

# Versteht mich mein Drucker? — Die Universaldruckersprache PostScript

C. Gollwitzer

<sup>1</sup>Experimentalphysik V, Universität Bayreuth, D-95440 Bayreuth

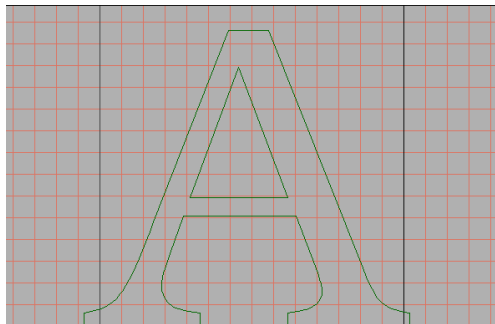
Donnerstagsbesprechung 24.11.2005

# Wie druckt der Pixeldrucker Buchstaben?

- ▶ Druckbilder werden heute aus Pixeln aufgebaut (Matrixdrucker, Laserdrucker, Tintenstrahldrucker, Nadeldrucker)
- ▶ Früher: Schreibmaschine, Typenraddrucker, Bleisatz
- ▶ Dokumente enthalten Text und Grafik (Liniengrafik), selten Pixel (Photographien)

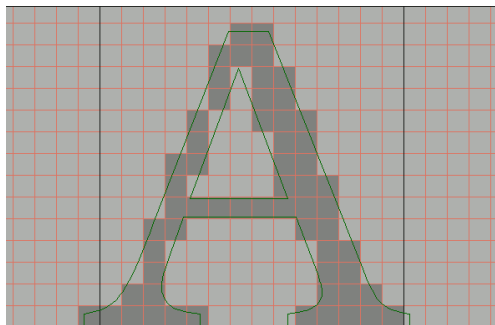
# Wie druckt der Pixeldrucker Buchstaben?

- ▶ Druckbilder werden heute aus Pixeln aufgebaut (Matrixdrucker, Laserdrucker, Tintenstrahldrucker, Nadeldrucker)
- ▶ Früher: Schreibmaschine, Typenraddrucker, Bleisatz
- ▶ Dokumente enthalten Text und Grafik (Liniengrafik), selten Pixel (Photographien)



# Wie druckt der Pixeldrucker Buchstaben?

- ▶ Druckbilder werden heute aus Pixeln aufgebaut (Matrixdrucker, Laserdrucker, Tintenstrahldrucker, Nadeldrucker)
- ▶ Früher: Schreibmaschine, Typenraddrucker, Bleisatz
- ▶ Dokumente enthalten Text und Grafik (Liniengrafik), selten Pixel (Photographien)



# Wer rastert?

## Rasterizer ist der Computer

- ▶ GDI-Drucker, ältere Nadeldrucker, viele Tintenstrahldrucker
- ▶ Computer muss die genaue Auflösung des Druckers kennen
- ▶ Treiber speziell für den Drucker erforderlich
- ▶ Druckdaten sind nur einmal verwendbar
- ▶ Verbesserungen des Rasterizers durch Update des Treibers
- ▶ Rechenleistung des Computers blockiert, heute meist schneller

## Rasterizer im Drucker

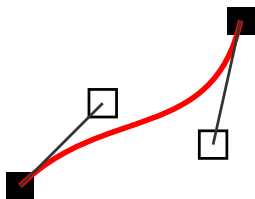
- ▶ PCL, PostScript, die meisten Laserdrucker
- ▶ Auflösung des Druckers egal
- ▶ Universaltreiber kann viele Drucker bedienen
- ▶ Druckdaten können gespeichert, angesehen und auf anderen Geräten in optimaler Qualität ausgegeben werden
- ▶ Update des Rasterizers kompliziert (Firmware update)
- ▶ Fire & forget, der Drucker rechnet

# PostScript als Programmiersprache

- ▶ Stackautomat  $\Rightarrow$  UPN:  
4 + 5  $\Rightarrow$  4 5 add =
- ▶ Aktion für ein Objekt entweder...
- ▶ ...wörtlich nehmen, d. h. auf den Stack legen (literal object): Integer 2, Real 2.3, (String), /Name, { prozedur }
- ▶ ...oder ausführen (executable object): eingebauter Operator (pop, add, mul...), Name
- ▶ Definieren einer Subroutine über Namen und *Dictionaries*:  
/plusneun { 9 add } def  
25 plusneun plusneun =
- ▶ def fügt plusneun dem aktuellen Dictionary hinzu
- ▶ plusneun wird bei Ausführung dort nachgeschlagen
- ▶ Vollwertige Programmiersprache: Demonstration durch PShttpd, Fraktalberechnung

# Grafikmodell – Pfade

- ▶ Linienzüge auf dem Papier, bestehend aus Geradenstücken und kubischen Bezier-Splines



- ▶ Pfade können mit einem Pinsel nachgefahren (stroke) oder gefüllt (fill) werden
- ▶ Pfadkonstruktion zeichnet zunächst *nichts*:  
100 100 moveto  
200 200 lineto

# Grafikmodell – Zeichenoperator

- ▶ Erst der Zeichenoperator zeichnet den aktuellen Pfad auf das Papier:

```
stroke
```

- ▶ Der Zeichenoperator benutzt den aktuellen Zustand des Pinsels, z.B. Linienbreite und Farbe

```
10 10 moveto 100 100 lineto 150 100 lineto  
1 0 0 setrgbcolor fill
```



# Grafikmodell – Text

- ▶ Laden eines Fonts:  
/Times-Roman 70 selectfont
- ▶ Text direkt zeichnen 50 200 moveto  
1 0 0 setrgbcolor  
(EP5) show
- ▶ Text an den Pfad anhängen  
( Uni Bayreuth) false charpath  
0 1 0 setrgbcolor stroke
- ▶ 35 Standard-Postscript-Fonts:  
Palatino-Roman Helvetica Times-Roman Courier  
NewCenturySchlbk-Roman AvantGarde-Book  
Bookman-Demi ZapfChancery-MediumItalic

# Grafikmodell – Transfermatrix (CTM)

- ▶ Koordinaten in Postscript  $\neq$  Pixel!
- ▶ 1 unit = 1 typografischer Punkt (1/72 Inch) bei Programmstart
- ▶ Umrechnung von PS-Koordinaten in (Hardware)-Pixel:  
CTM  
$$\vec{x}^{HW} = C\vec{x}^{PS} + \vec{d}$$
- ▶ Effekte wie Drehung, Scherung, Skalierung durch Veränderung der CTM möglich: concatmatrix oder über Hilfskürzel (rotate, scale, translate)

# Grafikmodell – Clipping path

- ▶ Eingrenzen des Druckbereichs durch jeden beliebigen Pfad möglich (`clip`)
- ▶ Vergrößern des Clipping path ist nicht möglich
- ▶ Clipping für einzelne Objekte (Grafiken, Schrifteffekte) durch Sichern und Wiederherstellen des Grafikzustands (`gsave`, `grestore`)
- ▶ Clipping an komplexen Pfaden (z.B. Text) ermöglicht interessante Effekte

## Verwandte Themen

- ▶ Fonts in Postscript: Type1, Type0, Type3, Type42(=TrueType), OpenType
- ▶ Document Structuring Convention (DSC) ermöglicht das Zerlegen in einzelne Seiten
- ▶ EPS ist ein einseitiges Dokument mit BoundingBox und erfüllt die DSC
- ▶ PDF ist PostScript ohne Programmiersprache, aber mit Index und Hypertext

# Literatur

1. Postscript Language Reference, 3rd ed., Reading, 1999  
<http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>
2. PShttpd – a webserver in Postscript  
<http://public.planetmirror.com/pub/pshttpd/>
3. Thinking in Postscript, Reading, 1990  
<http://amath.colorado.edu/documentation/postscript/ThinkingInPostScript.pdf>
4. Fraktale mit Postscript:  
<http://www.ankiewicz.com/design/fractals.html>
5. Iterated Function Systems in PostScript  
<http://www.marypat.org/ifs.html>