*

Received 4 May 2018; Accepted (in revised version) 10 October 2018

Abstract. We present a CUDA accelerated implementation of the Characteristic/Scan Conversion algorithm to generate narrow band signed distance fields in logically Cartesian grids. We outline an approach of task and data management on GPUs based on an input of a closed triangulated surface with the aim of reducing pre-processing and mesh-generation times. The work demonstrates a fast signed distance field generation of triangulated surfaces with tens of thousands to several million features in high resolution domains. We present improvements to the robustness of the original algorithm and an overview of handling geometric data.

AMS subject classifications: 68U05, 68U20

Key words: Signed distance field, GPU, CUDA, mesh generation, fluid dynamics.

1 Introduction

Signed distance fields (SDF) find uses in domains from computer graphics [2] to numerical modelling [3]. Determining the location of explicit or implicit surfaces in grids or generating meshes to describe objects is an area of active research in many computational paradigms. Triangulated surfaces are a popular working medium and the Stereolithography (STL) file format finds wide use in areas such as CFD [4] and 3D printing [14]. The quick generation of robust signed distance fields from triangulated surfaces is then of great interest to many industries and academic disciplines.

Often it is necessary to know only the distance to the surface within a small region around the geometry and narrow band SDFs are useful for quickly generating just the intersection between a computational mesh and an object. This finds application in embedded boundary methods in computational fluid dynamics where generating object data

*Corresponding author. *Email addresses:* ar694@cam.ac.uk (A. Roosing), ots22@cam.ac.uk (O. T. Strickson), nn10005@cam.ac.uk (N. Nikiforakis)

often takes a significant portion of the simulation set up time, which can become a bottleneck in fast prototyping when the subsequent numerical work is highly optimised and run on many-core architectures. For example, the signed distance field of a complex car body, as shown in Fig. 1, can be used to generate cut cells in a regular computational mesh to impose boundary conditions along a detailed perimeter without introducing significant mesh generation overhead or complex connectivity information. We focus on the generation of narrow band signed distance fields inside Cartesian grids but the algorithm discussed in this paper is potentially extendible to other paradigms.

Our main aim is to describe a robust algorithm to speedup the generation of level sets from triangulated surface information using graphics processing units (GPUs). In this paper we discuss the implementation and adjustment of the Characteristic/Scan Conversion (CSC) algorithm originally described by Mauch [5]. We will outline improvements to the original approach and present an implementation on GPUs with a focus on how to manage information about many thousands of connected features.

Park et al. [7] have developed an algorithm for generating signed distances on the GPU for hierarchical grids. They sample mesh cells based on the complexity of the surface geometry and present a good speedup compared to identical approaches on the CPU. Their use of angle-weighted pseudonormals at surface discontinuities is similar to the strategy we employ.

Sud et al. [11] describe a GPU signed distance field method based on Voronoi cells and slicing. Their speedup stems from the use of GPUs, culling far away features and clamping the rasterisation of the Voronoi cells. Though their approach is different from ours, the strategy of reducing calculations is similar to the current work. Their method does not store information about the connectivity of triangles and uses the CSC algorithm for suitable sub-problems, developing a new approach for problematic surface configurations. Our implementation is purely CSC based and addresses many of these geometric cases.

Sigg et al. [10] present a GPU implementation of the CSC algorithm for triangulated surfaces. Their work is focused on overcoming the need for vertex extrusions by combining edge and face extrusions. This is done in order to reduce the workload as well as avoid topological cases which the CSC algorithm finds problematic. Below we discuss a different methodology for the issues arising at vertices.

An implementation of the CSC algorithm also exists by Mauch [20]. We use some of the insights of that code but have developed an independent strategy with updated feature generation, a high degree of parallelism and algorithmic improvements.

There is a lack of discussion in existing literature about how to best organise STL features for use with the CSC algorithm on GPUs. Specifically, it is not immediately clear how to efficiently produce extrusions from nearby surface triangles when no strict feature order is imposed in the input file. There are also gaps in the literature when it comes to discussing some complex cases that can arise in common geometries such as saddle vertices and other configurations discussed below. The main contributions of this paper are describing the efficient handling of STL features on GPUs, showing robust