

Parallel Delaunay triangulation based on Lawsons incremental insertion

Coll, Narcis; Guerrieri, Marite Ethel

XIV Spanish Meeting on Computational Geometry, 27-30 June 2011

Parallel Delaunay triangulation based on Lawson's incremental insertion

Narcis Coll¹, Marite Guerrieri¹

¹ Departament d'Informàtica i Matemàtica Aplicada, Universitat de Girona
coll@iris.udg.edu, marite@matma.udg.edu

Abstract. In this paper we present an ongoing research work on Delaunay triangulations using GPU-based algorithms. The proposed algorithm is based on Lawson's incremental insertion, taking special care to avoid concurrent insertion of points in triangles and conflicts between edge swaps.

Introduction

Graphics Processing Units (GPUs) are specialized processors which use a highly parallel structure that makes them perfect for solving problems that can be partitioned into independent and smaller parts. The development of CUDA (Compute Unified Device Architecture) and some programming languages such as OpenCL (Open Computing Language), makes GPUs attractive to solve problems in a parallel way.

Delaunay triangulation is one of the fundamental topics in Computational Geometry and it is used in many areas such as terrain modelling, finite element methods, pattern recognition, computer graphics, data interpolation, robotics, etc. Although the 2D Delaunay triangulation can be computed in $O(n \log n)$ time, where n is the number of input points, it still consumes a lot of time, especially for current applications that often need to work with millions of points. In such cases, a parallel algorithm is necessary to accelerate its computation.

In Lawson's incremental algorithm, when a point is inserted, the triangle that contains it is found, and three new edges are inserted to attach the new vertex to the vertices of the containing triangle. Next, a recursive procedure tests whether the new vertex lies within the circumcircles of any neighbouring triangles. Each affirmative test triggers an edge flip that removes a locally non-Delaunay edge. Each edge flip reveals two additional edges that must be tested. Thus, two points lying in the same triangle cannot be inserted at the same time, and two points lying in different triangles can not be processed independently if they share neighbours. In this paper we propose a GPU-based algorithm in OpenCL language that avoids concurrent insertion of points in triangles and conflicts between edge swaps.

1 Related work

Several methods [3, 4, 7, 8] for parallel Delaunay triangulation use the divide and conquer strategy to construct partial Delaunay triangulations in subregions, and finally merge these partial triangulations to get the result. However, this kind of algorithms have the following drawbacks: First, the merge phase is quite complex because it involves not only building the faces connecting the subregions but also correcting of existing faces in the

¹Partially supported by research grant TIN2010-28894-C02-02.