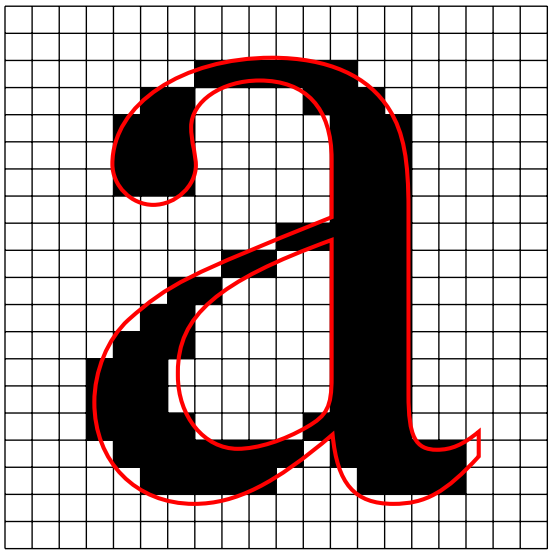
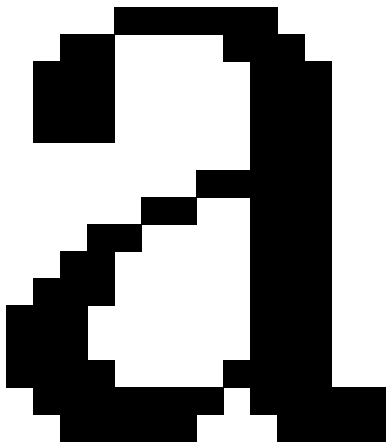
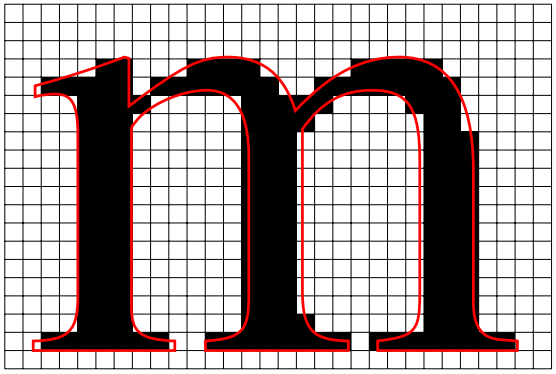
- Am Ende sind die geometrisch definierten Flächen in ein Raster gegebener Größe und Ausformung zu überführen, um sie auf dem Bildschirm oder dem Drucker darstellen zu können.
- Im einfachsten Falle handelt es sich dabei um ein Schwarz/Weiß-Raster. Ansonsten sind Grauabstufungen oder auch Farben möglich, wobei unterschiedliche Farbräume üblich sind.
- Idealerweise sind Rasterpunkte symmetrisch (etwa rotationssymmetrische Kreise oder achsensymmetrische Quadrate). Jedoch kommen auch Ellipsen oder Rechtecke vor. Auch ist es (TFT-Bildschirme!) denkbar, dass ein Pixel aus mehreren Sub-Pixeln besteht, die unterschiedliche Farben darstellen und nur benachbart sind.
- METAFONT unterstützt nur Schwarz/Weiß-Raster, kann aber mit Verzerrungen umgehen. PostScript unterstützt in jedem Falle Farben, selbst wenn das Ausgabe-Gerät nur ein Schwarz/Weiß-Raster anbietet.

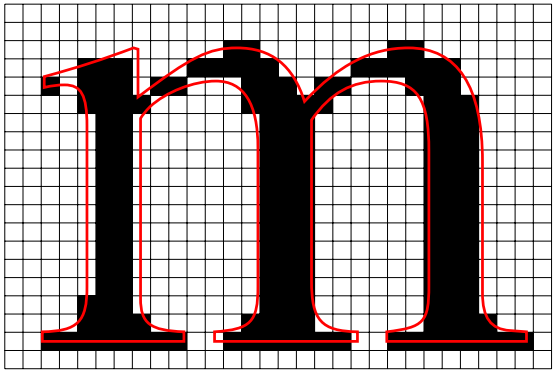


- Gegeben sei eine Kurve, die schwarz auszufüllen ist.
- Ferner sei eine Abbildung gegeben, die das verwendete Koordinatensystem auf das Raster abbildet. Idealerweise sei angenommen, dass die Rasterfelder quadratisch sind.
- Dann werden genau die Rasterfelder schwarz, deren Mittelpunkt (im Ausgangskoordinatensystem) innerhalb des Pfades liegt.

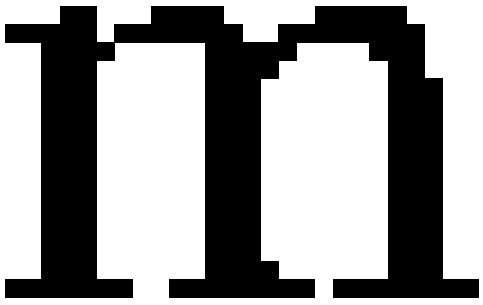


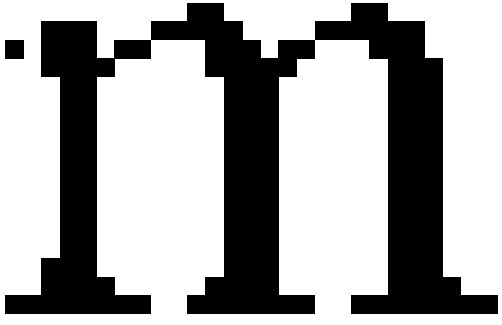
- Gegeben sei eine Kurve, die schwarz auszufüllen ist.
- Ferner sei eine Abbildung gegeben, die das verwendete Koordinatensystem auf das Raster abbildet. Idealerweise sei angenommen, dass die Rasterfelder quadratisch sind.
- Dann werden genau die Rasterfelder schwarz, deren Mittelpunkt (im Ausgangskoordinatensystem) innerhalb des Pfades liegt.



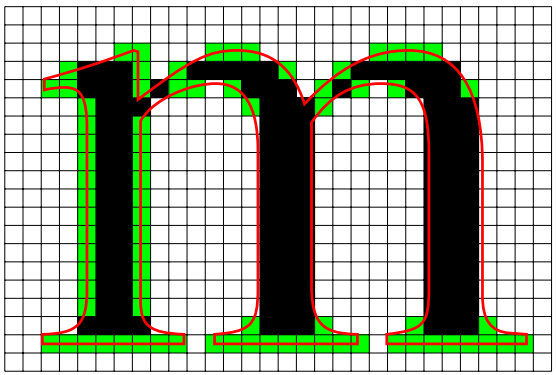


- Die Problematik dieses Verfahrens liegt in der hohen Empfindlichkeit gegen Verschiebungen.
- Keine der beiden Rasterungen für das kleine m ist optimal. Die untere Variante ist aber katastrophal im Vergleich zu der ersten, weil sie um eine halbe Rasterweite nach rechts und nach oben verschoben worden ist.
- In der oberen Variante ist der rechte Zwischenraum breiter als der linke.
- In der unteren Variante ist der linke Schaft deutlich schmaler als die rechten beiden Schäfte. Ferner sind die Serifen deutlich unregelmäßig, und die linke obere Serife ist sogar vom Rest des Buchstabens abgetrennt.



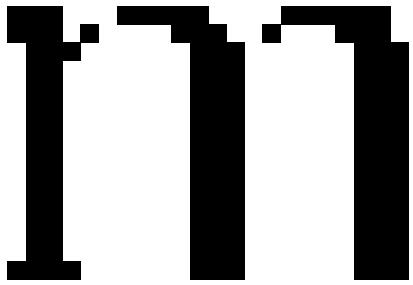


- Die Problematik dieses Verfahrens liegt in der hohen Empfindlichkeit gegen Verschiebungen.
- Keine der beiden Rasterungen für das kleine m ist optimal. Die untere Variante ist aber katastrophal im Vergleich zu der ersten, weil sie um eine halbe Rasterweite nach rechts und nach oben verschoben worden ist.
- In der oberen Variante ist der rechte Zwischenraum breiter als der linke.
- In der unteren Variante ist der linke Schaft deutlich schmaler als die rechten beiden Schäfte. Ferner sind die Serifen deutlich unregelmäßig, und die linke obere Serife ist sogar vom Rest des Buchstabens abgetrennt.

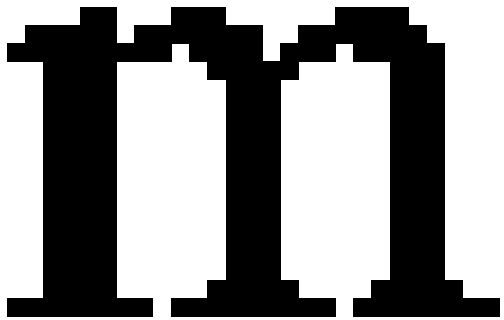


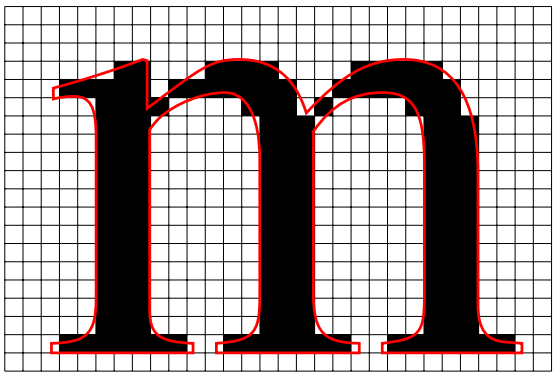
- Die Rasterung einer Fläche ist besonders dort problematisch, wo ihr Rand tangential zu waagrechten oder senkrechten Linien verläuft.
- Wenn diese Linien in etwa der Mitte eines Rasterfeldes verlaufen, ist die Rasterung extrem instabil. Geringste Verschiebungen führen dann dazu, dass ganze Spalten oder Zeilen von Rasterfeldern hinzukommen oder verschwinden.
- Erstrebenswert ist es daher, die auszufüllende Kurve von Anfang an so zu konfigurieren, dass die gefährdeten Ränder entlang der Rastergrenzen verlaufen.
- Links sind alle »Wackelkandidaten« grün gefärbt.

- Variante ohne »Wackelkandidaten«.

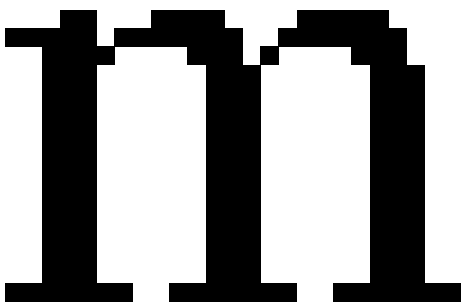


- Variante mitsamt allen »Wackelkandidaten«.

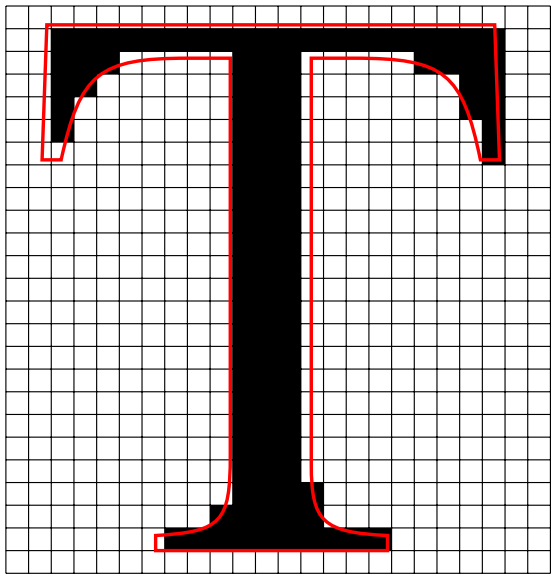




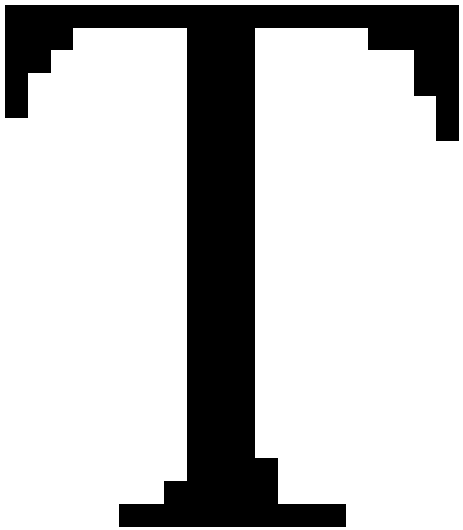
- Bei dieser Version des m wurden die Ränder der drei Schäfte auf die Rastergrenzen ausgerichtet.
- Das war nur möglich, indem alle drei Schäfte etwas zusammenrückten.
- Eine gute Rasterung lässt sich nur erreichen, wenn die Form in geeigneter Weise an die Rasterung angepasst wird.
- METAFONT versucht dies zu automatisieren, wenn die Variable `autoround` einen positiven Wert besitzt.



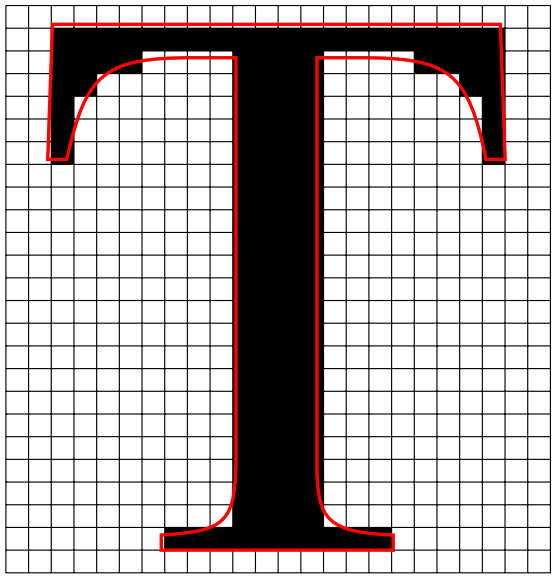
- Bei dieser Version des m wurden die Ränder der drei Schäfte auf die Rastergrenzen ausgerichtet.
- Das war nur möglich, indem alle drei Schäfte etwas zusammenrückten.
- Eine gute Rasterung lässt sich nur erreichen, wenn die Form in geeigneter Weise an die Rasterung angepasst wird.
- METAFONT versucht dies zu automatisieren, wenn die Variable `autoround` einen positiven Wert besitzt.



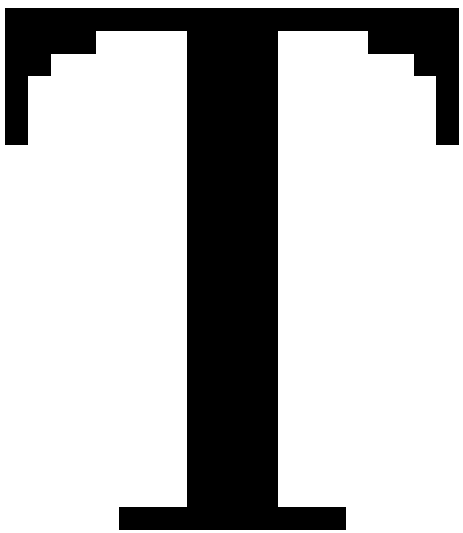
- Selbst dann, wenn bei symmetrisch gestalteten Formen beide Seiten getrennt betrachtet akzeptabel sind, würde das Auge eine Asymmetrie sofort wahrnehmen.
- Beim T liegt hier das Problem nicht nur darin, dass die rechte Seite des Schafts instabil ist. Es fällt auch auf, dass die oberen Serifen ungleichmäßig gestaltet sind.



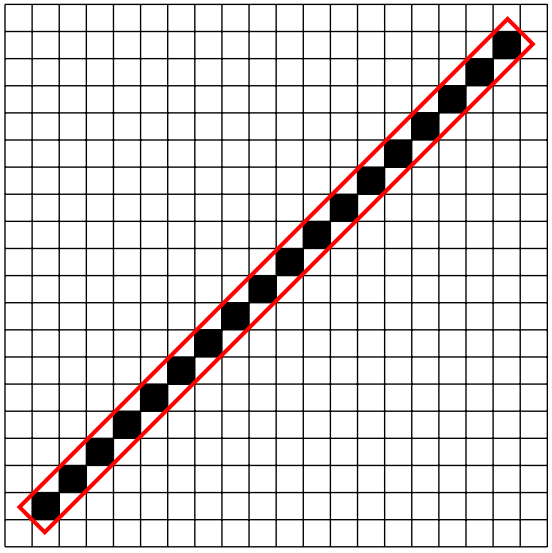
- Selbst dann, wenn bei symmetrisch gestalteten Formen beide Seiten getrennt betrachtet akzeptabel sind, würde das Auge eine Asymmetrie sofort wahrnehmen.
- Beim T liegt hier das Problem nicht nur darin, dass die rechte Seite des Schafts instabil ist. Es fällt auch auf, dass die oberen Serifen ungleichmäßig gestaltet sind.



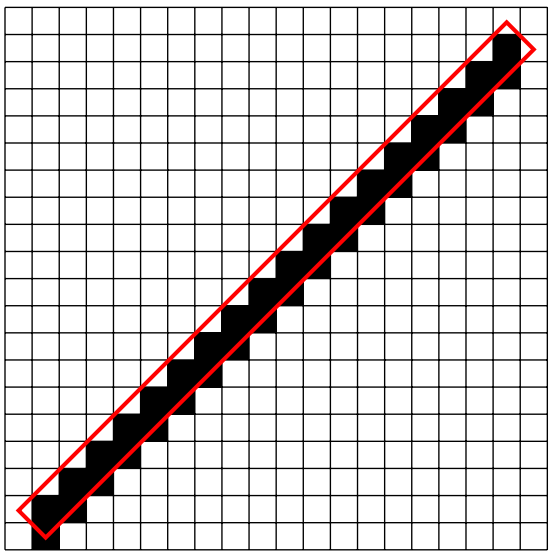
- Das Problem kann nur zuverlässig gelöst werden, wenn die Symmetrie-Achse der Form so gelegt wird, dass sie durch die Mittelpunkte der Rasterfelder verläuft.
- Dessen ungeachtet sind weiterhin Ränder der Form mit horizontalen oder senkrechten Tangenten auf die Rasterkanten auszurichten, auch wenn dies zur Umgestaltung der Form führt.



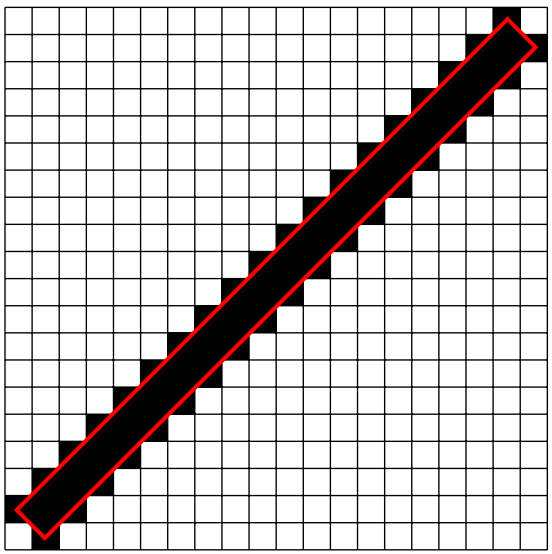
- Das Problem kann nur zuverlässig gelöst werden, wenn die Symmetrie-Achse der Form so gelegt wird, dass sie durch die Mittelpunkte der Rasterfelder verläuft.
- Dessen ungeachtet sind weiterhin Ränder der Form mit horizontalen oder senkrechten Tangenten auf die Rasterkanten auszurichten, auch wenn dies zur Umgestaltung der Form führt.



- Auch diagonal geführte Linien können Instabilitäten aufweisen.
- Diese Diagonale hat eine Breite von etwa $r\sqrt{2}$, wobei r für die Rasterseitenlänge steht.
- So sieht das Ergebnis aus, wenn der Rand der Figur nicht berücksichtigt wird, d.h. wenn ein Rasterfeld-Mittelpunkt genau auf dem Rand liegt, wird er nicht mitgezeichnet.

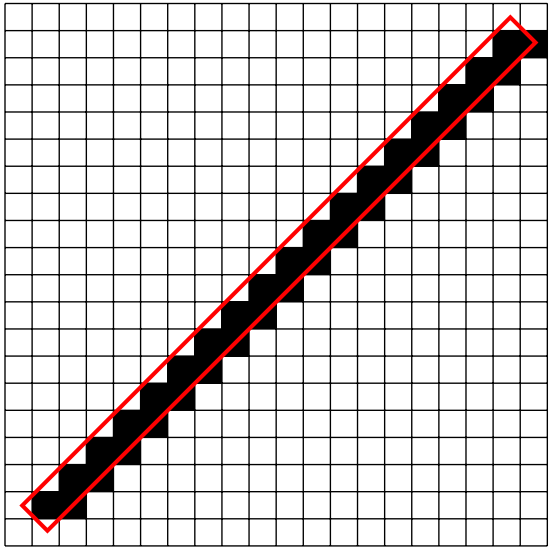


- Wird die Kurve minimal nach unten verschoben, erhalten wir über das Doppelte der vorherigen Rasterpunkte.

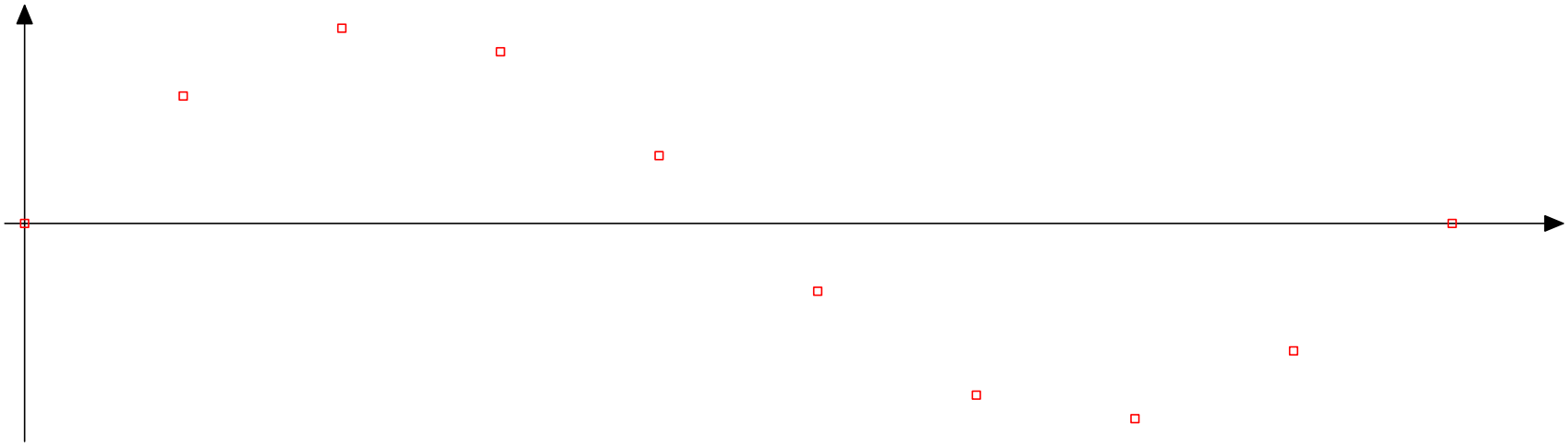


- Wenn der Rand mit berücksichtigt wird bzw. die Breite der Form minimal vergrößert wird, erhalten wir über die dreifache Menge an schwarz gefärbten Rasterpunkten im Vergleich zur Ausgangssituation.

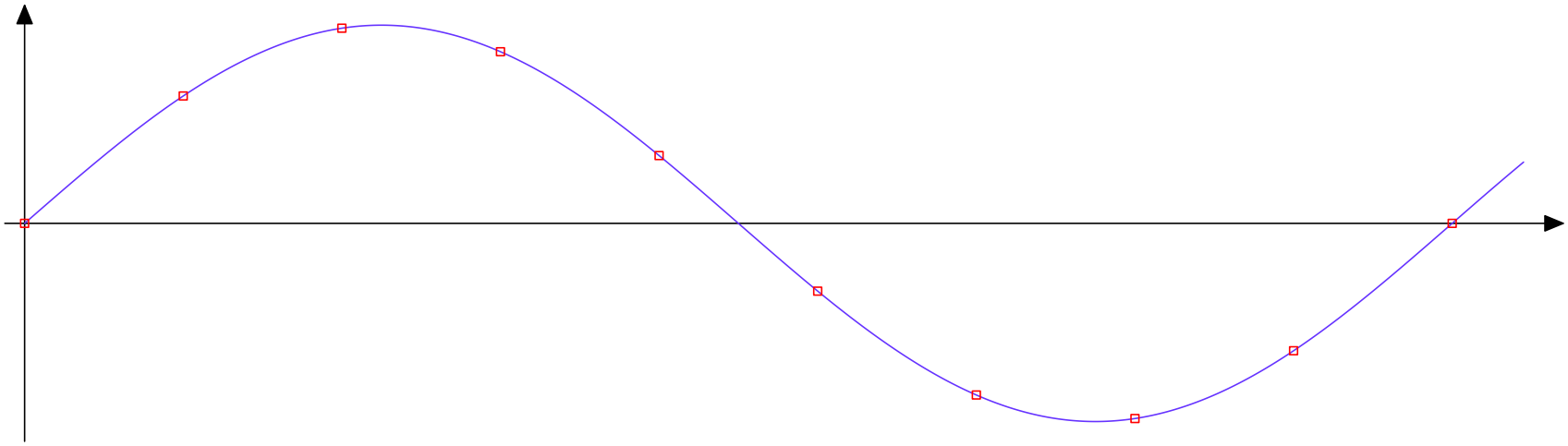
Lösungsansatz für Diagonalen



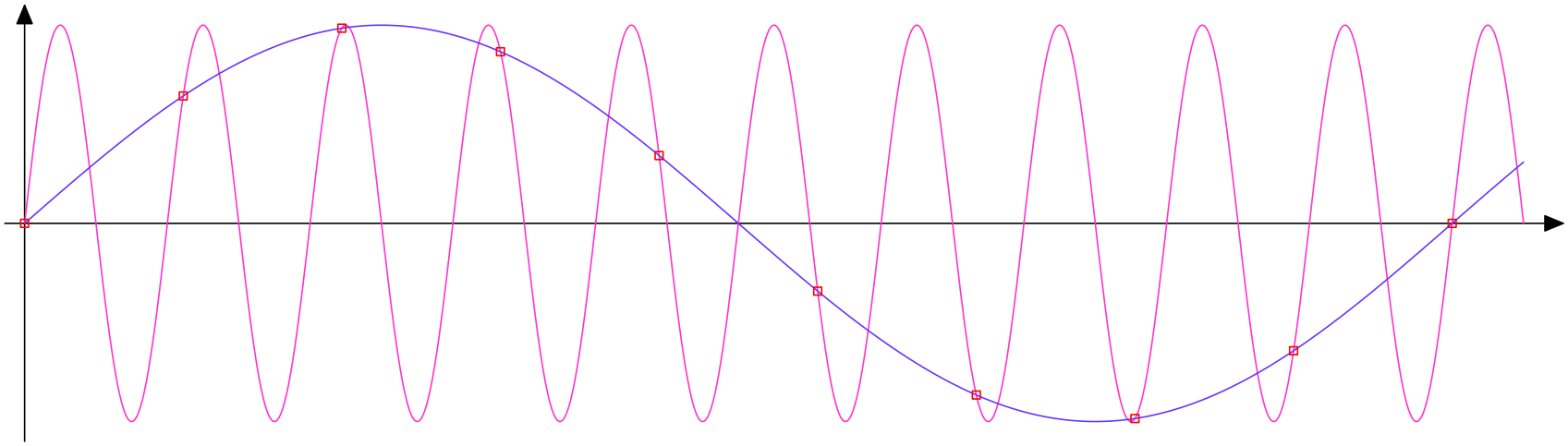
- METAFONT versucht die Frage zu vermeiden, ob ein Rasterfeld zu schwärzen ist, wenn dessen Mittelpunkt genau auf dem Rand der Figur liegt.
- Die Lösung besteht darin, dass METAFONT implizit alle Punkte um $(\delta, \delta\epsilon)$ verschiebt, wobei δ eine sehr kleine Größe ist und $\delta\epsilon < \frac{\delta}{2}$ ist.
- Dies ist auch hilfreich, wenn nicht Figuren zu füllen sind, sondern entlang ihres Randes mit einem Stift zu zeichnen sind. Hier würde eine Kurve $c(t) = (t, t)$ bei $t = \frac{1}{2}$ bei einer Rundung auf ganze Zahlen von $(0, 0)$ auf $(1, 1)$ springen, obwohl die beiden zugehörigen Rasterfelder sich nur an den Eckfeldern berühren. Wenn die obige Verschiebung hinzukommt, dann kommt auch das Rasterfeld $(1, 0)$ bei $t = \frac{1}{2} - 2\delta\epsilon$ hinzu.



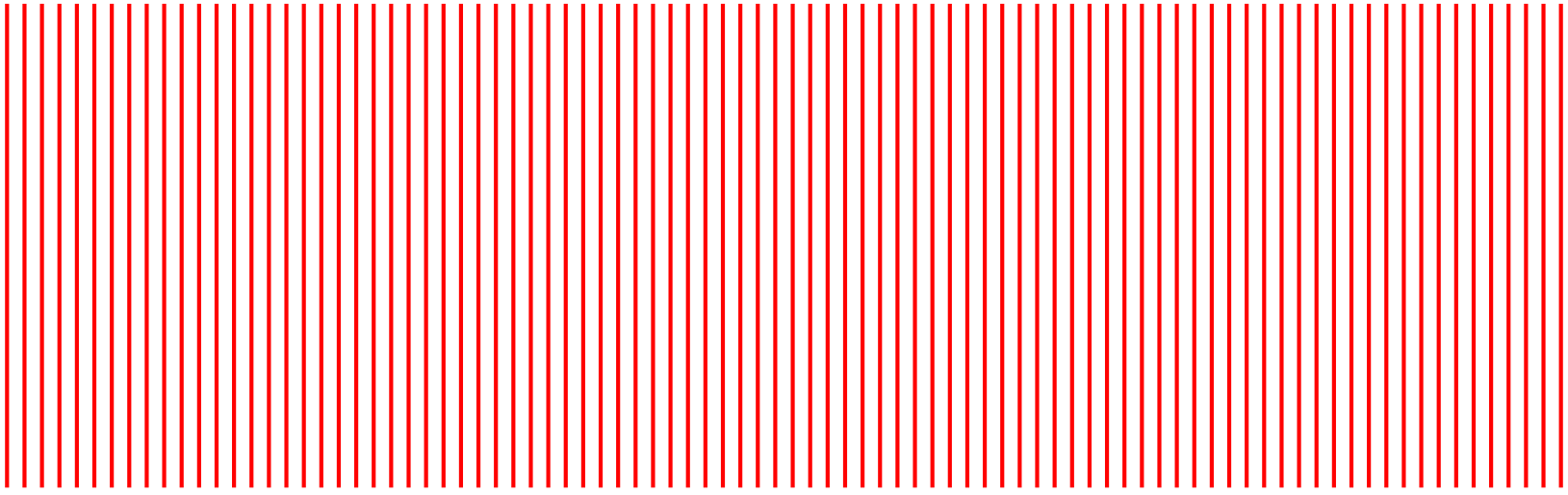
- Ein Problem der Signalverarbeitung: Ein zu digitalisierendes Signal wird zu bestimmten Zeitpunkten gemessen. Aus den Messwerten ist danach eine Näherung des ursprünglichen Signals zu rekonstruieren.



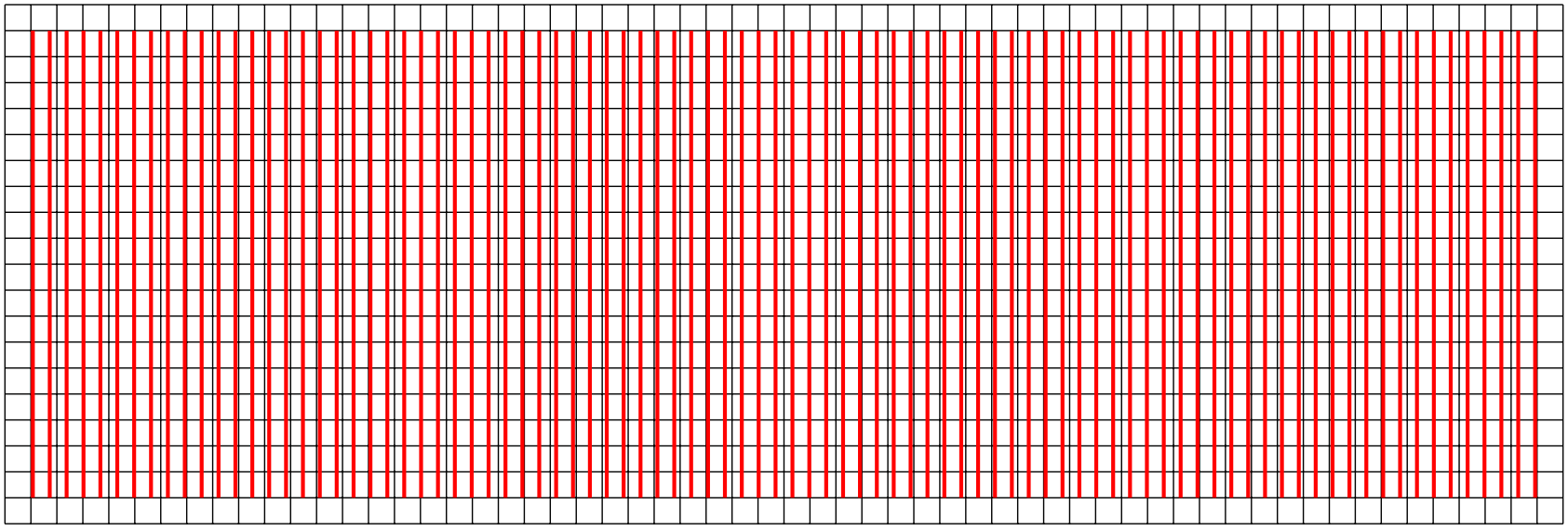
- Durch die gemessenen Punkte passt die dargestellte Sinus-Kurve.



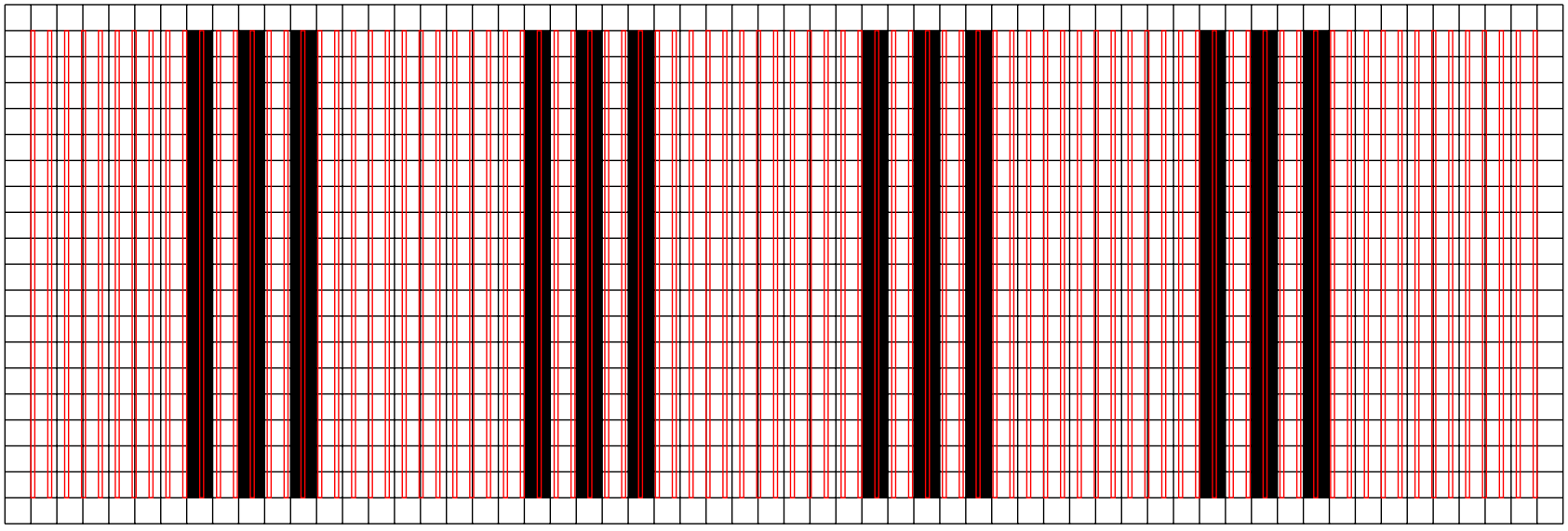
- Aber gleichzeitig passt auch eine sehr viel höherfrequentige Sinus-Kurve durch die gleichen Messwerte.
- Die Signale, die neben dem originalen Signal zu den gleichen Messwerten führen und damit alternativ rekonstruiert werden können, werden Aliase genannt.
- Der Begriff entstand bei dem 1919 entwickelten Überlagerungsempfänger und beschrieb den Effekt, dass ein Sender an zwei beim Oszillator einstellbaren Frequenzen zu hören ist. Im Deutschen wird dafür auch der Begriff Spiegelfrequenz verwendet.



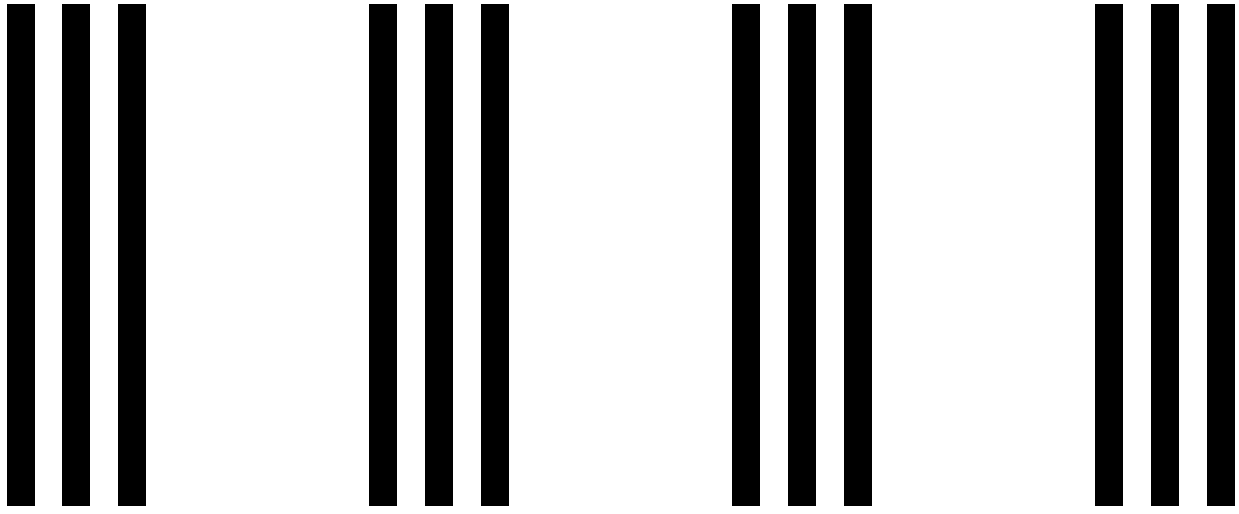
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster.



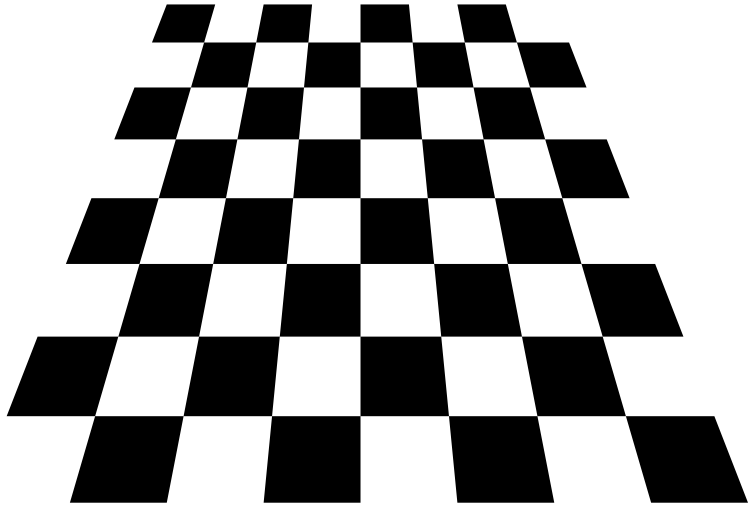
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster, das mit einem vergleichsweise groben Gitter gerastert wird.



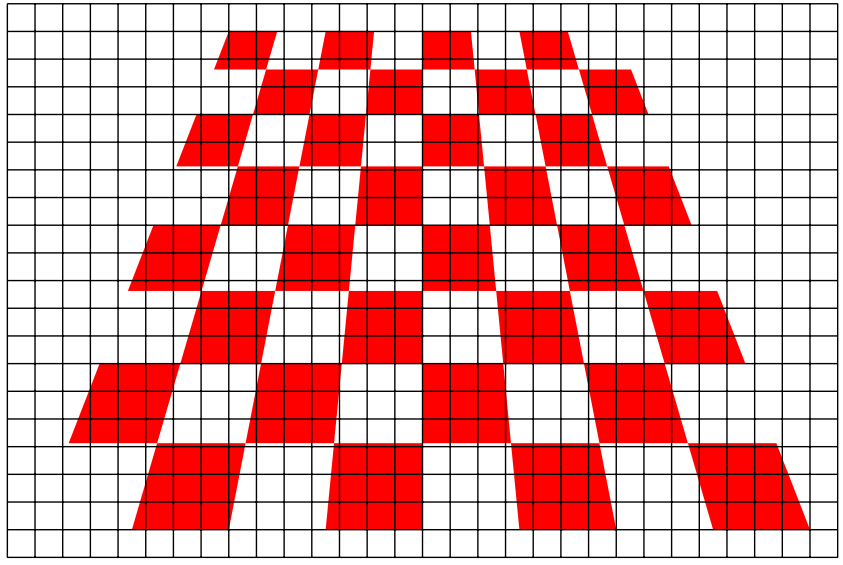
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster, das mit einem vergleichsweise groben Gitter gerastert wird.
- Dann entsteht ein neues Linienmuster, das keine Ähnlichkeit mit dem ursprünglichen Muster hat.



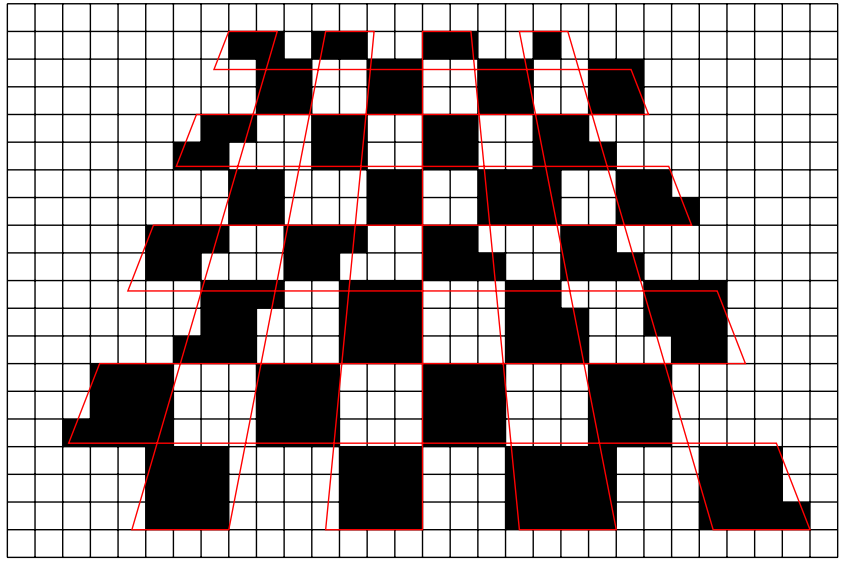
- Bei dem neu entstandenen Linienmuster handelt es sich um einen Alias des Ausgangs-Musters in Bezug auf die gegebene Rasterung.
- Der Begriff »moiré« kommt aus dem Französischen und bezeichnet einen Seidenstoff, der sich durch Gittermuster auszeichnet.



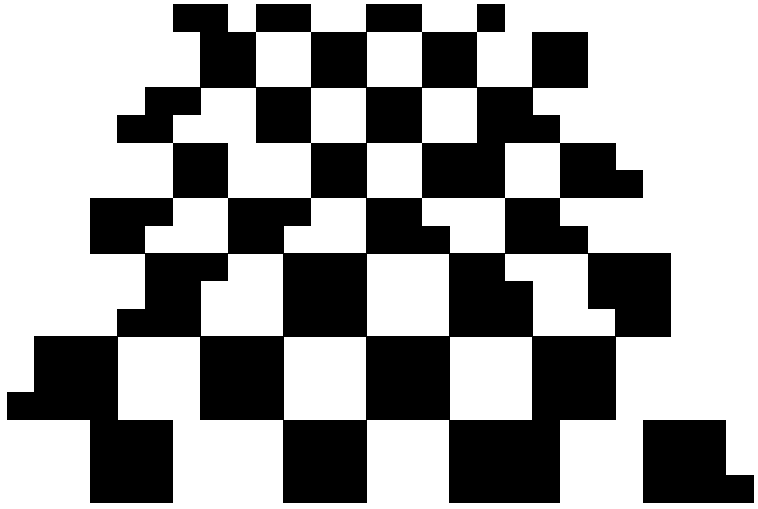
- Gegeben sei ein Schachbrett-Muster, das perspektivisch dargestellt ist mit immer kleiner werdenden Flächen.
- So etwas ist immer eine Herausforderung für eine Rasterung, da das Ausrichten und geringfügige Verformen hier nicht weiterhelfen...



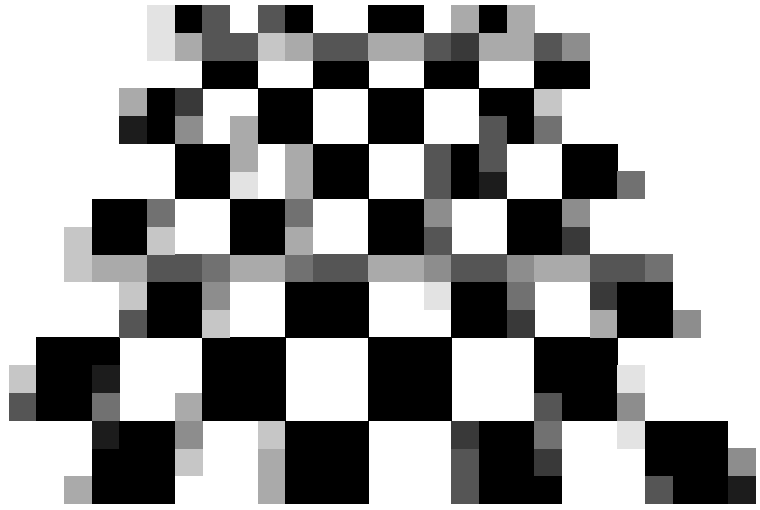
- Gegeben sei ein Schachbrett-Muster, das perspektivisch dargestellt ist mit immer kleiner werdenden Flächen.
- So etwas ist immer eine Herausforderung für eine Rasterung, da das Ausrichten und geringfügige Verformen hier nicht weiterhelfen...



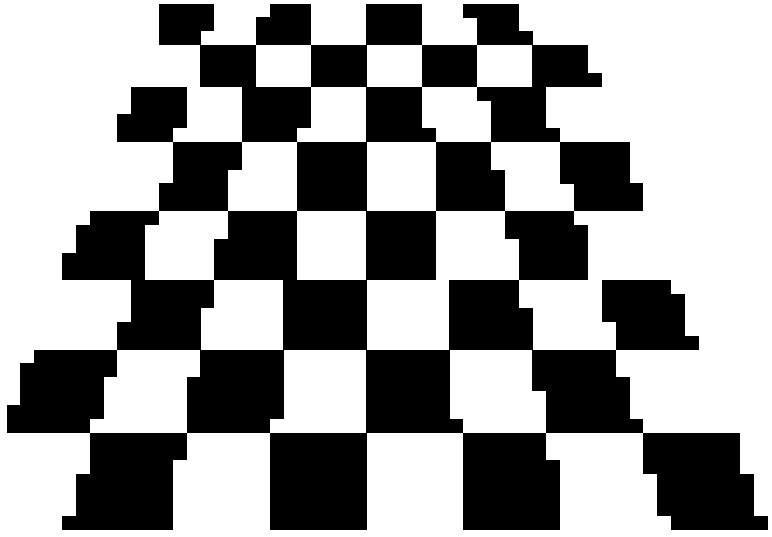
- Wenn das Schachbrett-Muster auf einfache Weise gerastert wird, ist wohl noch die vorderste Reihe in Ordnung. Wenn das Muster jedoch zu klein wird, ist es nach der Rasterung kaum noch zu erkennen.



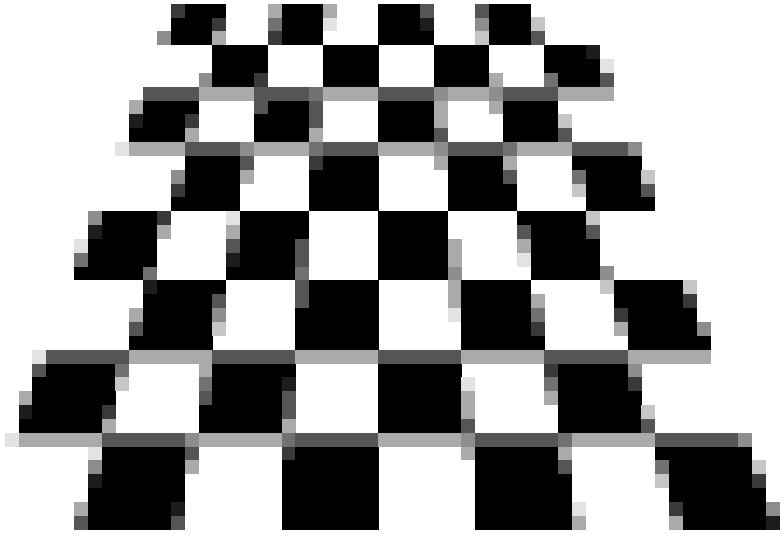
- Wenn das Schachbrett-Muster auf einfache Weise gerastert wird, ist wohl noch die vorderste Reihe in Ordnung. Wenn das Muster jedoch zu klein wird, ist es nach der Rasterung kaum noch zu erkennen.



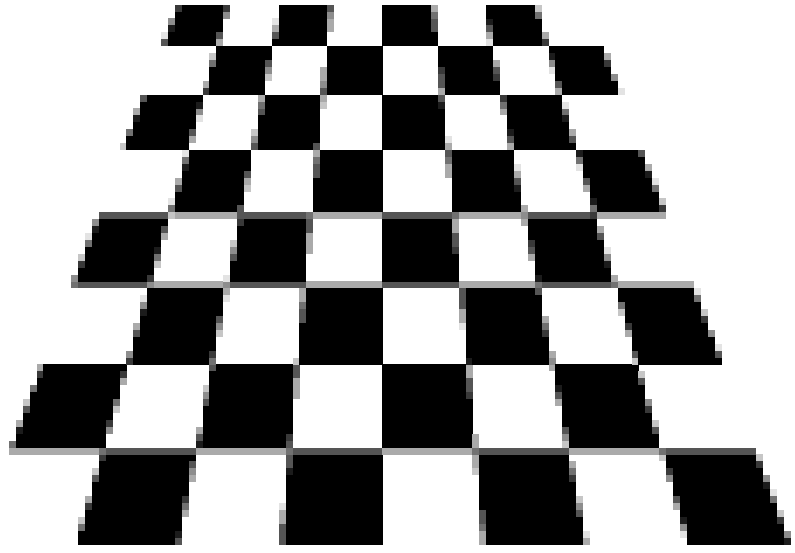
- Wenn auch Graustufen neben Schwarz und Weiß zur Verfügung stehen, bietet es sich an, bei nur teilweise bedeckten Rasterfeldern einen Grauwert zu verwenden.
- In diesem Beispiel wurden in jedem Rasterfeld insgesamt 9 Punkte getestet, ob sie in die Fläche fallen oder nicht. Der Grauwert wurde dann entsprechend des Anteils der in die Kurve fallenden Punkte gewählt.
- Das Verfahren, Graustufen (bzw. Mischfarben) bei teilweise gefüllten Rasterfeldern zu verwenden, wird Anti-Aliasing genannt.



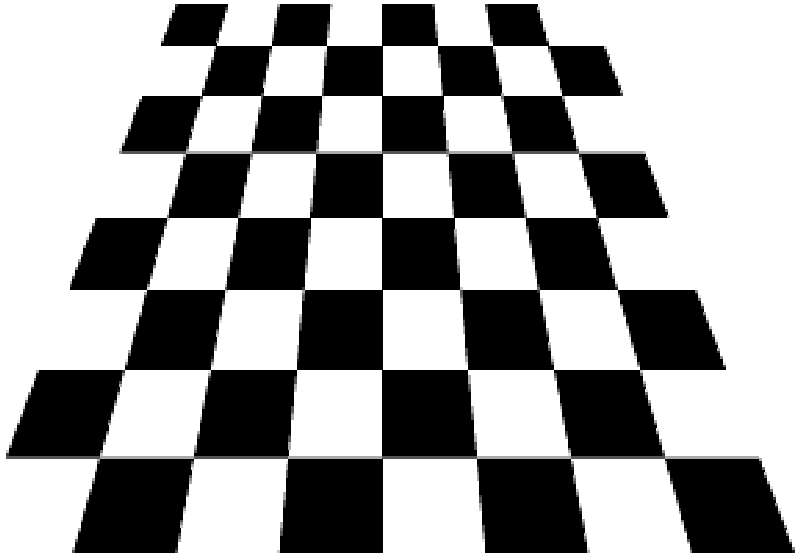
- Wenn wir die Rasterung verdoppeln, können wir das ursprüngliche Muster besser erkennen. Aber es sieht immer noch etwas holprig aus.



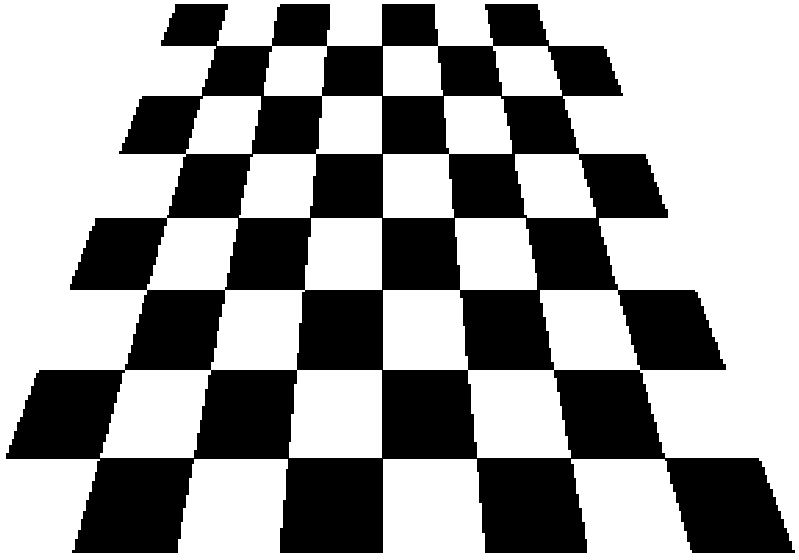
- Anti-Aliasing erleichtert hier dem Auge das Wiedererkennen des bekannten Schachbrettmusters, da die holprigen Kanten wegfallen.
- Das funktioniert aber nur, wenn die Rasterung so fein ist, dass die Grautöne nicht mehr bewusst wahrgenommen werden.



- Unser Auge rastert ebenfalls.
- Ziel ist es daher, ein mit der Rasterung des Mediums darstellbares Bild zu finden, das in Bezug auf das Auge ein Alias zum originalen ungerasterten Bild ist.
- Die Bildschirmauflösung reicht dafür in der Regel aus, jedoch ist Anti-Aliasing unverzichtbar.



- Noch einmal etwas feiner gerastet **mit** Anti-Aliasing.



- Noch einmal etwas feiner gerastet **ohne** Anti-Aliasing.

- Bei GhostScript kann der Anti-Aliasing-Algorithmus getrennt für Schriften und Grafiken eingestellt werden:

  `-dTextAlphaBits=`n mit $n = 1, 2$ oder 4
  `-dGraphicsAlphaBits=`n mit $n = 1, 2$ oder 4

- Die Zahl der Messpunkte beträgt dann 2^n pro Rasterfeld. Bei $n = 1$ ist entsprechend Anti-Aliasing ausgeschaltet.

- Das Format der Type-1-Schriften wurde von Adobe zusammen mit PostScript entwickelt und ist seit 1985 in Nutzung.
- Anders als bei PostScript wurde jedoch das Format der Type-1-Schriften erst 1990 veröffentlicht.
- 1994 wurde das Format als Teil 3 des ISO-Standards 9541 unter dem Titel *Glyph Shape Representation* als Standard akzeptiert.
- Trotz der Konkurrenz durch TrueType sind die Type-1-Schriften bis heute auf dem Markt präsent.
- OpenType ist die Nachfolge-Technologie, die sowohl von Adobe als auch Microsoft unterstützt wird. OpenType unterstützt dabei sowohl die Type-1- als auch die TrueType-Technologie und fügt noch einige Erweiterungen hinzu.
- Literatur: *Adobe Type 1 Format* und Yannis Haralambous: *Fonts & Encodings*, O'Reilly.

Grundsätzlich ähneln Type-1-Schriften weitgehend den Type-3-Schriften, abgesehen davon, dass

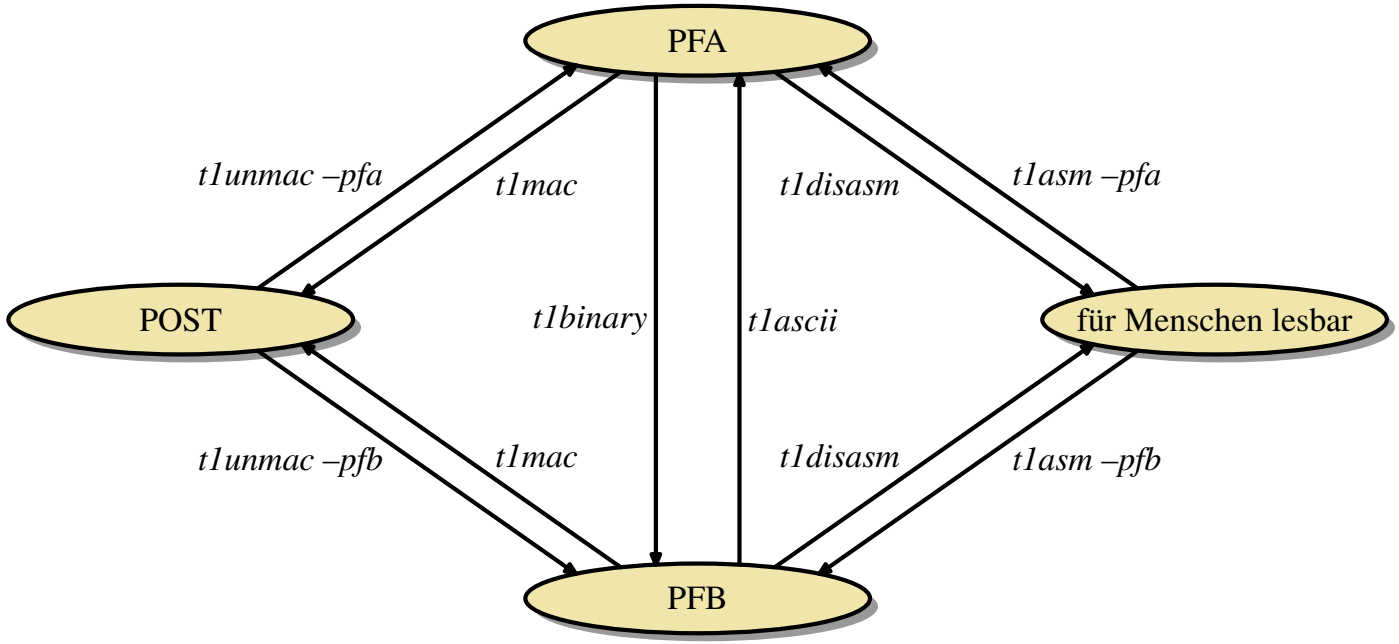
- nicht die Standard-Operatoren von PostScript zur Beschreibung der Kurven verwendet werden, dass
- zusätzlich Hinweise zur Optimierung der Rasterung integriert sind (*hints*) und dass
- das Format größtenteils binär und möglicherweise sogar verschlüsselt ist (auf Basis des *eexec*-Operators, der eine einfache Stromchiffre verwendet, deren Algorithmus und von Adobe verwendeter Schlüssel inzwischen öffentlich sind).

Zu einem Type-1-Schriftschnitt gehören folgende vier Komponenten:

- Ein öffentliches assoziatives Array, das dem eines Type-3-Schriftschnitts ähnelt und in jedem Falle als ASCII-Text abgelegt wird,
- ein privates assoziatives Array, das die Hinweise zur Rasterung aufnimmt und das nur in binärer und möglicherweise sogar nur in verschlüsselter Form vorliegt,
- Prozeduren und
- die Beschreibung der Kurven, die wiederum nur in binärer und möglicherweise sogar in verschlüsselter Form vorliegen.

Varianten des Formats: PFA, PFB und POST

- Unter Unix sind die Formate PFA (A wie ASCII) und PFB (B wie binär) üblich. Bei PFB werden die binären Teile hexadezimal dargestellt.
- Für den Macintosh gibt es das POST-Format.
- Alle drei Varianten lassen sich dank den Werkzeugen von Lee Hetherington und Claire Connelly ineinander überführen.



- Konvertierungswerkzeuge von `http://www.lcdf.org/type/`

utmr8a.disasm

```
%!PS-AdobeFont-1.0: NimbusRomNo9L-Regu 1.05
% ...
12 dict begin
/FontName /NimbusRomNo9L-Regu def
% ... more contents of the public dictionary ...
currentdict end
currentfile eexec
dup /Private 15 dict dup begin
% ... private dictionary ...
end
readonly put
noaccess put
dup /FontName get exch definefont pop
mark currentfile closefile
cleartomark
```

- Hier und im folgenden werden Auszüge des Times-Roman-Schriftschnitts von URW++ präsentiert, der unter der GPL zur Verfügung steht. Betrachtet wird der Text, den *t1disasm* generiert hat.

Schlüssel	Typ	Bedeutung
FontType	integer	1
FontMatrix	array	Transformations-Matrix für die Kurven innerhalb der Definitionen für die Zeichen; typisch ist ein 1000 × 1000-Koordinatensystem
FontInfo	dict	assoziatives Array mit weiteren Feldern, die den Schriftschnitt beschreiben
Encoding	dict	bildet Werte aus dem Bereich $[0, 255]$ in Namen für die einzelnen Zeichen ab
FontBBox	array	Bounding-Box aller übereinander gezeichneter Zeichen
CharStrings	dict	Kurven der einzelnen Zeichen

Schlüssel	Typ	Bedeutung
FontName	name	Name des Schriftschnitts
PaintType	integer	1 (fill) oder 2 (stroke)
Private	dict	Hinweise zur Rasterung
UniqueID	integer	Eindeutige Zahl zwischen 0 und $2^{24} - 1$
XUID	array	Alternativ zu UniqueID: Eindeutige Folge von Zahlen, wobei die erste von Adobe vergeben wird

utmr8a.disasm

```
/FontName /NimbusRomNo9L-Regu def
/PaintType 0 def
/WMode 0 def
/FontBBox {-168 -281 1000 924} readonly def
/FontType 1 def
/FontMatrix [0.001 0.0 0.0 0.001 0.0 0.0] readonly def
/Encoding StandardEncoding def
/UniqueID 5020931 def
```

- Die Einträge für `/CharStrings` und `/Private` kommen erst nachträglich hinzu.

Einträge im öffentlichen assoziativen Array 410

utmr8a.disasm

```
/FontInfo 10 dict dup begin
/version (1.05) readonly def
/Notice ((URW)++,Copyright 1999 by (URW)++ ...) readonly def
/Copyright (Copyright (URW)++ ...) readonly def
/FullName (Nimbus Roman No9 L Regular) readonly def
/FamilyName (Nimbus Roman No9 L) readonly def
/Weight (Regular) readonly def
/ItalicAngle 0.0 def
/isFixedPitch false def
/UnderlinePosition -100 def
/UnderlineThickness 50 def
end readonly def
```

- `/FontInfo` gehört mit zum öffentlichen assoziativen Array und ist selbst eines.

- Im Prinzip können die Namen der einzelnen Zeichen, die in `/Encoding` aufgenommen werden, beliebig gewählt werden.
- In Bezug auf PDF kann das ein Problem sein: Wenn interaktiv nach einem Text in einem Dokument gesucht wird, wie kann festgestellt werden, welches Zeichen im Schriftschnitt welchem Zeichen (etwa in Unicode) entspricht?
- Die Lösung besteht darin, dass die Namen der einzelnen Zeichen gewissen Konventionen entsprechen müssen.
- Das wurde ursprünglich festgelegt über ISO 10036 zu Zeiten als Unicode noch nicht zur Verfügung stand. Siehe `http://10036ra.org/`
- Inzwischen wurde die Adobe Glyph List (AGL) eingefroren: `https://github.com/adobe-type-tools/agl-aglfn/blob/master/glyphlist.txt`
- Neue Namen werden entsprechend der *Adobe Glyph List Specification* gebildet, die sich jeweils einer Sequenz von Unicode-Codepoints zuordnen lassen.

utmr8a.disasm

```
currentfile eexec
```

- `currentfile` ist ein Operator, der einen Verweis auf die aktuelle Eingabequelle (mit dem Type-1-Schriftschnitt) zurückliefert.
- `eexec` nimmt den Verweis auf eine Eingabedatei, um ihn die damit referenzierte Eingabe zu entschlüsseln und anschließend auszuführen.
- In einer PFA-Datei ist bis zu diesem Punkt (abgesehen von den ersten 6 Bytes) alles Klartext und erst danach beginnt das eigentliche binäre Format.

Zum assoziativen Array `/Private` gehören folgende Komponenten:

- Globale Hinweise zur Rasterung,
- mehrere historische Parameter,
- einige Operatoren, die für die Prozeduren benötigt werden,
- die Prozeduren selbst (`/Subrs`) und
- weitere Prozeduren (`/OtherSubrs`).

Zu den binären Daten gehört ferner noch `/CharStrings`, das jedoch erst nachträglich in das übergeordnete assoziative Array eingeordnet wird. Es enthält die Kurvendefinitionen der einzelnen Zeichen einschließlich individueller Hinweise zur Rasterung.

utmr8a.disasm

```
/BlueValues [-20 0 450 470 662 682] def
/BlueScale 0.039625 def
/StdHW [30] def
/StdVW [85] def
/StemSnapH [30 38 43 53 60 73] def
/StemSnapV [78 85 91 103 109 115] def
```

- `/BlueValues` spezifiziert die y-Koordinaten für die wichtigsten Höhenlinien.
- Die optionalen Parameter `/BlueScale` und `/BlueShift` spezifizieren, wie weit die Ausrichtung aufgrund der Höhenlinien erfolgt.
- `/StdHW` und `/StemSnapH` spezifizieren die wichtigsten horizontalen Strichstärken. Analog gibt es `/StdVW` und `/StemSnapV` für die vertikalen Strichstärken.

- Die `/BlueValues` spezifizieren Höhenbereiche für den gesamten Schriftschnitt.
- Sie werden typischerweise genutzt, um die Basislinie zu markieren, die x-Höhe und die Versalhöhe.
- Glatt abschließende Versalien wie etwa das „H“ tangieren den Bereich, gehen aber nicht in diesen hinein.
- Bei runden Zeichen (wie etwa beim „O“ oder „o“) oder spitz abschließenden (wie beim „h“) dringt der Pfad in den markierten Bereich hinein (Überschuss), geht aber über die andere Linie nicht hinaus.

utmr8a.disasm

```
/password 5839 def
/MinFeature {16 16} def
```

- Diese Parameter sind entsprechend der Spezifikation von Adobe aus historischen Gründen notwendig.
- Eine weitere Erklärung gibt es nicht dazu. Entsprechend finden sie sich in dieser Form in jedem Type-1-Schriftschnitt.

utmr8a.disasm

```
/RD {string currentfile exch readstring pop} executeonly def
/ND {noaccess def} executeonly def
/NP {noaccess put} executeonly def
```

- Diese Operatoren müssen ebenfalls innerhalb des `/Private`-Arrays definiert werden.
- `/ND` bzw. `/NP` fügen eine Prozedur in ein assoziatives Array bzw. ein reguläres Array ein und beschränken den Zugriff.

utmr8a.disasm

```
/Subrs 251 array
dup 0 {
   3 0 callothersubr pop pop setcurrentpoint return
} NP
dup 1 {
   0 1 callothersubr return
} NP
dup 2 {
   0 2 callothersubr return
} NP
dup 3 {
   return
} NP
% ...
ND
```

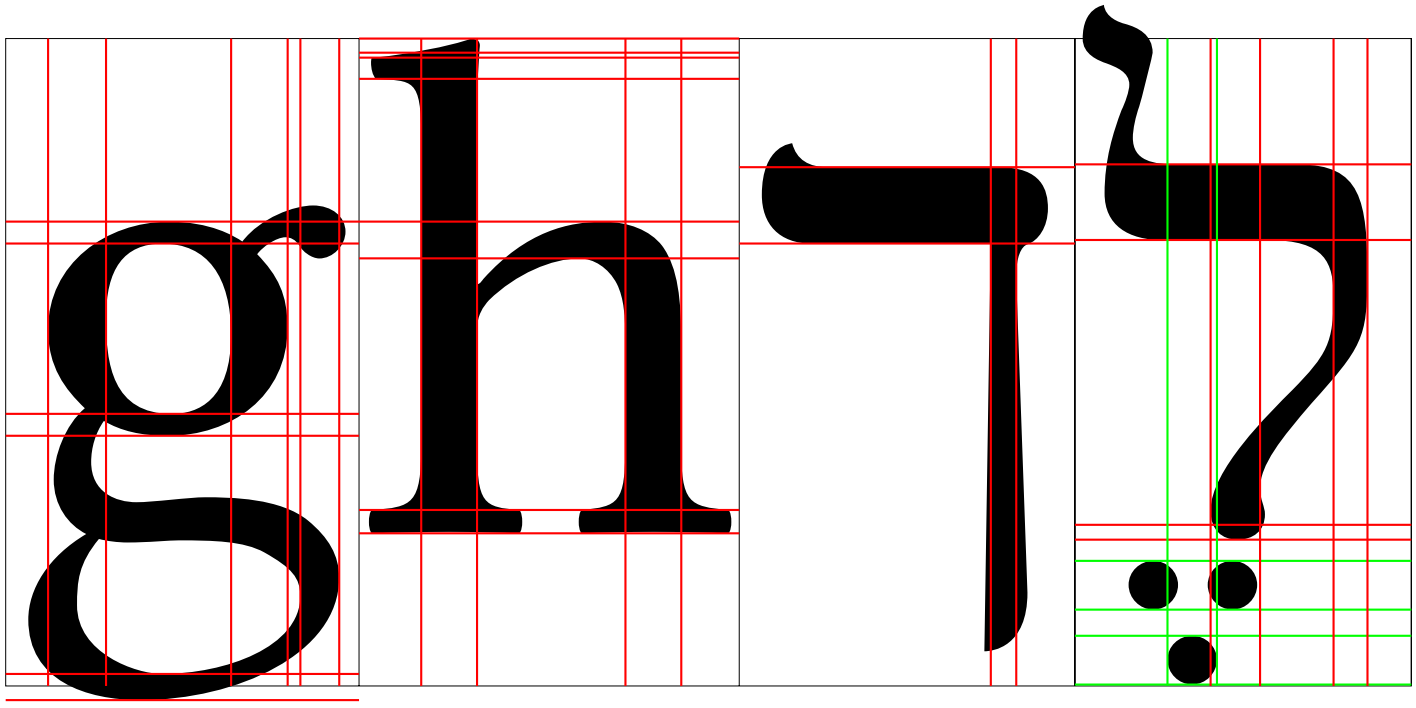
- Die Prozeduren sind in einem regulären Array und somit nur über die jeweilige Nummer erreichbar.
- Die Prozeduren 0 bis 3 sind vordefiniert. Der entsprechende Text wird deswegen normalerweise ignoriert.
- Mit `callothersubr` erfolgt ein Aufruf der anderen Prozeduren. Mit `return` endet die Ausführung einer Prozedur. Ein `return` ist zwingend.

- Der Operator `hsbw` mit den beiden Parametern *sbx* und *wx* spezifiziert mit (*sbx*, 0) den am weitesten links liegenden Punkt der Kurve und den nachfolgenden Weiterbewegungsvektor mit (*wx*, 0).
- Ferner wird die aktuelle Position auf (0, *sbx*) gesetzt. Diese Position geht jedoch nicht in die Kurve selbst ein. Diese muss zwingend mit einer relativen Bewegung beginnen.
- Dies muss der erste Operator sein.
- Alternativ kann `sbw` verwendet werden, falls der Weiterbewegungsvektor eine von 0 abweichende y-Komponente hat.

Operator	Beschreibung
dx dy rmoveto	bewegt sich relativ zur aktuellen Position um (*dx*, *dy*)
dx hmoveto	ist äquivalent zu *dx* `0 rmoveto`
dy vmoveto	ist äquivalent zu `0` *dy* `rmoveto`
dx dy rlineto	Linie mit relativen Koordinaten
dx hlineto	horizontale Linie
dy vlineto	vertikale Linie
dx1 dy1 dx2 dy2 dx3 dy3 rcurveto	Bézier-Kurve, wobei alle Koordinaten relativ zum Ausgangspunkt genommen werden
dx1 dx2 dy2 dy3 hvcurveto	äquivalent zu *dx1* `0` *dx2 dy2* `0` *dy3* `rcurveto`
dy1 dx2 dy2 dx3 vhcurveto	äquivalent zu `0` *dy1 dx2 dy2 dx3* `0 rcurveto`
closepath	Ende einer Kurve; mehrere Kurvendefinitionen sind zulässig; der aktuelle Punkt bleibt bestehen
endchar	Ende einer Zeichendefinition

Operator	Beschreibung
hstem	Spezifikation eines horizonalen Strichs mit der unteren y-Koordinate und der Höhe
vstem	Spezifikation eines vertikalen Schafts mit der niedrigeren x-Koordinate und der Breite
vstem3	adressiert die Problematik des »m«, indem mit insgesamt 6 Parametern die Angaben für drei einzelne `vstem`-Operatoren zusammengefasst werden.
hstem3	analog zu vstem3, jedoch horizontal

Operatoren für individuelle Hinweise zur Rasterung 422



- Das Beispiel zeigt alle mit *hstem* oder *vstem* markierten vertikalen und horizontalen Schäfte der Buchstaben „g“, „h“, dem Kapf (in der Variante für das Ende eines Worts) und das Lamed mit dem Vokalzeichen Segol (drei Punkte), das als Akzent darunter gesetzt wird.
- Schriftschnitt: *LibertinusT1Math*

utmr8a.disasm

```
2 index /CharStrings 316 dict dup begin
% ...
/a {
   % ...
} ND
% ...
/.notdef {
   % ...
} ND
end
```

- Da die Definition textuell in der Deklaration von `/Private` eingebettet ist, muss mit `2 index` explizit das übergeordnete assoziative Array referenziert werden (das immer noch auf dem Stack an genannter Position liegt).
- Die einzelnen Prozeduren enthalten jeweils individuelle Hinweise zur Rasterung und die Kurvendefinition. Hierbei dürfen nur die speziellen Operatoren verwendet werden.
- Die Namen müssen mit denjenigen übereinstimmen, die in `/Encoding` referenziert werden.

Definition des Schnitts für den Buchstaben f 424

utmr8a.disasm

```
/f {
   20 333 hsbw
   0 15 hstem 418 32 hstem 655 28 hstem 83 84 vstem
   289 450 rmoveto
   -123 hlineto 116 vlineto
   58 19 31 38 vhcurveto
   21 0 14 -10 18 -29 rrcurveto
   16 -26 12 -10 17 0 rrcurveto
   23 19 18 23 hvcurveto
   36 -44 26 -60 vhcurveto
   -62 0 -53 -27 -26 -46 rrcurveto
   -26 -44 -8 -36 -1 -80 rrcurveto
   -82 hlineto -32 vlineto 82 hlineto
   -314 vlineto
   0 -73 -11 -12 -72 -4 rrcurveto
   -15 vlineto 260 hlineto 15 vlineto
   -82 3 -11 11 0 75 rrcurveto
   314 vlineto 122 hlineto
   closepath endchar
} ND
```