Robust and Efficient Delaunay Triangulations of Points on Or Close to a Sphere*

Manuel Caroli[1], Pedro M.M. de Castro[1], Sébastien Loriot[1], Olivier Rouiller[1], Monique Teillaud[1], and Camille Wormser[2]

[1] INRIA Sophia Antipolis – Méditerranée, France
{Manuel.Caroli,Pedro.Machado,Monique.Teillaud}@sophia.inria.fr
[2] ETH Zürich, Switzerland
Camille.Wormser@inf.ethz.ch

Abstract. We propose two ways to compute the Delaunay triangulation of points on a sphere, or of *rounded* points close to a sphere, both based on the classic incremental algorithm initially designed for the plane. We use the so-called space of circles as mathematical background for this work. We present a fully robust implementation built upon existing generic algorithms provided by the CGAL library. The efficiency of the implementation is established by benchmarks.

1 Introduction

The CGAL project [3] provides users with a public discussion mailing list, where they are invited to post questions and express their needs. There are recurring requests for a package computing the Delaunay triangulation of points on a sphere or its dual, the Voronoi diagram. This is useful in many domains such as geology, geographic information systems, information visualization, or structural molecular biology, to name a few. An easy and standard solution to the problem of computing such a Delaunay triangulation consists in constructing the 3D convex hull of the points: They are equivalent [13,38]. The convex hull is one of the most popular structures in computational geometry [20,11]; optimal algorithms and efficient implementations are available [1,2].

Another fruitful way to compute Delaunay on a sphere consists of reworking known algorithms designed for computing triangulations in $\mathbb{R}^2$. Renka adapts the distance in the plane to a geodesic distance on a sphere and triangulates points on a sphere [37] through the well-known flipping algorithm for Delaunay triangulations in $\mathbb{R}^2$ [30]. As a by-product of their algorithm for arrangements of circular arcs, Fogel et al. can compute Voronoi diagrams of points lying exactly on the sphere [26]. Using two inversions allows Na et al. to reduce the computation of a Voronoi diagram of sites on a sphere to computing two Voronoi diagrams

* This work was partially supported by the ANR *(Agence Nationale de la Recherche)* under the "Triangles" project of the *Programme blanc* ANR-07-BLAN-0319
http://www.inria.fr/geometrica/collaborations/triangles/

P. Festa (Ed.): SEA 2010, LNCS 6049, pp. 462–473, 2010.

in $\mathbb{R}^2$ [33], but no implementation is available. Note that this method assumes that data points are lying exactly on a sphere.

As we are motivated by applications, we take practical issues into account carefully. While data points lying exactly on the sphere can be provided either by using Cartesian coordinates represented by a number type capable of handling algebraic numbers exactly, or by using spherical coordinates, in practice data-sets in Cartesian coordinates with double precision are most common. In this setting, the data consists of rounded points that do not exactly lie on the sphere, but close to it.

In Section 4, we propose two different ways to handle such rounded data. Both approaches adapt the well-known incremental algorithm [12] to the case of points on, or close to the sphere. It is important to notice that, even though data points are rounded, we follow the exact geometric computation paradigm pioneered by C. K. Yap [39]. Indeed, it is now well understood that simply relying on floating point arithmetic for algorithms of this type is bound to fail (see [29] for instance).

The first approach (Section 4.1) considers as input the projections of the rounded-data points onto the sphere. Their coordinates are algebraic numbers of degree two. The approach computes the Delaunay triangulation of these points exactly lying on the sphere.

The second approach (Section 4.2) considers circles on the sphere as input. The radius of a circle (which can alternatively be seen as a *weighted* point) depends on the distance of the corresponding point to the sphere. The approach computes the weighted Delaunay triangulation of these circles on the sphere, also known as the *regular* triangulation, which is the dual of the Laguerre Voronoi diagram on the sphere [38] and the convex hull of the rounded-data points.

These interpretations of rounded data presented in this work are supported by the space of circles [10,24] (Section 3).

We implemented both approaches, taking advantage of the genericity of CGAL. In Section 5, we present experimental results on very large data-sets, showing the efficiency of our approaches. We compare our code to software designed for computing Delaunay triangulations on the sphere, and to convex-hull software [28,35,1,6,2,37,25]. The performance, robustness, and scalability of our approaches express their added value.

2 Definitions and Notation

Let us first recall the definition of the *regular triangulation* in $\mathbb{R}^2$, also known as *weighted Delaunay triangulation*. A circle c with center $p \in \mathbb{R}^2$ and squared radius r^2 is considered equivalently as a *weighted point* and is denoted by $c = (p, r^2)$. The *power product* of $c = (p, r^2)$ and $c' = (p', r'^2)$ is defined as $pow(c, c') = \|pp'\|^2 - r^2 - r'^2$, where $\|pp'\|$ denotes the Euclidean distance between p and p'. Circles c and c' are orthogonal iff $pow(c, c') = 0$. If $pow(c, c') > 0$ (i.e., the disks