

PostScript-Ausgabe von OpenGL

Arne Dür

Institut für Mathematik

Februar 2003

Aufgabenstellung:

Gegeben: 3D-Objekt, generiert mit Hilfe von OpenGL

Gesucht: Programm, das ein Bild des Objekts als PostScript-Vektorgrafik in einer EPS-Datei abspeichert

Problem: *Sichtbarkeit von Objekten* muss korrekt wiedergegeben werden, obwohl der pixelweise Vergleich mit Hilfe des Tiefenbuffers nicht zur Verfügung steht

Beispiel:

- Teekanne oder Quadriken in Geneps
- römische Fläche in Surflab
- Grafik in MatLab, Maple, usw.

OpenGL:

- Open Graphics Library
- Software–Schnittstelle für Grafik–Hardware
- plattform–unabhängiger Industrie-Standard (verfügbar für Windows, Linux, Apple, ...)
- Konkurrent: Direct3D (Microsoft)

PostScript:

- Seitenbeschreibungssprache der Firma Adobe für Text und Grafik
- plattform- und geräte–unabhängig, jeder Previewer oder Drucker kann optimal rastern
- Encapsulated-PostScript-Format (EPS) erlaubt einfaches Einbinden von Grafik in Dokumente

Rastergrafik:

- 2D-Geometrie mit *ganzzahligen Koordinaten* (Raster), daher Qualitätsverlust bei Skalierung
- Objekte werden durch Pixel beschrieben
- Bildschirm mit 21 Zoll und 100 dpi:

$$1600 \times 1200 \approx 2 \text{ MPixel}$$

d.h. bei 32 Bit Farbtiefe ca. 8 MByte

- Drucker mit A3+ und 1200 dpi:

$$15600 \times 22800 \approx 356 \text{ MPixel}$$

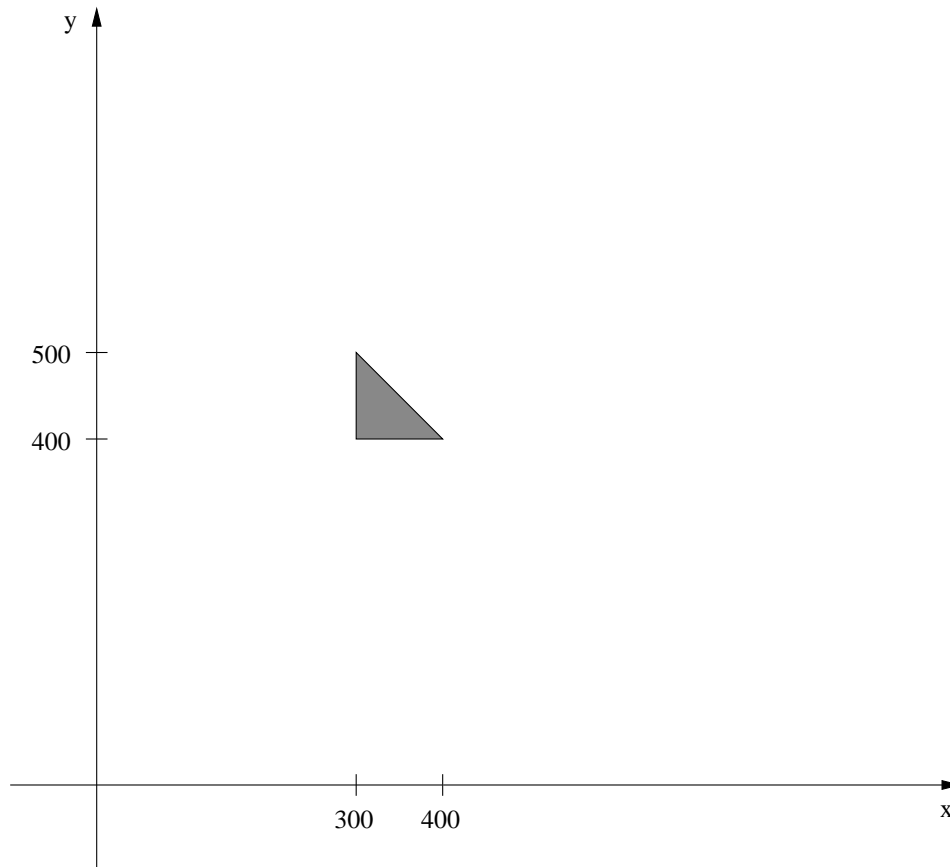
d.h. bei 32 Bit Farbtiefe ca. 2.8 GByte

- Vergleich Bildschirm/Drucker zeigt, dass Screen-Dumps zu grob auflösen

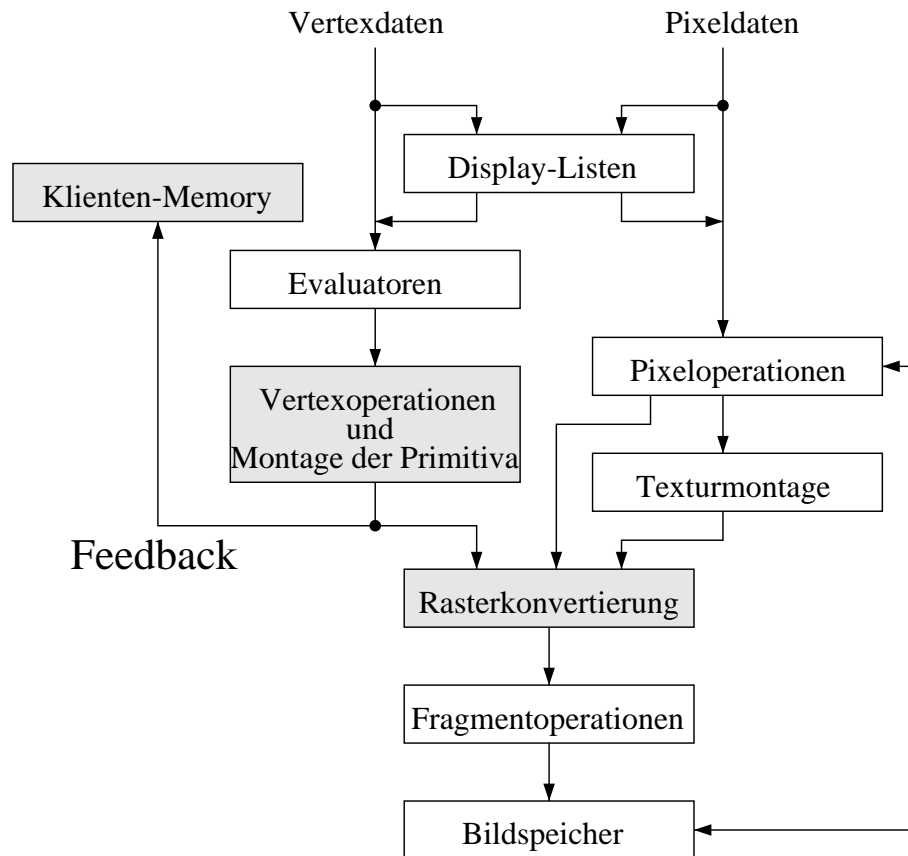
Vektorgrafik:

- 2D-Geometrie mit *reellen Koordinaten* (wie in Vektorrechnung), daher Skalierung ohne Qualitätsverlust möglich
- Objekte werden durch Typ und geometrische Daten spezifiziert, z.B. ein graues Dreieck in PostScript:

```
0.5 setgray
newpath
300 400 moveto
100 0 rlineto
-100 100 rlineto
closepath
fill
```



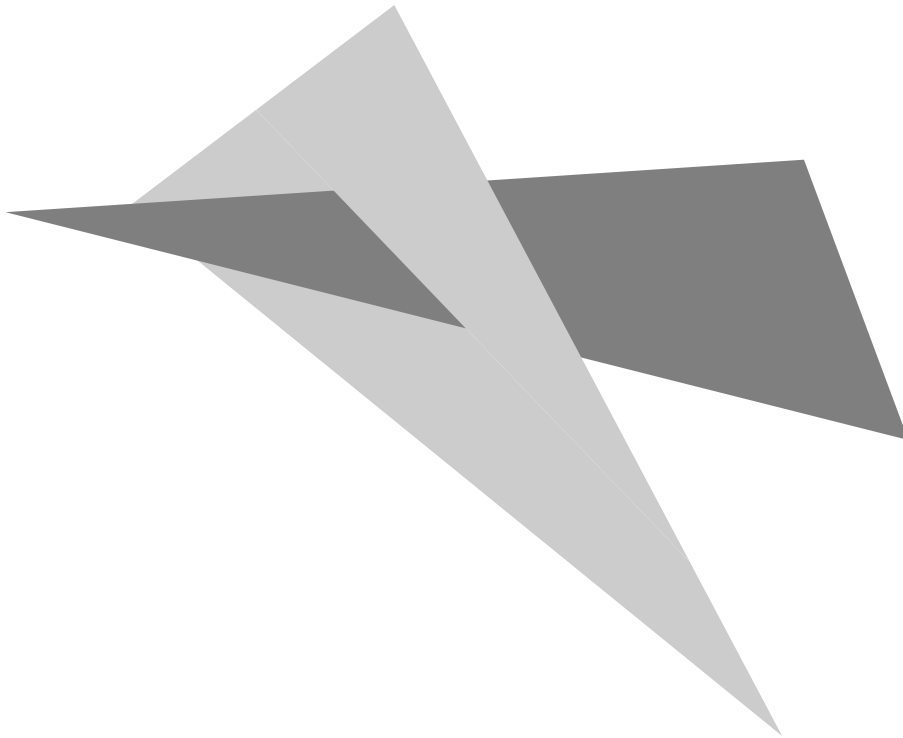
OpenGL-Rendering-Pipeline:



- Die geometrischen Primitiva von OpenGL sind
 - Polygone ,
 - Streckenzüge und
 - Punkte .
- Im Feedback-Modus können die fertig montierten Primitiva vor der Rasterung zurückgeholt werden, wobei auf Texturierung oder Sortierung mit dem Tiefenbuffer verzichtet werden muss.
- Idee und erste Implementierung von Mark Kilgard

Maler-Algorithmus:

- Die im Feedback-Modus erhaltenen Primitiva von OpenGL können automatisch nach PostScript übertragen werden.
- Da in PostScript neue Figuren alte übermalen, werden die Primitiva von hinten nach vorne sortiert.
- Um eine korrekte Sortierung zu ermöglichen, müssen die Primitiva im Allgemeinen zerlegt werden. Dazu werden *Verfahren der Geometrie* eingesetzt.



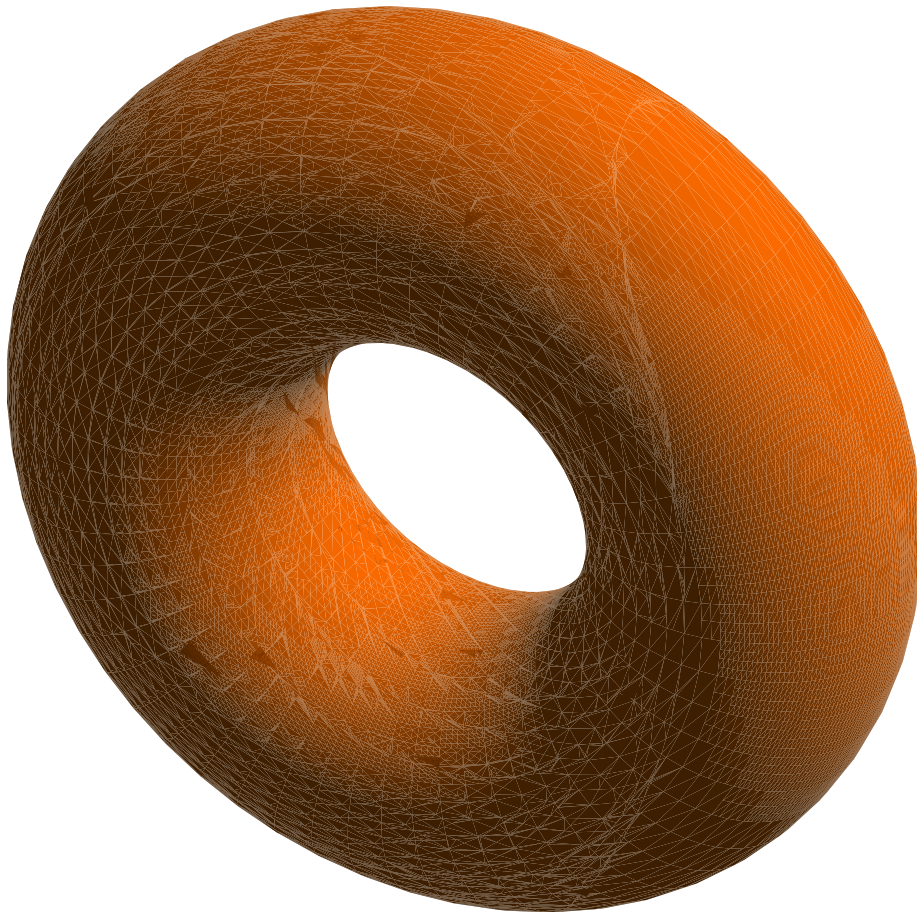
Binary Space Partitioning (BSP)

(Fuchs, SIGGRAPH, 1980; freie Implementierung von Geuzaine bei <http://www.geuz.org/gl2ps>)

Das erste Polygon zerlegt den Raum in zwei Halbräume, die restlichen Polygone werden mit den Halbräumen geschnitten. Dies wird für jeden Halbraum wiederholt, bis alle Polygone verbraucht sind.

- + korrekte Sortierung durch *Partitionsbaum* gegeben
- hohe Komplexität (viele neue Polygone)

Die folgende EPS-Grafik hat 983383 Bytes.



Polygon Intersection Topological Sorting (PITS)

(Dür,Leimgruber,Elsner,Miterrutzner,Maran 2003)

Polygone werden nur bei Durchdringungen zerlegt, die zerlegte Polygone werden bezüglich der Überlappungen verglichen.

- + geringere Komplexität (oft nur 50% EPS)
- korrekte Sortierung nicht immer garantiert (wegen möglicher *Zyklizität* der Quasiordnung, aber praktische Erfahrungen mit Szenen der Mathematik positiv)

Die folgende EPS-Grafik hat 498329 Bytes.

