Mapping Graphs on the Sphere to the Finite Plane

Henk Bekker, Koen De Raedt

Institute for Mathematics and Computing Science, University of Groningen,
P.O.B. 800 9700 AV Groningen, The Netherlands,
bekker@cs.rug.nl, csg8042@wing.rug.nl

Abstract. A method is introduced to map a graph on the sphere to the finite plane. The method works by first mapping the graph on the sphere to a tetrahedron. Then the graph on the tetrahedron is mapped to the plane. Using this mapping, arc intersection on the sphere, overlaying subdivisions on the sphere and point location on the sphere may be done by using algorithms in the plane.

1 Introduction

In plane computational geometry three basic operations are line segment intersection, overlaying two subdivisions and point location. In a natural way these operations may also be defined on the sphere. For these problems, the correspondence between the plane and the sphere is so strong that it is tempting to try and solve the problems on the sphere by using algorithms working in the plane. That is however not possible because topologically the sphere differs from the plane. An obvious step to remedy this situation is to try and adapt the algorithms working in the plane, so that they work on a sphere. For naive and non-optimal implementations this might work. However, transforming sophisticated and optimised algorithms is not trivial. In fact, then every detail of the algorithm has to be reconsidered.

Instead of adapting algorithms we propose to adapt the problem, that is, we propose to map the graphs on the sphere to the plane, so that algorithms working in the plane may be used to solve the problems on the sphere. To make this scheme work the mapping has to fulfil three conditions.

1. The mapping has to be continuous and one-to-one.
2. The mapping has to be finite, that is, the image of the graphs on the sphere should not have points at infinity.
3. Each arc of a great circle on the sphere has to be mapped on one straight-line segments in the plane.

Let us comment on these three conditions.
1: The mapping has to be continuous because when the mapping is only piecewise continuous the image of the graph on the sphere would consist of patches in the plane. As a result, the operations in the plane would have to be done

on each of these patches. With a one-to-one mapping we mean a mapping that maps every point on one graph to a point on the other graph, so, we do not mean that the graph structure remains the same. The mapping has to be one-to-one because after the operations in the plane have been done the result has to be mapped back on the sphere in a unique way.
2:The mapping has to be finite because, in general, the algorithms working in the plane can not handle points at infinity.
3: Arcs of a great circle on the sphere have to be mapped on straight-line segments in the plane. This condition is not absolutely essential. We could use a mapping that maps an arc of a great circle on some curve segment in the plane, and then use an algorithm in the plane that works with curve segments instead of line segments. However, curve segment intersection, point location in curved graphs, and overlaying curved graphs is very inefficient compared with the corresponding straight-line segment algorithms.

In the literature no mapping is given that fulfils these three conditions. In this article we introduce a mapping that fulfils these three conditions almost. Only condition 3 is not met completely. In the mapping we propose some arcs are mapped on *two* or *three* connected line segments in the plane instead of *one* line segment. In the following section we describe the mapping in an operational way. After that we discuss some details and alternatives. As an example application we compute the overlay of two subdivisions on a sphere.

Problem motivation: During the past ages cartographers have proposed many mappings to map the sphere to the plane (e.g. Mercator-, cylindric-, and stereographic projection)[1]. None of these mappings maps an arc of a great circle on the sphere to a straight line segment in the plane. To our knowledge no such mapping exists. In this article we introduce a mapping that maps an arc of a great circle on the sphere most often to one line segment in the plane, and sometimes to two or three connected line segments in the plane. Our motivation for developing this mapping is in computational geometry. Some time ago we proposed [2] a linear time algorithm to compute the Minkowski sum of two convex polyhedra A and B. The crucial part of that algorithm consists of calculating the overlay of two subdivisions on the sphere, where the subdivisions are the slope diagrams of A and B [3]. By mapping this problem to the plane the 3D problem of calculating the Minkowski sum of two convex polyhedra is reduced to the problem of calculating an overlay in the plane. For this a linear time algorithm may be used [4], so, Minkowski addition of convex polyhedra in 3D may be done in linear time. In [5] algorithms are given to do overlay, point location and arc intersection on the sphere. However, these algorithms do not work in linear time, so, they can not be used for an efficient implementation of the Minkowski sum.

2 The mapping

We consider a sphere centered at the origin. On this sphere some structure is given consisting of points and arcs of great circles. To be a bit more concrete, and without loss of generality, in the sequel we assume that the structure is a subdivision on the sphere. We represent the subdivision by a graph SGA (Sphere Graph A). To keep things simple, we assume that the subdivision is connected, so SGA is also connected. To make SGA as similar as possible to the subdivision we embed SGA on the sphere so that nodes of SGA are embedded at positions of the corresponding points of the subdivision, and edges of SGA are embedded as arcs of great circles of the corresponding edges of the subdivision. In this way SGA and its generating subdivision are similar, so we can simply work with SGA. (See figure 2 upper.)

We want to map SGA to the plane, but as an intermediate step we first map SGA to a tetrahedron T with the following properties.

1. The base of T is parallel with the x,y plane.
2. T is almost regular.
3. T is centered at the origin.

Let us comment on these three properties.
1: T should have a unique top. This means that only one of the four vertices of T should have the maximum z coordinate in the positive z direction. By choosing the base of T in the x,y plane this condition is fulfilled.
2: We first construct a regular tetrahedron TTT with vertex coordinates
$(0,0,0)(1,0,0)(0.5,\frac{\sqrt{3}}{2},0)(0.5,\frac{\sqrt{3}}{6},\frac{\sqrt{3}}{3})$. The first three vertices are in the x-y plane, the last vertex is the top. TTT has side length 1. Now we shift the first three vertices of TTT each with a random displacement in the x,y plane over a distance of say at most 0.1, and the top vertex with a random displacement in 3D over a distance of at most 0.1. This gives us a tetrahedron TT.
3:We vectorially add the vertex positions of TT, divide this vector by four, and shift TT over minus this vector. This gives us T.

Now we are going to map SGA to T. First we map all nodes of SGA to T, and then we add additional nodes. To this end, we copy the graph SGA in a new graph TGA (Tetrahedron Graph A) but we do not copy the node positions of SGA into TGA. Using central projection we map the nodes of SGA on T as follows. For every node n of SGA we construct a ray, starting at the origin and containing the position of node n. The intersection point p of this ray with one of the faces of T is assigned to the corresponding node position in TGA.

Every node of TGA is now on T. However, every edge of TGA represents a line segment, and not every of these line segments is on a face of T. That is because there are edges the endpoints of which are on different faces of T. Suppose that edge e has an endpoint on face f_i and an endpoint on face f_j, with $i \neq j$, and that f_i and f_j meet in edge e_T. We add a node to TGA, located on e. To calculate the position of the new node we construct a ray, starting at the origin, and intersecting e and e_T. The intersection point of this ray with e_T is the position of the new node in TGA. We mark these added nodes so that they can be deleted on the way back.

Now we are going to map TGA on the plane P containing the lower face of T. Crucial for this mapping is that TGA has no node located at the top vertex of

T. If there is a node at the top vertex a new T has to be generated, and SGA has to be mapped again on T. Because T is a random tetrahedron the probability that a node is mapped at the top vertex is virtually zero.

Having verified that no node of TGA is at the top vertex of T we determine a projection point *pp*. From *pp* the graph TGA will be projected onto P. *pp* has to fulfil two conditions.

1. *pp* has to be located inside T.
2. The z coordinate of *pp* should be greater than the z coordinate of the position of the highest node of TGA.

Let us comment on this.
1: That *pp* is located inside T implies that *pp* is not on the boundary of T. So, from *pp* the graph TGA is completely seen from the inside of T, and every node and edge of TGA is seen in a unique direction.
2: Choosing *pp* higher than the highest node position of TGA has the effect that by projecting from *pp*, every node and edge of TGA is projected on P.

We construct *pp* as follows. Call the z coordinate of the top vertex of T H_{tv} and the z coordinate of the highest node of TGA H_{hn}. We construct a plane *ppp* parallel with the x-y plane with $z = (H_{hn} + H_{tv})/2$, and a line *lpp* perpendicular to the x-y plane, containing the top vertex of T. The intersection point of *lpp* and *ppp* is *pp*. See figure 1.

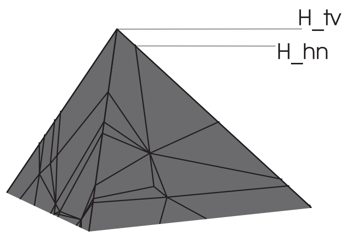

Fig. 1. The top of the tetrahedron T, the z coordinate H_{tv} of its top vertex and the z coordinate H_{hn} of the highest node of TGA. The projection point *pp* is located inside T, under the top vertex of T and at $z = (H_{hn} + H_{tv})/2$.

Using central projection from *pp* we project TGA in the plane P, resulting in graph PGA (Plane Graph A). PGA is constructed as follows. First we copy the graph TGA to PGA but not the node positions. For every node n of TGA we construct a line l containing *pp* and the node position n. The position of the intersection point of l with P is assigned to corresponding node of n in PGA.

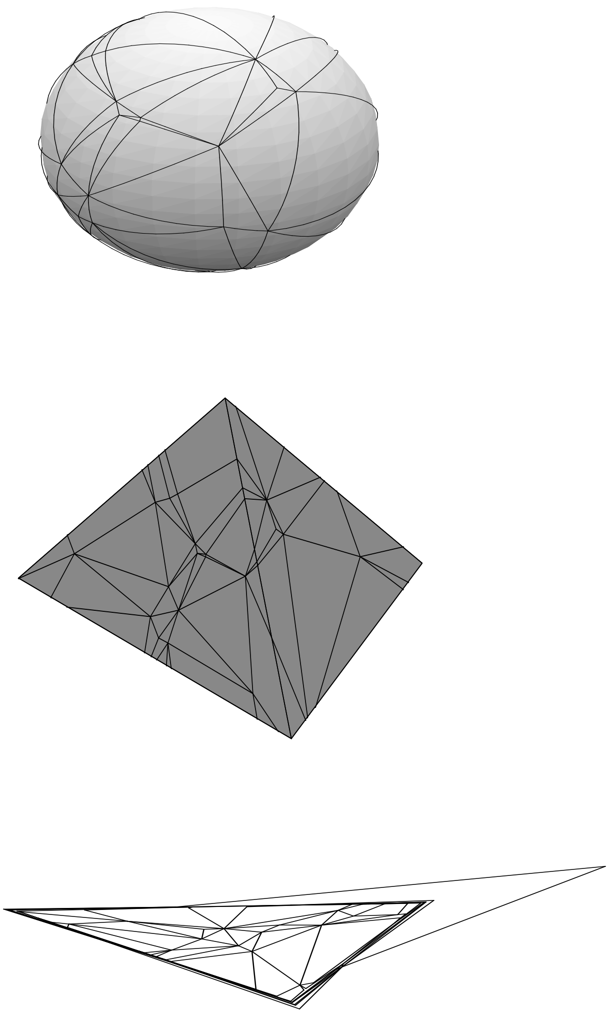

Fig. 2. The three stages of mapping a graph on a sphere to the plane. The view direction is the same for all three figures. Top: A sphere with a graph SGA. Middle: SGA mapped on a random tetrahedron T, giving the graph TGA. SGA can be recognised in TGA. Lower: The graph TGA mapped to a plane containing the lower face of T, giving the graph PGA. PGA is very inhomogeneous, and it is difficult to recognise TGA in PGA.

The process of mapping SGA on a plane may be summarised as follows.

```
repeat
  make random T;
  map SGA on T;
until (no node on top-vertex);
determine projection point pp;
project TGA on P;
```

3 Discussion

Essential parts In the method we presented four parts are essential.

1. Projecting SGA on T should be done from the center of the sphere. In this way an arc of a great circle on the sphere is mapped on T as one, two or three connected straight line segments.
2. To map SGA in a one-to-one way to T, the center of the sphere should be located inside T.
3. P should be chosen so that only one point of T has the maximal distance to P. The simplest way to fulfil this condition is to choose P so that it contains the lower face of T.
4. The projection point *pp* should be higher than the highest node of TGA and should be inside T. This has the effect that TGA is mapped as a whole, in a one-to-one way to P.

Overhead and time complexity TGA has more nodes than SGA, so, also PGA has more nodes than SGA. The number of additional nodes of TGA is proportional to the number of edges of TGA that intersect an edge of T. In 2D, the expectancy of the number of edges of a random graph with N edges intersected by a line is $\propto \sqrt{N}$. So, on the average the number of nodes in TGA is $\propto \sqrt{N}$ greater than the number of nodes in SGA. Therefore, the relative overhead $\frac{\sqrt{N}}{N}$ goes to zero as $\frac{1}{\sqrt{N}}$ when the number of edges N goes to infinity. In our experiments on randomly generated graphs, ranging from 10 to 10000 nodes, we observed that a graph SGA with 3000 nodes gives a graph TGA with $\approx 6\%$ more nodes. The time complexity of the method is linear in the number of nodes of SGA, so, also in the number of nodes of TGA and PGA. When TGA has a node at the top-vertex of T, T should be regenerated. In our experiments this never happened, so, we think the overhead related to this situation may be neglected.

Inhomogeneity of PGA It is difficult to recognise in PGA the original graph SGA. When SGA is more or less homogeneous TGA is also, but PGA is always very inhomogeneous. That is because that part of TGA that is on the lower face of T is mapped undistorted to P, while the parts of TGA that are on the other faces of T are projected on P in a direction that is almost parallel with these faces. To avoid numerical problems associated with the inhomogeneity of PGA, we implemented our algorithm in exact arithmetic, or more precise, we implemented our algorithm in C++ and LEDA [6]. When exact arithmetic is used for mapping the sphere on the plane, for the operations in the plane (for example overlaying, line intersection or point location), and for mapping back the result on the sphere, it is no problem that PGA is inhomogeneous.

4 Alternatives and improvements

Alternative position of T In section 2 we proposed to position T with its center at the origin. That is however not required. Any position of T will do as long as T contains the origin. This freedom may be used to improve the mapping somewhat. The alternative position of T is as follows. The x and y coordinates of T are the same as before, and the z position of T is chosen so that the origin is located slightly above the lower face. For example, the distance from the lower face to the origin could be chosen as $\frac{1}{100}H_T$, where H_T is the height of T in the z direction. Positioning T in this way has the effect that almost the whole lower half of SGA is mapped on the lower face of T. So, the density of edges and nodes of TGA on the other faces of T decreases. As a result, the number of edges TGA crossing edges of T decreases, so, the number of additional nodes decreases. Moreover, the density of nodes of TGA near the top vertex of T decreases, so, the probability that a node of TGA is located at the top vertex of T decreases. These effects can be seen in figure 3. The same SGA is used as in figure 2. It can be seen that the density of nodes and edges on the non-horizontal faces of T has decreased, and that the z coordinate of the highest node of TGA is lower.

The price we pay for this alternative is that in the middle of the lower face of T TGA has a high node and edge density. In fact, almost half the number of nodes and edges of SGA is mapped there. However, because we use exact arithmetic this is no problem.

Alternative mapping of TGA to P PGA is very inhomogeneous. The inhomogeneity can be strongly reduced by using another mapping of TGA to P. Unfortunately, due to limited space, we can not explain this mapping in detail. Compared with the previous mapping, those parts of PGA that are within the lower face of T remain unchanged, and the parts that were first mapped just outside the lower face of T are now stretched in a direction outward from the center of the lower face of T. The stretching is done over a distance of the order of the edge length of T. See figure 4. We implemented this mapping as a procedure that runs through all nodes of TGA and maps them one at a time on P. So, the time complexity of this mapping is proportional to the number of nodes of TGA, just like the mapping we discussed earlier. Also for this mapping, TGA should not have a node at the top vertex of T.

5 An example application

Until now we have been discussing how to map SGA on the plane. The main goal of our method is however not to map a single graph on the plane, but to

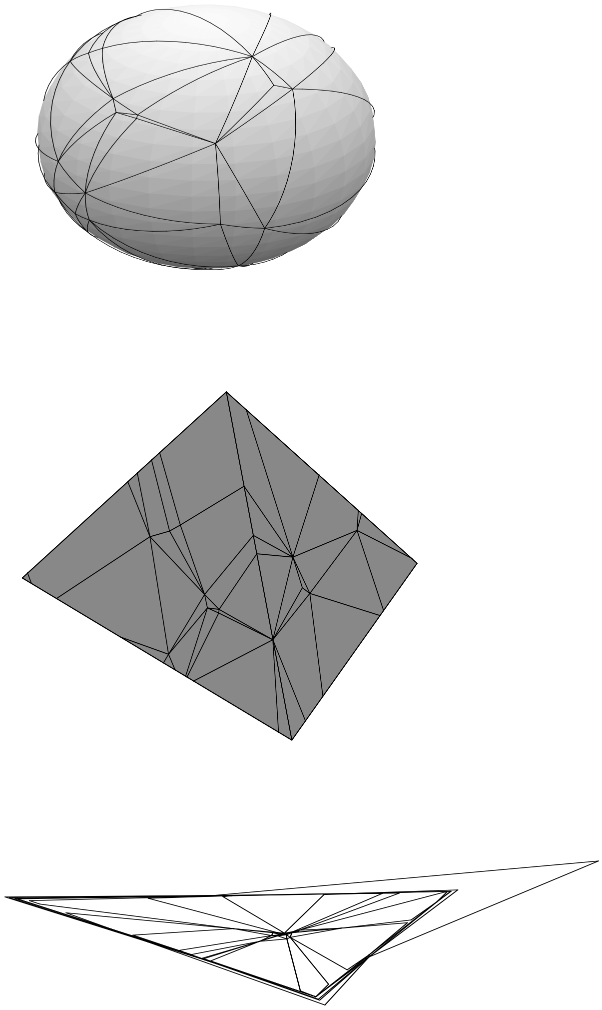

Fig. 3. The same initial graph SGA as in figure 2, mapped on the tetrahedron T in an alternative way. T is positioned so that the origin is slightly above the lower face. It can be seen that, compared with figure 2, the density of nodes and edges on the non-horizontal faces of T has decreased, and that the z coordinate of the highest node of TGA is lower. Also it can be seen that in PGA there is a high node and edge density in the middle.

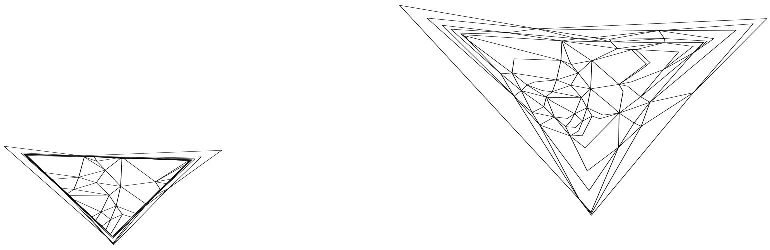

Fig. 4. Left: PGA resulting from the ordinary mapping. Right: PGA resulting from the alternative mapping, giving a more homogeneous graph. In the center of these two figures the graphs are the same.

map two graphs on the plane, to compute for example their overlay, and map the result back on the sphere. As an example we will do that. See figure 5.

We start with two graphs SGA and SGB on the sphere. These graphs are mapped on the tetrahedron, giving the graphs TGA and TGB. These graphs are mapped on the plane, giving PGA and PGB. The overlay of PGA and PGB is calculated with an algorithm implemented in LEDA, giving the graph PGAB. Then PGAB is mapped back on T giving TGAB, and TGAB is mapped back on the sphere giving SGAB. Finally, the marked nodes in SGAB that were created when mapping SGA and SGB to T are deleted when they do not coincide with other nodes. The whole process is shown in figure 4. When working with two graphs, both TGA and TGB have to be considered when it is checked whether a node is mapped on the top vertex of T, and when it is determined which node has the greatest z coordinate.

Literature

[1] D. H. Maling, Coordinate Systems and Map Projections. George Philop and Sons Limited, London, 1973.
[2] H. Bekker, J. B. T. M. Roerdink: An Efficient Algorithm to Calculate the Minkowski Sum of Convex 3D Polyhedra. Proc. of the Int. Conf. on Computational Science, San Francisco, CA, USA,2001
[3] A. V. Tuzikov, J. B. T. M. Roerdink, H. J. A. M. Heijmans: Similarity Measures for Convex Polyhedra Based on Minkowski Addition. Pattern Recognition 33 (2000) 979-995
[4] U. Finke, K. H. Hinrichs: Overlaying simply connected planar subdivisions in linear time. Proc. of the 11th Int. symposium on computational geometry, 1995.
[5] M. V. A. Andrade, J. Stolfi, Exact Algorithms for Circles on the Sphere. International Journal of Computational Geometry and Applications, Vol. 11, No. 3 (2001) 267-290.
[6] K. Melhorn, S. Näher: LEDA A Platform for Combinatorial and Geometric Computing. Cambridge University press,Cambridge. 1999

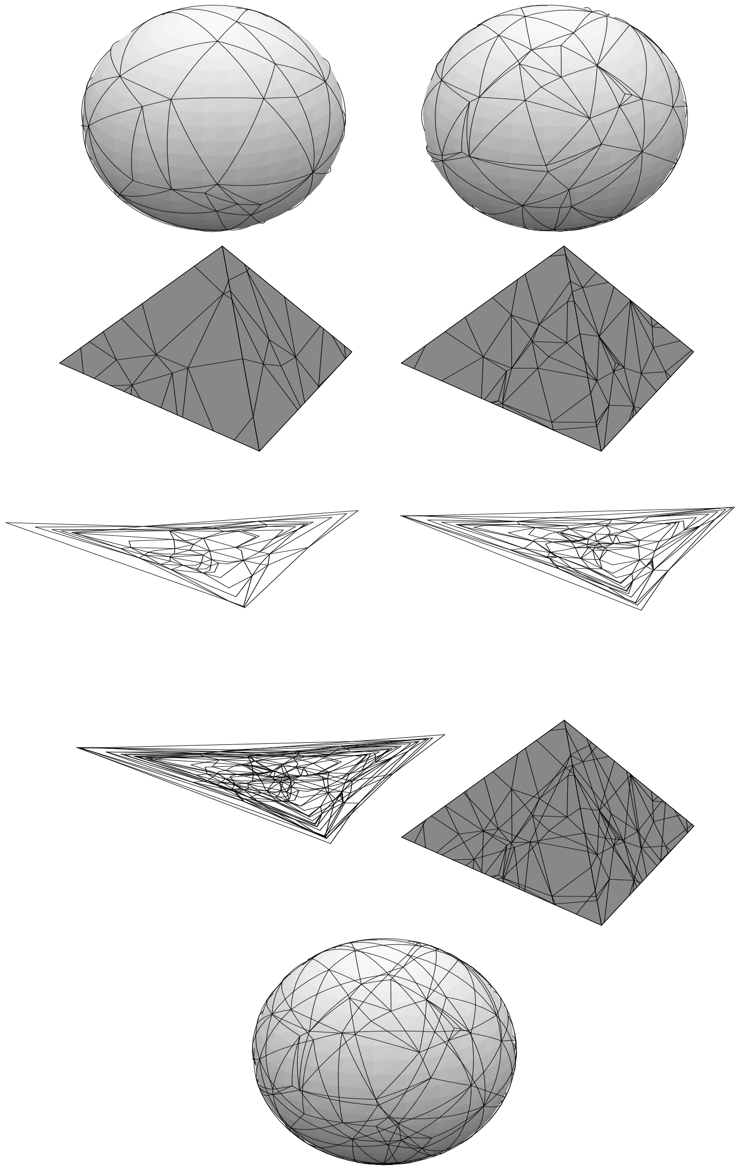

Fig. 5. The process of overlaying two graphs SGA and SGB on the sphere by mapping these graphs to the plane, calculating the overlay in the plane and mapping the result back to the sphere. Top row: SGA and SGB. Second row: TGA and TGB. Third row: PGA and PGB. The alternative mapping of TGA and TGB to P has been used, so PGA and PGB are not very inhomogeneous. Fourth row: the overlay of PGA and PGB, called PGAB, and PGAB mapped back to T, called TGAB. Fifth row: TGAB mapped to the sphere giving the overlay of SGA and SGB. In this figure the center of T coincides with the center of the sphere.