Open Geometry
A Graphics Programming Library Based on Open GL

Georg Glaeser

Chair for Geometry, University of Applied Arts in Vienna
O. Kokoschka-Platz 2, A-1010 Wien
email: georg.glaeser@hermes.hsak.ac.at

Hellmuth Stachel

Institute for Geometry, Vienna University of Technology
Wiedner Hauptstr. 8-10/113, A-1040 Wien
email: stachel@geometrie.tuwien.ac.at

1. What is Open Geometry?

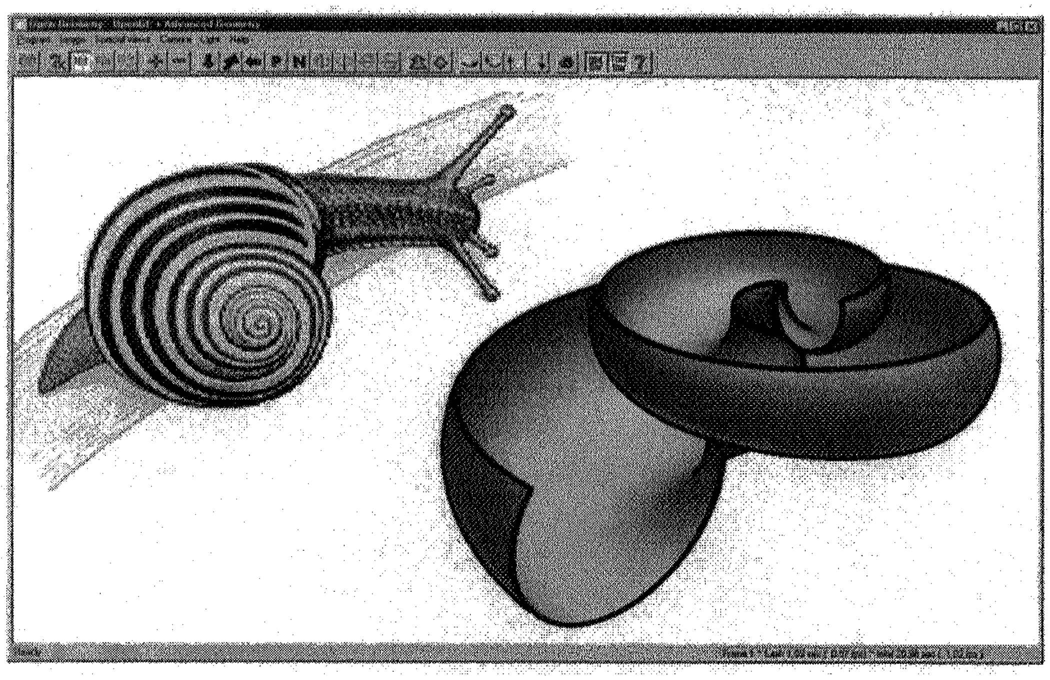

Figure 1: A typical *Open Geometry*-window on the screen. It shows a spiral surface swept by an arc of a circle and – painted as background texture – an image of a snail. The scene can be manipulated interactively. E.g., light direction and projection can be altered. In the Windows NT environment, the image can be printed with printer resolution.

Open Geometry [1] is a Graphics programming environment based on OPENGL[1]) and geometry-libraries that were developed by the authors in [2], [3] and [4]. The programming language is C++. The programs will run under various operating platforms[2]). The libraries enable the reader to realize direct geometrical thinking without having to care much about implementations.

The enclosed modules provide the reader with solutions for the most common intersection problems and measuring tasks of both planar and spatial geometry, the creation of arbitrary geometric objects (e.g., by means of different kinds of "sweeping"), and the creation of the most general solids by means of Boolean operations (intersection, union and complements of solid polyhedra).

The bookware documents the versatile and robust geometry library (which the reader can use and expand easily) and brings about a deeper insight into object oriented thinking and programming. Additionally, some relevant background information is given. E.g., the theory of Boolean intersections of solid polyhedra is explained thoroughly, covering all relevant problems in detail.

The authors emphasize the need of robust and efficient code. The results of programming are compatible to various advanced CAD systems like 3D STUDIO MAX or AUTOCAD.

2. A Simple Example

```
#include "opengeom.h" // This is the magic include file
#include "defaults3d.h" // Default camera and light
Sphere Ball;
Box SectionPlane;
RegPrism Cylinder;
void Scene::Init( )
{
   SectionPlane.Def( Cyan, 10, 10, 0.2 );
   SectionPlane.Rotate( Xaxis, −40 );
   SectionPlane.Translate( −5, −5, 8 );
   Ball.Def( LightOrange, P3d( 1, −2, 4 ), 3 );
      // center ( 1, −2, 4 ), radius 3
   Cylinder.Def( Green, 2, 9, 60, SOLID );
      // radius 2, height 9, axis = z−axis
}
void Scene::Draw( )
{
   SectionPlane.Shade( );
   Ball.Shade( );
   Cylinder.Shade( );
}
```

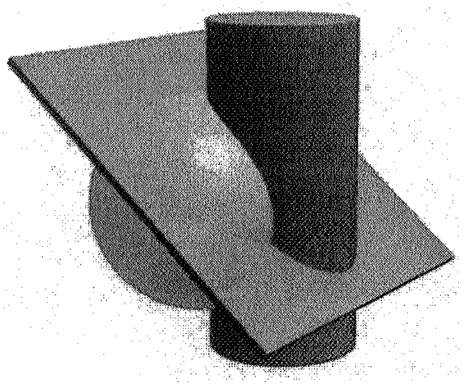

Figure 2: A simple 3D-scene, including the source code.

The bookware comes with dozens of sample programs. The following very simple program draws some geometric primitives (Figure 2) like a sphere, a cylinder and a "plane" (which is actually a thin box). The code needs only short explanation: The header-files allow to use the libraries. The program splits into an initialization part and a drawing part. (A third part can be an animation part.)

[1]) OPENGL ("Open Graphics Library") is a software interface to graphics hardware. The library contains more than one hundred distinct routines that allow to produce interactive graphics applications.

[2]) Among them Windows 95, Windows NT, Alpha Stations (Digital workstations) and Silicon Graphics workstations.

3. Advantages of Open Geometry

In the above example, hidden-surface removal is done by means of "z-buffering" which comes with OPENGL. The primitives, however, are not really united, as one might expect. The algorithms of *Open Geometry* now allow to apply Boolean operations like intersection, union, difference. The result are new objects (Figure 3).

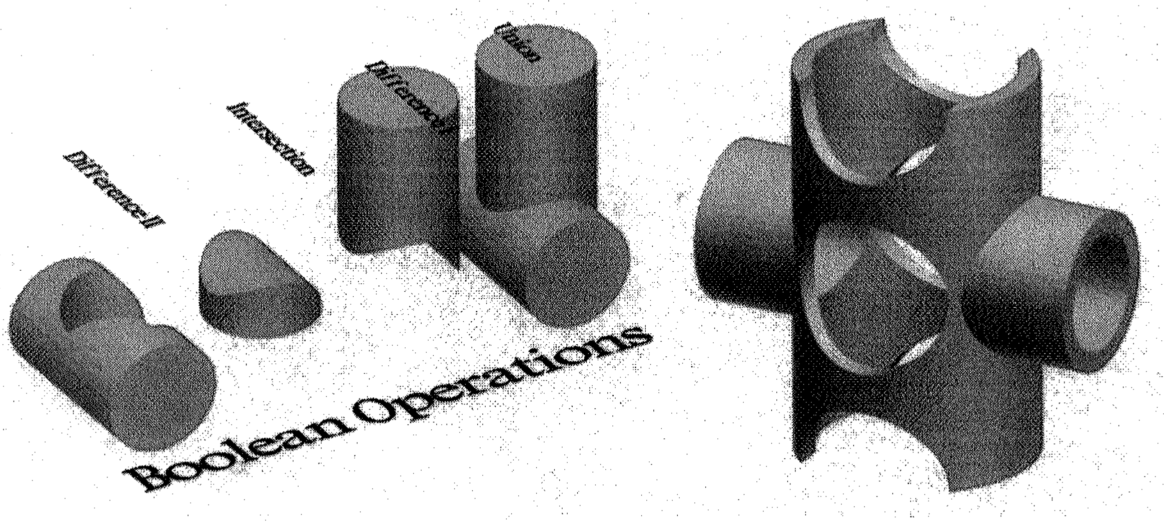

Figure 3: This image was also created by *Open Geometry*.

Figure 4 (left) illustrates how objects from other data-bases (the aircraft was created by 3D STUDIO MAX) can be imported by *Open Geometry* (the background texture is a scanned photograph). Textures can also be attached to 3D-polygons (Figure 4, right).

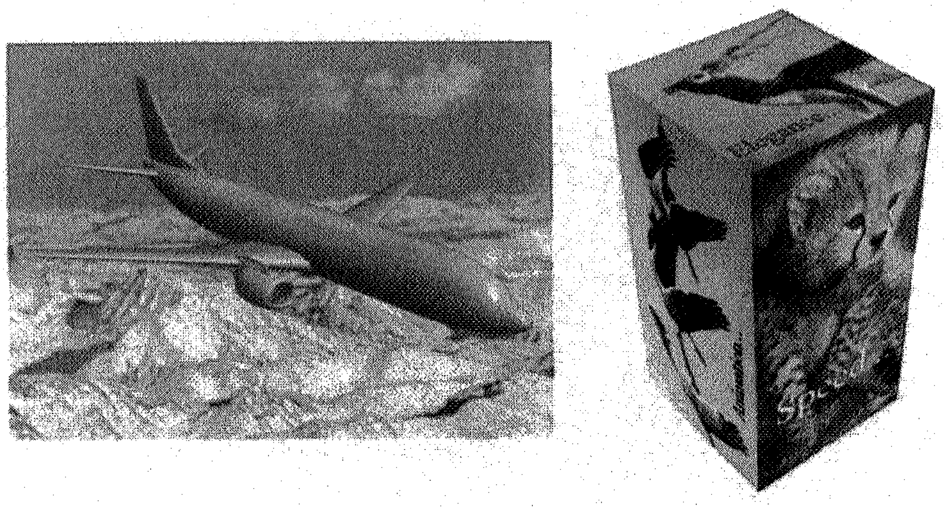

Figure 4: Left: Importing data from other programs. Right:Attaching textures to polygons.

The unique advantage of *Open Geometry*, however, is that it is not limited in any direction: Since the programmer can extend the libraries, even complicated geometric processes can be visualized (Figure 5).

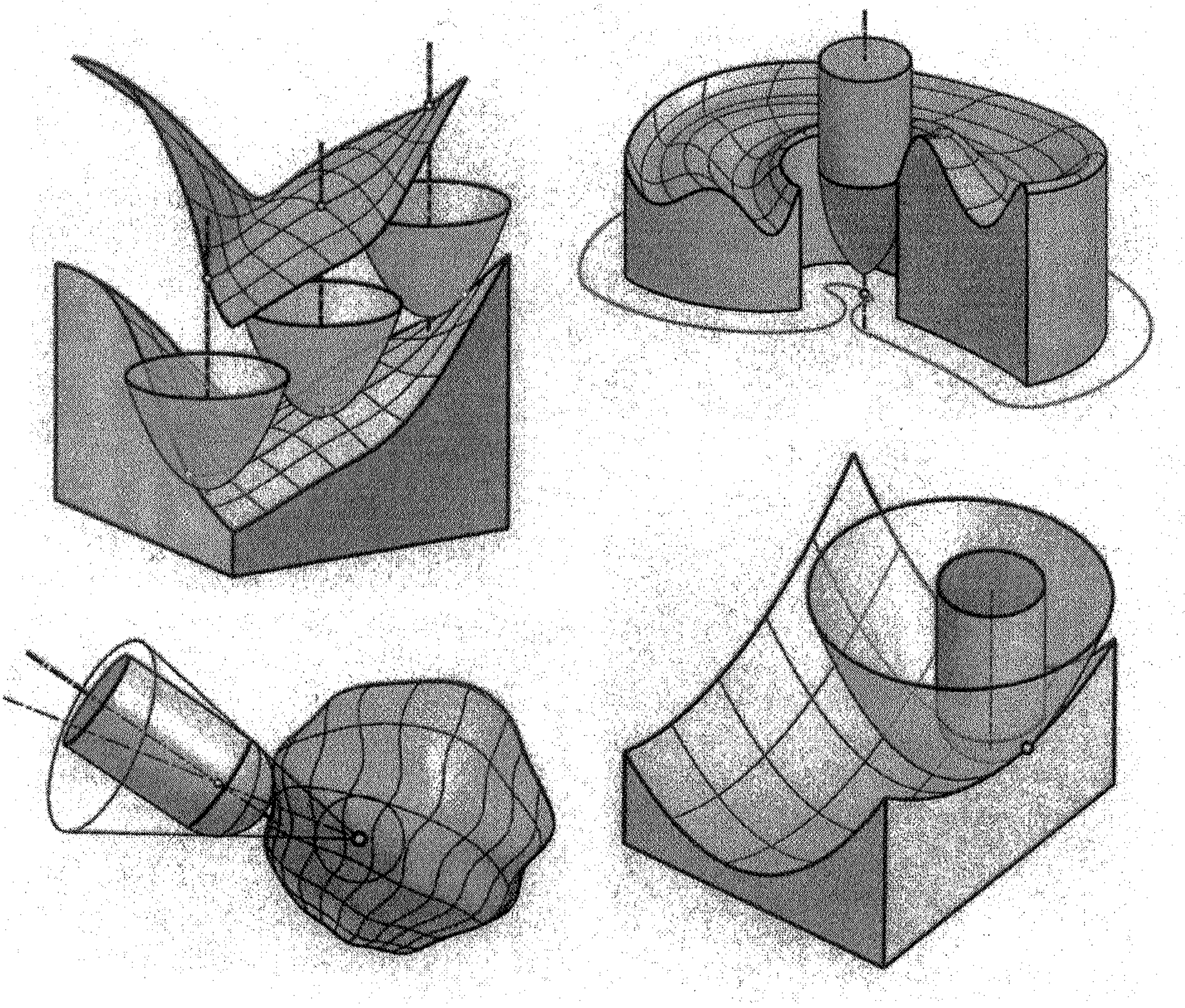

Figure 5: Simulation of milling processes [5] with *Open Geometry*.

References

[1] G. Glaeser, H. Stachel: *Open Geometry.* Springer-Verlag, New York 1998. See also http://www.hsak.ac.at/opengeom.

[2] G. Glaeser: *Objekt-orientiertes Graphik-Programmieren mit der Pascal-Unit Supergraph.* Teubner, Stuttgart 1992.

[3] G. Glaeser: *Fast Algorithms for 3D-Graphics.* Springer-Verlag, New York 1994.

[4] H. Stachel: *CAD3D.* Educational CAD-Software, Wien 1993.

[5] H. Pottmann, J. Wallner, G.Glaeser, B. Ravani: *Geometric criteria for gouge-free three-axis milling of sculptured surfaces.* Proc. of DETC'98, Atlanta.