Medien-
Technik

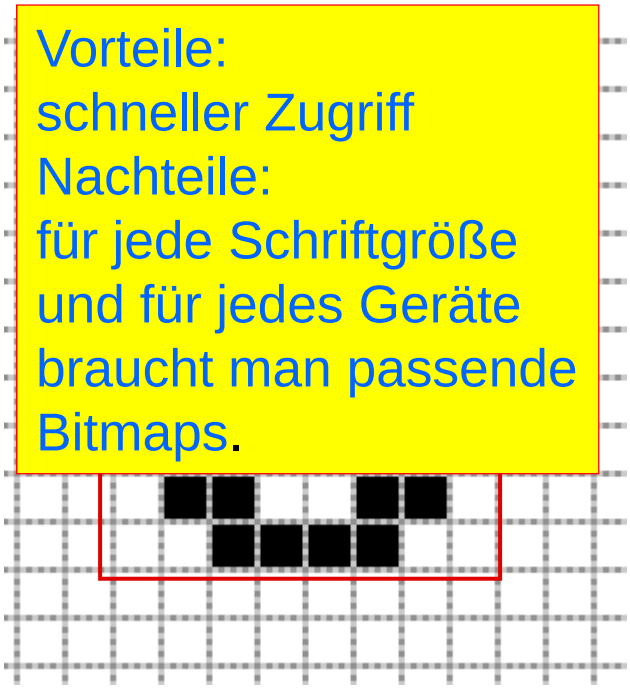
Digitale Schriften

Rasterschriften

Umriss-Schriften

Rasterschrift:
jede Glyphe wird
als Bitmap abgelegt.

Vorteile:
schneller Zugriff
Nachteile:
für jede Schriftgröße
und für jedes Geräte
braucht man passende
Bitmaps.



Bitmap

00111100
01100110
01100110
01100110
01100110
01100110
01100110
01100110
00111100

$= 0*128+0*64+1*32+1*16$
$+1*8+1*4+0*2+0*1$
$= 60 = 0x3c$

3c 66 66 66 66 66 66 66 3c

9*8 **Pixel** (picture element)

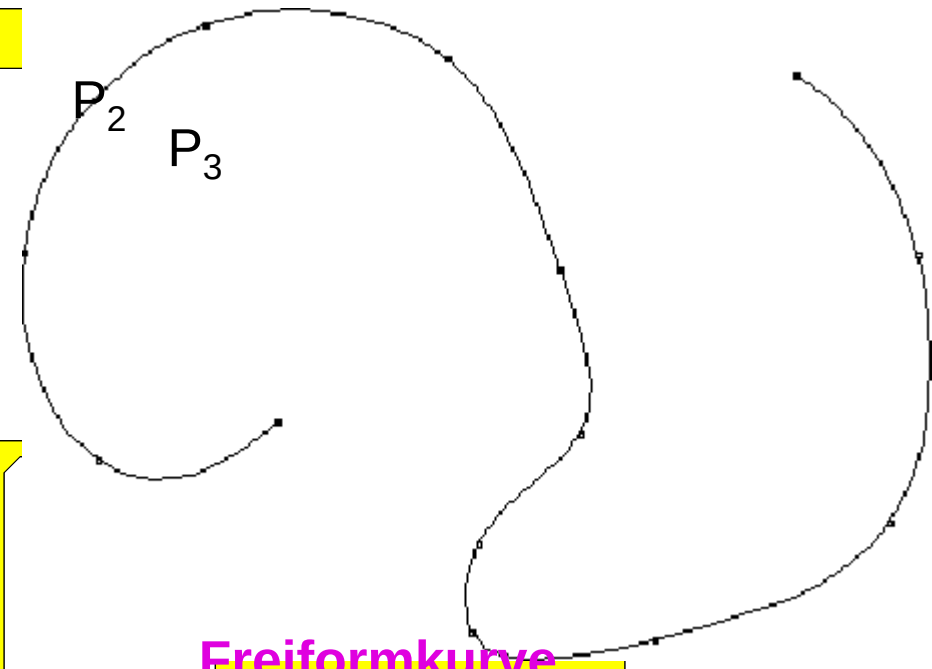
Medien-
Technik

Digitale Schriften
Speicherbedarf

	Bildschirm	Laserdrucker	Belichter
Auflösung	1/72 Zoll	1/600 Zoll	1/2500 Zoll
10 pt	10 Pixel	83 Pixel	347 Pixel
Speicherbedarf Bitmap	10-20 Byte	1 kB	30 kB
Speicher Outline	100-200 B	100-200 B	100-200 B

Medien-
Technik

Digitale Schriften
Outlines



Polygone I

II

Qui est-ce ?

Pierre Bezier

$(P_0, P_1, ..., P_{15})$

Polygone:

Vorteil:
einfach zu speichern
und zu zeichnen.
Bei Skalierung nicht
mehr glatt.

Freiformkurve
Wie kann man sie darstellen?

Praxis:

Wie wenig Punkte

braucht man ?

Medien-
Technik

Digitale Schriften
Bezier-Kurven

Parametergleichung
für diese Strecke:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + t \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} \text{ mit } 0 \le t \le 1$$

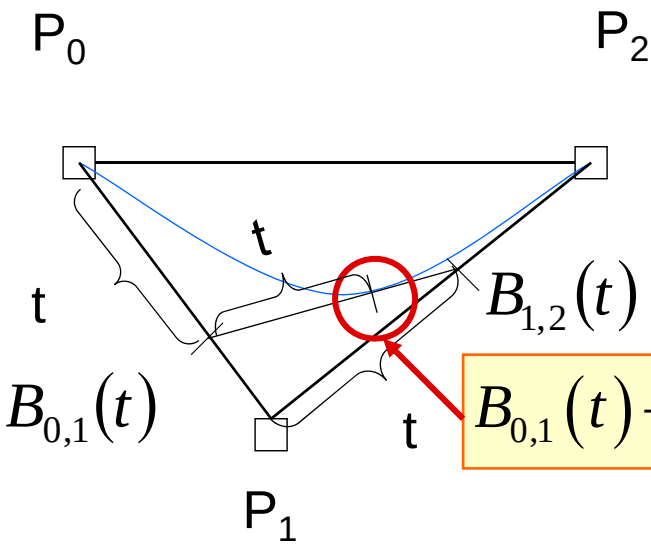
$$B_{0,1}(t) = P_0 + t(P_1 - P_0) \text{ mit } 0 \le t \le 1$$

Medien-
Technik

Digitale Schriften
Bezier-Kurven

$$B_{0,1}(t) = P_0 + t(P_1 - P_0) \text{ mit } 0 \le t \le 1$$

$$B_{1,2}(t) = P_1 + t(P_2 - P_1) \text{ mit } 0 \le t \le 1$$



$$B_{0,1}(t) + t\left(B_{1,2}(t) - B_{0,1}(t)\right) \text{ mit } 0 \le t \le 1$$

Quadratische Bezierkurve definiert durch 3 Punkte

$$B_{0,2}(t) := B_{0,1}(t) + t\left(B_{1,2}(t) - B_{0,1}(t)\right) \text{ mit } 0 \le t \le 1$$

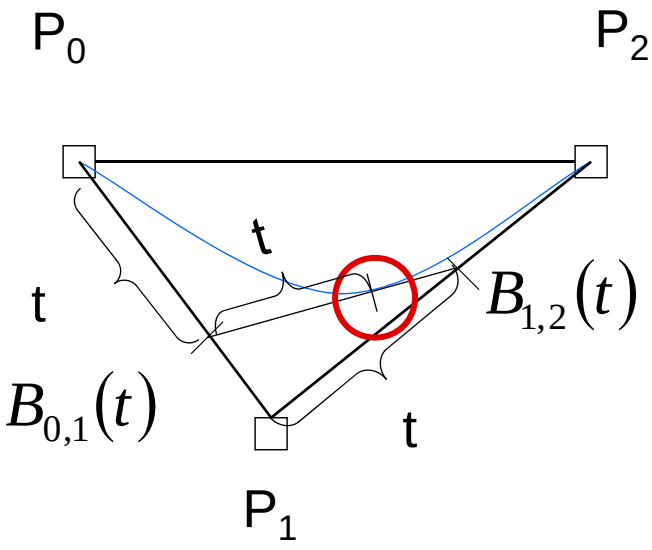
$$B_{0,2}(t) := P_0 + t(P_1 - P_0) + t\left(P_1 + t(P_2 - P_1) - P_0 - t(P_1 - P_0)\right) \text{ mit } 0 \le t \le 1$$

Medien-
Technik

Digitale Schriften
Bezier-Kurven

$$B_{0,1}(t) = P_0 + t(P_1 - P_0) \text{ mit } 0 \le t \le 1$$

$$B_{1,2}(t) = P_1 + t(P_2 - P_1) \text{ mit } 0 \le t \le 1$$



Reine Termumformung:

$$B_{0,1}(t) = (1-t)P_0 + tP_1 \text{ mit } 0 \le t \le 1$$

$$B_{1,2}(t) = (1-t)P_1 + tP_2 \text{ mit } 0 \le t \le 1$$

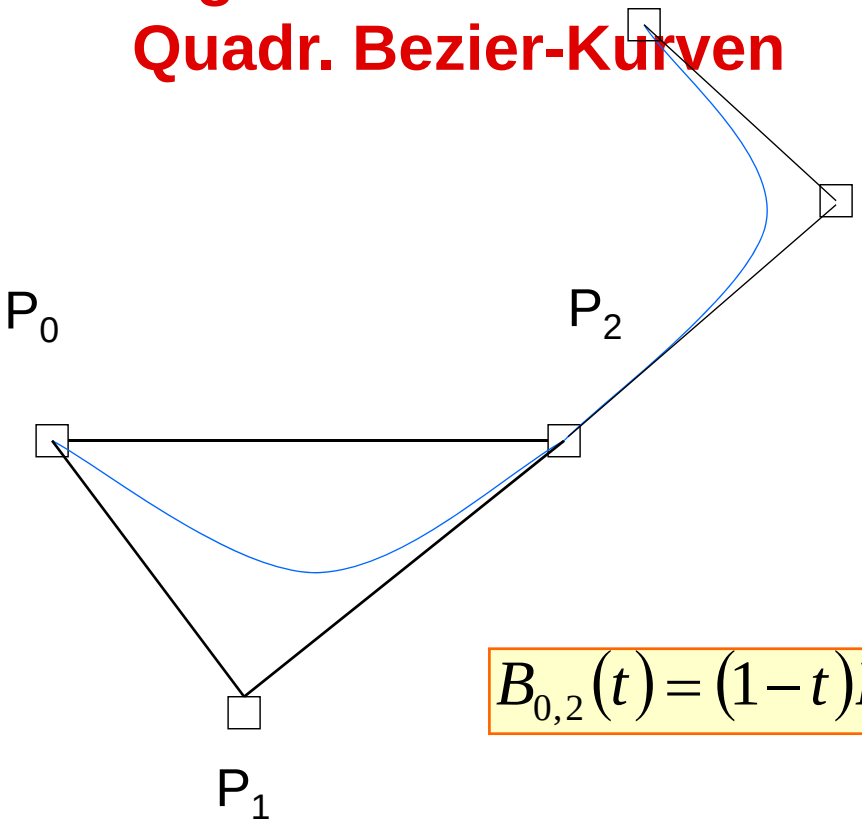
Quadratische Bezierkurve definiert durch 3 Punkte

$$B_{0,2}(t) := B_{0,1}(t) + t\left(B_{1,2}(t) - B_{0,1}(t)\right) \text{ mit } 0 \le t \le 1$$

Reine Termumformung: $B_{0,2}(t) = (1-t)B_{0,1}(t) + tB_{1,2}(t) \text{ mit } 0 \le t \le 1$

Medien-
Technik

Digitale Schriften
Quadr. Bezier-Kurven



$$B_{0,2}(t) = (1-t)B_{0,1}(t) + tB_{1,2}(t) \text{ mit } 0 \le t \le 1$$

Eigenschaften:

Quadr. Bezierkurve ist Stück einer Parabel (Polynom 2. Grades)

Die Strecke P_0P_1 ist Tangente in P_0

Die Strecke P_1P_2 ist Tangente in P_2

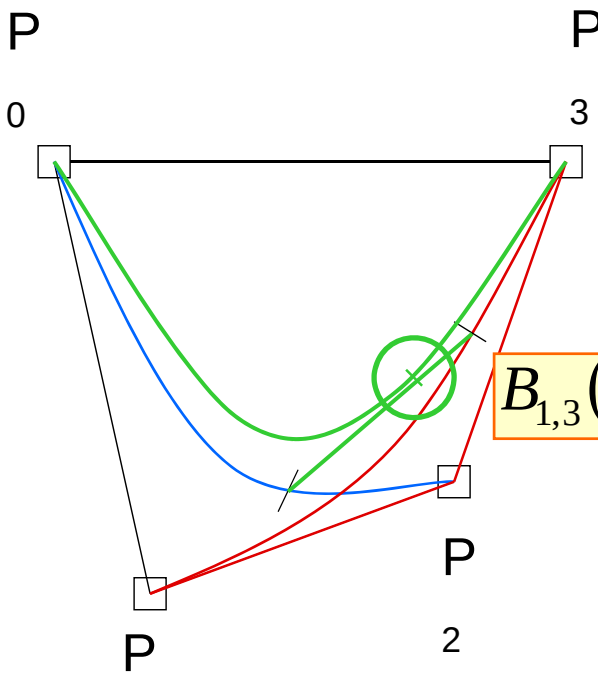
Bezierkurven lassen sich glatt aneinander fügen

Medien-
Technik

Digitale Schriften
Kubische Bezier-Kurven

Ergebnis:

$$B_{0,3}(t) = (1-t)B_{0,2}(t) + tB_{1,3}(t) \text{ mit } 0 \le t \le 1$$



Kubische Bezierkurve!

$$B_{1,3}(t) = (1-t)B_{1,2}(t) + tB_{2,3}(t) \text{ mit } 0 \le t \le 1$$

Quadr. Bezierkurve definiert durch P_1, P_2, P_3

$$B_{0,2}(t) = (1-t)B_{0,1}(t) + tB_{1,2}(t) \text{ mit } 0 \le t \le 1$$

Quadr. Bezierkurve definiert durch P_0, P_1, P_2

Demo-Applet: http://www.cg.inf.ethz.ch/Html/Lehrveranstaltungen/Vorlesungen/applets.html

Medien-
Technik

Digitale Schriften
Bezier-Kurven

Verallgemeinerung auf n+1 Stützpunkte P_0 .. P_n:

$$B_{0,n}(t) = (1-t)B_{0,n-1}(t) + tB_{1,n}(t) \text{ mit } 0 \le t \le 1$$

P_0 .. P_{n-1} P_1 .. P_n

Definition durch Rekursion

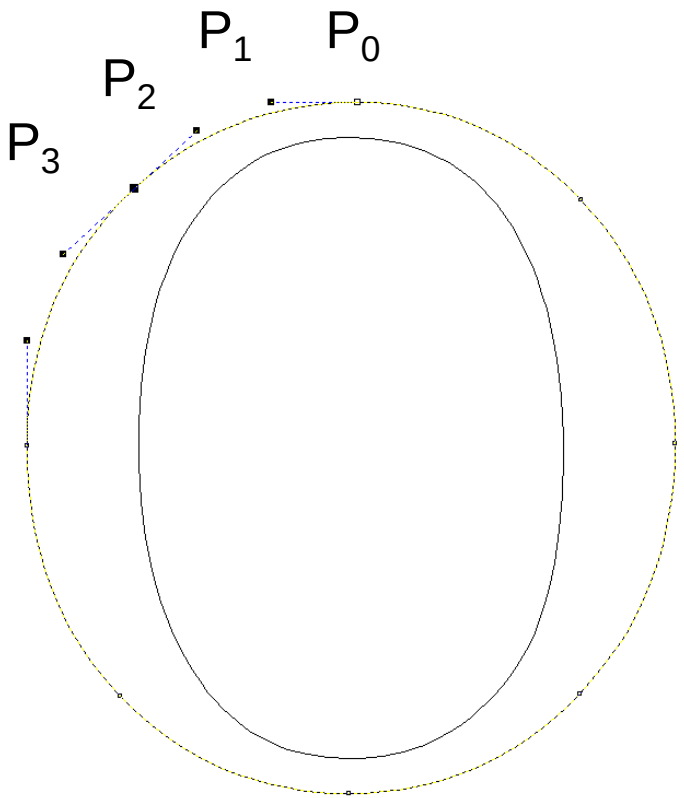
$D_n := F(D_{n-1})$

$D_1 :=$ explizit angeben

Berechnung durch Induktion

Medien-
Technik

Digitale Schriften
PostScript / TrueType



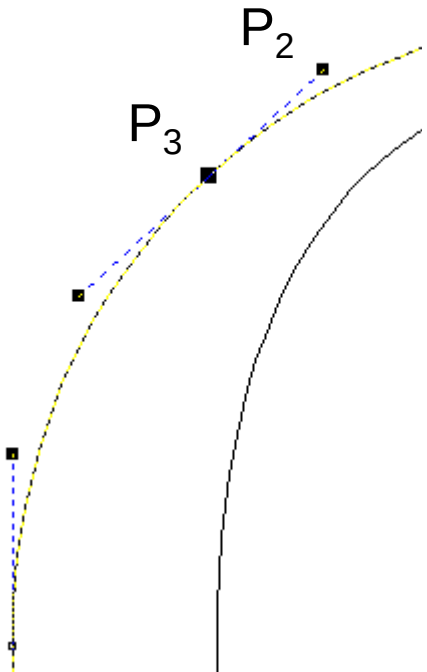
PostScript
Type 1 Fonts:

Folge von Pfadbefehlen:

P moveto
P1 P2 P3 curveto

u.a.

Corel Draw
- **Anordnen**
- **In Kurven konvertieren**
- **Kombination aufheben**

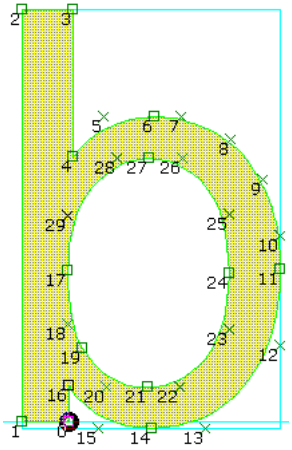


TrueType
TT-Polygone
TT-QSplines

Typographischer Standard für Outlines:
Ikarus von URW

Medien-
Technik

Truetype



Außenlinie im Uhrzeigersinn
Innenlinie gegen Uhrzeigersinn
solid to the right –Füllregel
non-zero winding fill (siehe SVG)

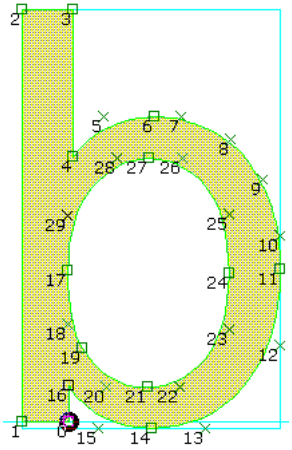
☐ ist on-öine-Punkt
× ist off-line-Punkt

Punkte 4, 5, 6 bilden quadratische Bezierkurve
zwischen 2 konsekutiven off-line-Punkten
z.B. 7 und 8 wird impliziter on-line-Punkt
mitten dazwischen angenommen

http://www.truetype-typography.com/b-outln.gif

Medien-
Technik

Truetype

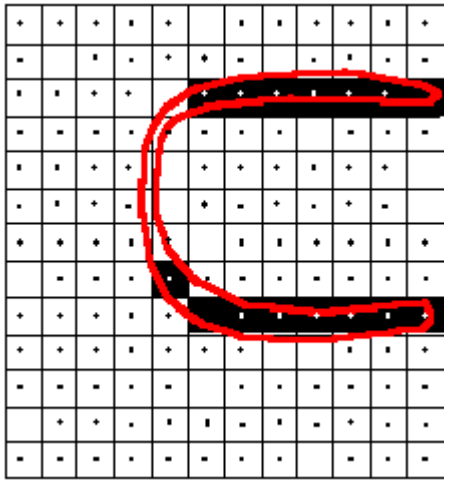


3 Arten von Glyphen

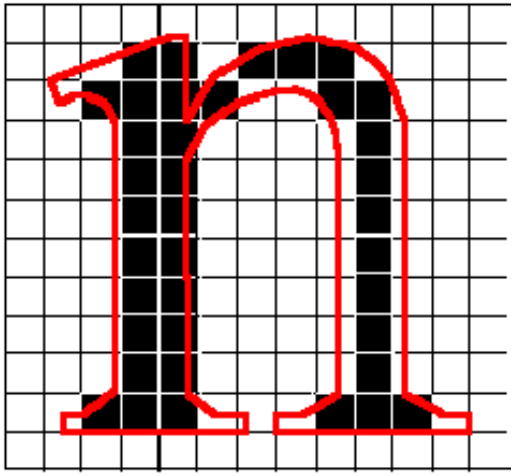
1. Einfache Glyphen (bestehn aus outline-Daten)
2. Zusammengesetzte Glyphen, referenzieren andere Glyphen
3. Glyphen ohne Kontour (z.B. white space)

Medien-
Technik

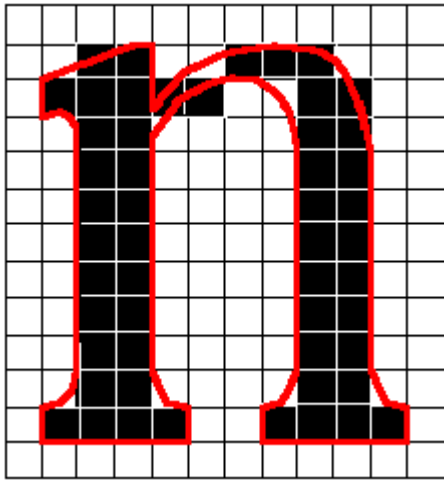
Digitale Schriften
Probleme beim Rastern



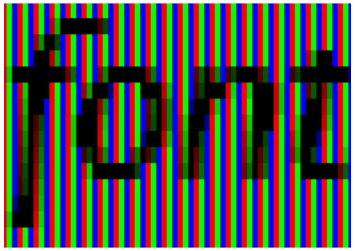
Unterbrechungen
durch Ausdünnung



Der Umriss liegt
ungünstig im Raster



Korrekturen durch
„hints“



Subpixel Rendering
Cleartype

Outline-Schrift-Datei:

- Font-Metrik-Tabelle
- Glyphen (Outlines)
- Hints
- Kerningtabelle (Ligaturen)

Medien-
Technik

Vorlesung „Medientechnik SS 2010“

Dr. Manfred Jackel
Studiengang Computervisualistik
Universität Koblenz-Landau
Campus Koblenz
Postfach 201602
56016 Koblenz

© Manfred Jackel
E-Mail: jkl@uni-koblenz.de
WWW: www.uni-koblenz.de/~jkl
mtech.uni-koblenz.de

Literatur zu diesem Kapitel

Apple TrueType Reference Manual

Microsoft Open Type Spec

Hyperlinks zu diesem Kapitel

Bezier-Applet:
http://www.cg.inf.ethz.ch/Html/Lehrveranstaltungen/Vorlesungen/applets.html

RZU Universität Zürich: PS-Fonts

B-Splines: TU Dresden

Graphische Datenverarbeitung Uni Siegen

Computergrafik Universität Osnabrück

DTP Universität Karlsruhe

DTP Lexikon

www.fontpool.com

Typografie Universität Graz

Typo-Tips

http://www.truetype-typography.com/

http://developer.apple.com/fonts/

Grafik-Quellen:

Folie 10: Microsoft Developers Network