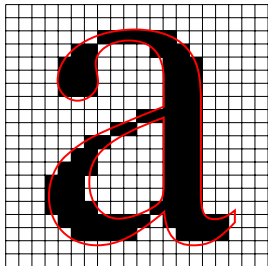
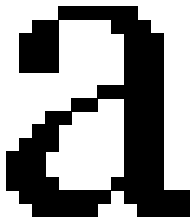


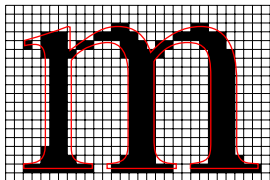
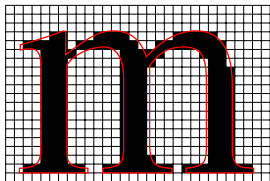
- Am Ende sind die geometrisch definierten Flächen in ein Raster gegebener Größe und Ausformung zu überführen, um sie auf dem Bildschirm oder dem Drucker darstellen zu können.
- Im einfachsten Falle handelt es sich dabei um ein Schwarz/Weiß-Raster. Ansonsten sind Graustufungen oder auch Farben möglich, 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 Ausgangskordinatensystem) innerhalb des Pfades liegt.



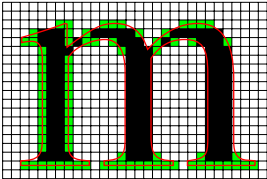
- Gegeben sei eine Kurve, die schwarz auszufüllen ist.
- Ferner sei eine Abbildung gegeben, die das verwendete Koordinatensystem auf das Raster abbildet. Idealerweise sei angenommen, dass die Rasterfelder quadratisch sind.
- Dann werden genau die Rasterfelder schwarz, deren Mittelpunkt (im Ausgangskordinatensystem)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

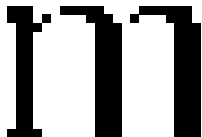


- Die Problematik dieses Verfahrens liegt in der hohen Empfindlichkeit gegen Verschiebungen.
- Keine der beiden Rasterungen für das kleine m ist optimal. Die untere Variante ist aber katastrophal im Vergleich zu der ersten, weil sie um eine halbe Rasterweite nach rechts und nach oben verschoben worden 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.



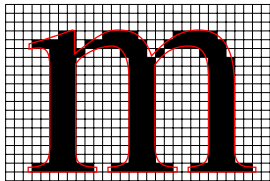
- Die Rasterung einer Fläche ist besonders dort problematisch, wo ihr Rand tangential zu waagrechten oder senkrechten Linien verläuft.
- Wenn diese Linien in etwa der Mitte eines Rasterfeldes verlaufen, ist die Rasterung extrem instabil. Geringste Verschiebungen führen dann dazu, dass ganze Spalten oder 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«.

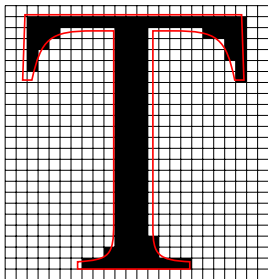




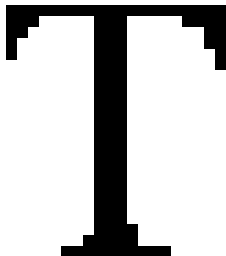
- Bei dieser Version des m wurden die Ränder der drei Schäfte auf die Rastergrenzen ausgerichtet.
- Das war nur möglich, indem alle drei Schäfte etwas zusammenrückten.
- Eine gute Rasterung lässt sich nur erreichen, wenn die Form in geeigneter Weise an die Rasterung angepasst wird.
- METAFONT versucht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



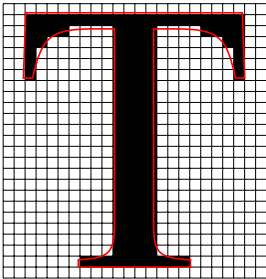
- Bei dieser Version des m wurden die Ränder der drei Schäfte auf die Rastergrenzen ausgerichtet.
- Das war nur möglich, indem alle drei Schäfte etwas zusammenrückten.
- Eine gute Rasterung lässt sich nur erreichen, wenn die Form in geeigneter Weise an die Rasterung angepasst wird.
- METAFONT versucht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



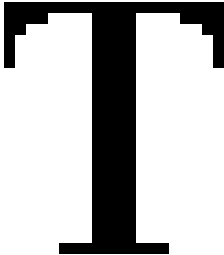
- Selbst dann, wenn bei symmetrisch gestalteten Formen beide Seiten getrennt betrachtet akzeptabel sind, würde das Auge eine Asymmetrie sofort wahrnehmen.
- Beim T liegt hier das Problem nicht nur darin, dass die rechte Seite des Schafts instabil ist. Es fällt auch auf, dass die oberen Serifen ungleichmäßig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



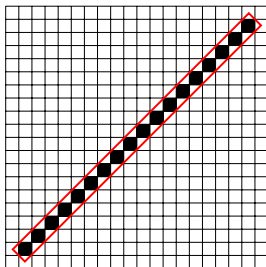
- Selbst dann, wenn bei symmetrisch gestalteten Formen beide Seiten getrennt betrachtet akzeptabel sind, würde das Auge eine Asymmetrie sofort wahrnehmen.
- Beim T liegt hier das Problem nicht nur darin, dass die rechte Seite des Schafts instabil ist. Es fällt auch auf, dass die oberen Serifen ungleichmäßig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



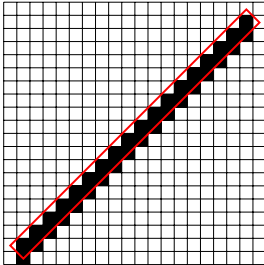
- Das Problem kann nur zuverlässig gelöst werden, wenn die Symmetrie-Achse der Form so gelegt wird, dass sie durch die Mittelpunkte der Rasterfelder verläuft.
- Dessen ungeachtet sind weiterhin Ränder der Form mit horizontalen oder senkrechten Tangenten auf die Rasterkanten auszurichten, auch wenn dies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



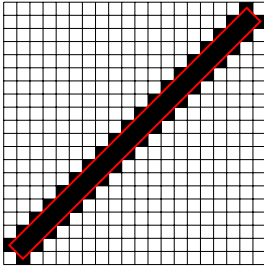
- Das Problem kann nur zuverlässig gelöst werden, wenn die Symmetrie-Achse der Form so gelegt wird, dass sie durch die Mittelpunkte der Rasterfelder verläuft.
- Dessen ungeachtet sind weiterhin Ränder der Form mit horizontalen oder senkrechten Tangenten auf die Rasterkanten auszurichten, auch wenn dies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



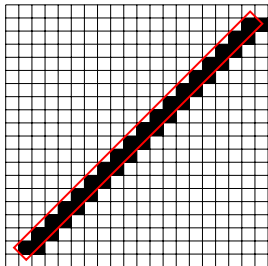
- Auch diagonal geführte Linien können Instabilitäten aufweisen.
- Diese Diagonale hat eine Breite von etwa $r\sqrt{2}$, wobei r für die Rasterseitenlänge steht.
- So sieht das Ergebnis aus, wenn der Rand der Figur nicht berücksichtigt wird, d.h. wenn ein Rasterfeld-Mittelpunkt genau auf dem Rand liegt,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



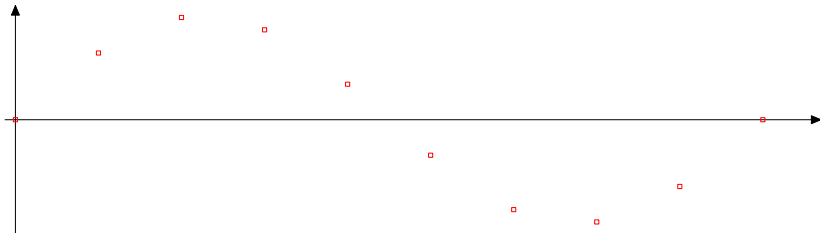
- Wird die Kurve minimal nach unten verschoben, erhalten wir über das Doppelte der vorherigen Rasterpunkte.



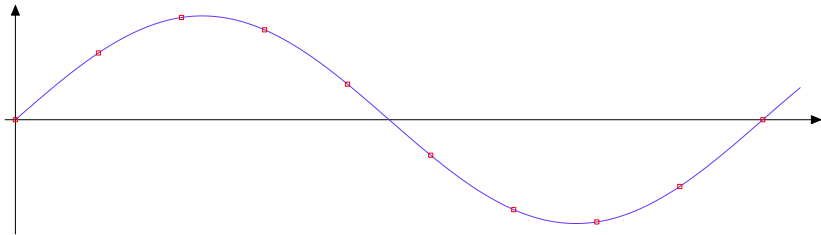
- Wenn der Rand mit berücksichtigt wird bzw. die Breite der Form minimal vergrößert wird, erhalten wir über die dreifache Menge an schwarz gefärbten Rasterpunkten im Vergleich zur Ausgangssituation.



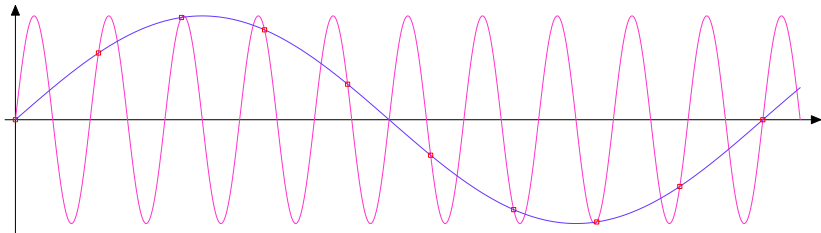
- METAFONT versucht die Frage zu vermeiden, ob ein Rasterfeld zu schwärzen ist, wenn dessen Mittelpunkt genau auf dem Rand der Figur liegt.
- Die Lösung besteht darin, dass METAFONT implizit alle Punkte um $(\delta, \delta\epsilon)$ verschiebt, wobei δ eine sehr kleine Größe ist und $\delta\epsilon < \frac{\delta}{2}$ 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.



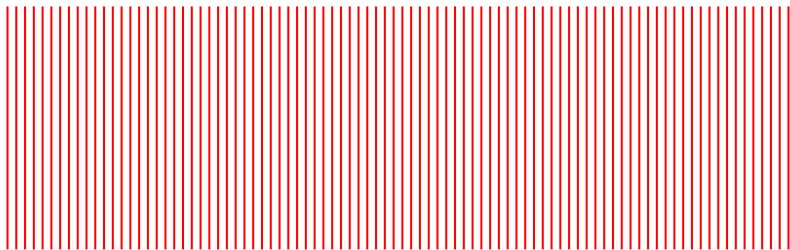
- Ein Problem der Signalverarbeitung: Ein zu digitalisierendes Signal wird zu bestimmten Zeitpunkten gemessen. Aus den Messwerten ist danach eine Näherung des ursprünglichen Signals zu rekonstruieren.



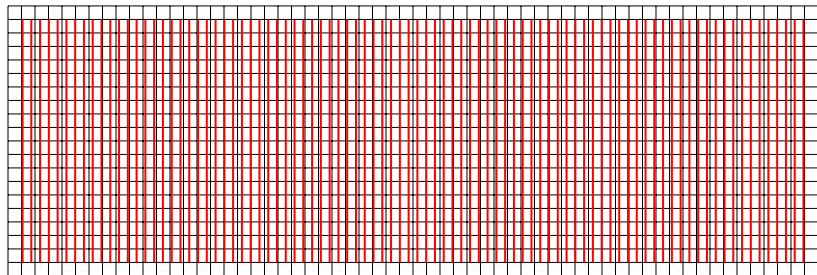
- Durch die gemessenen Punkte passt die dargestellte Sinus-Kurve.



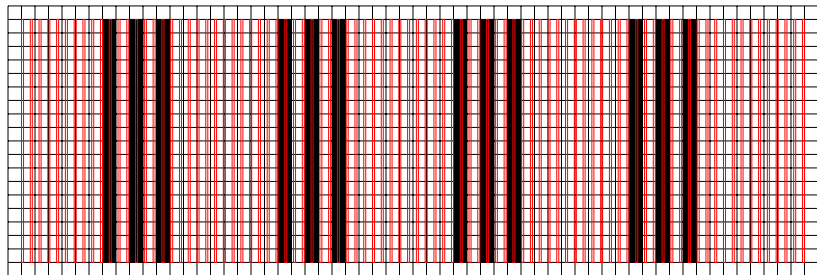
- Aber gleichzeitig passt auch eine sehr viel höherfrequente Sinus-Kurve durch die gleichen Messwerte.
- Die Signale, die neben dem originalen Signal zu den gleichen Messwerten führen und damit alternativ rekonstruiert werden können, werden Aliase genannt.
- Der Begriff entstand bei dem 1919 entwickelten 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.



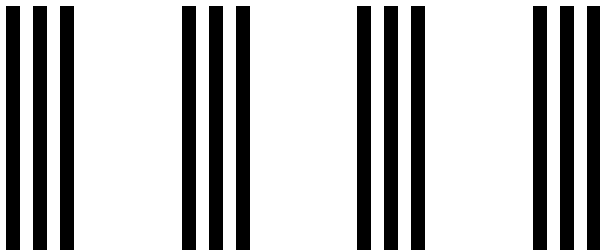
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster.



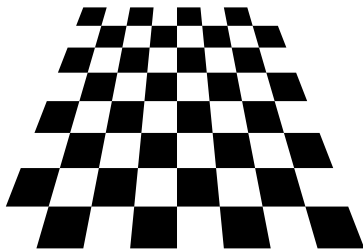
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster, das mit einem vergleichsweise groben Gitter gerastert wird.



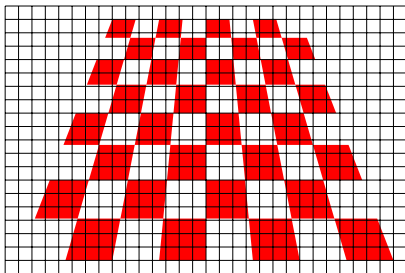
- Die gleiche Problematik findet sich auch bei der Rasterung zweidimensionaler Flächen.
- Gegeben sei ein sehr feines Linienmuster, das mit einem vergleichsweise groben Gitter gerastert wird.
- Dann entsteht ein neues Linienmuster, das keine Ähnlichkeit mit dem ursprünglichen Muster hat.



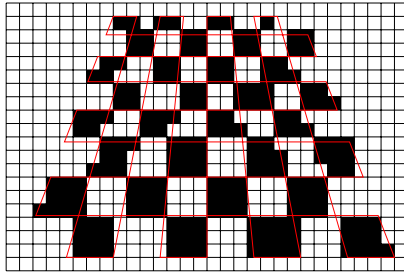
- Bei dem neu entstandenen Linienmuster handelt es sich um einen Alias des Ausgangs-Musters in Bezug auf die gegebene Rasterung.
- Der Begriff »moiré« kommt aus dem Französischen und bezeichnet einen Seidenstoff, der sich durch Gittermuster auszeichnet.



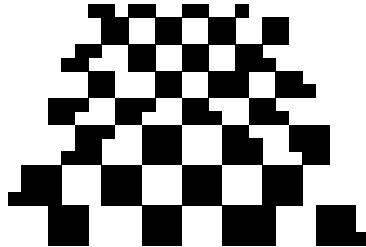
- Gegeben sei ein Schachbrett-Muster, das perspektivisch dargestellt ist mit immer kleiner werdenden Flächen.
- So etwas ist immer eine Herausforderung für eine Rasterung, da das Ausrichten und geringfügige Verformen hier nicht weiterhelfen...



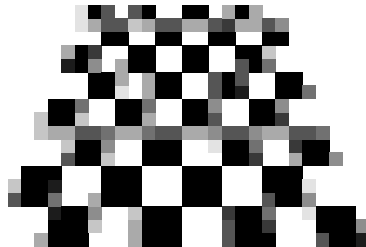
- Gegeben sei ein Schachbrett-Muster, das perspektivisch dargestellt ist mit immer kleiner werdenden Flächen.
- So etwas ist immer eine Herausforderung für eine Rasterung, da das Ausrichten und geringfügige Verformen hier nicht weiterhelfen...



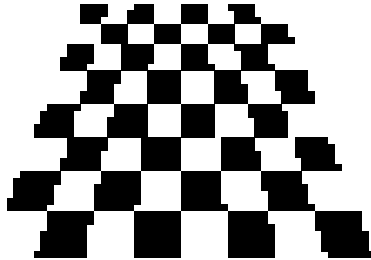
- Wenn das Schachbrett-Muster auf einfache Weise gerastert wird, ist wohl noch die vorderste Reihe in Ordnung. Wenn das Muster jedoch zu klein wird, ist es nach der Rasterung kaum noch zu erkennen.



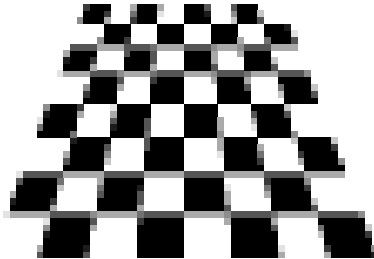
- Wenn das Schachbrett-Muster auf einfache Weise gerastert wird, ist wohl noch die vorderste Reihe in Ordnung. Wenn das Muster jedoch zu klein wird, ist es nach der Rasterung kaum noch zu erkennen.



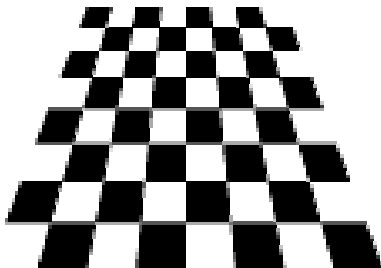
- Wenn auch Graustufen neben Schwarz und Weiß zur Verfügung stehen, bietet es sich an, bei nur teilweise bedeckten Rasterfeldern einen Grauwert zu verwenden.
- In diesem Beispiel wurden in jedem Rasterfeld insgesamt 9 Punkte getestet, ob sie in die Fläche fallen oder nicht. Der Grauwert wurde dann entsprechend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



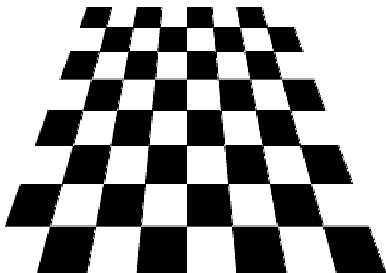
- Wenn wir die Rasterung verdoppeln, können wir das ursprüngliche Muster besser erkennen. Aber es sieht immer noch etwas holprig aus.



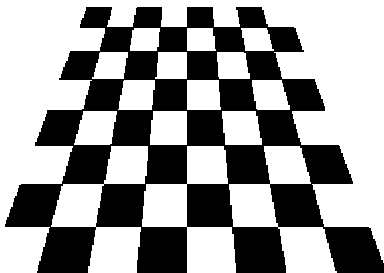
- Anti-Aliasing erleichtert hier dem Auge das Wiedererkennen des bekannten Schachbrettmusters, da die holprigen Kanten wegfallen.
- Das funktioniert aber nur, wenn die Rasterung so fein ist, dass die Grautöne nicht mehr bewusst wahrgenommen werden.



- 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.

- PDF steht für *portable document format*.
- PDF folgt der Tradition von PostScript und ist entsprechend geräte- und plattformunabhängig.
- Im Gegensatz zu PostScript ist PDF optimiert für die interaktive Präsentation von Dokumenten am Bildschirm.
- (Anders als bei NeWS und Display PostScript geht es wirklich weiterhin um Dokumente und nicht um die Nutzung der Mächtigkeit von PostScript für grafische Benutzeroberflächen.)

PDF erfüllt folgende Anforderungen:

- Anders als beim Drucker, der alle Seiten hintereinander druckt, sollten die Seiten einzeln vom Benutzer abgerufen werden können.
- Der Aufwand, eine Seite darzustellen, sollte nur linear abhängen von der Zahl der darzustellenden Elemente.
- Im Normalfall sollte ein PDF-Dokumente alle Informationen einschließlich aller verwendeten Schriftschnitte enthalten.
- Zusätzliche Navigationshilfen stehen zur Verfügung einschließlich hierarchischer Dokumentstrukturen und Links.
- Animationen für Präsentationen werden unterstützt.

- Bei PostScript gibt es einen globalen Zustand, der von dem Programm sukzessive verändert wird.
- Entsprechend ist zur Darstellung der n -ten Seite es notwendig, die Seiten 1 bis $n - 1$ zu erzeugen. Dies ist im allgemeinen Falle unvermeidlich, da die Generierung der vorherigen Seiten den globalen Zustand so verändern kann, dass dadurch die Generierung der n -ten Seite beeinflusst wird.
- Zwar gibt es bei PostScript Strukturierungskonventionen, die die Selektion einer Seite zulassen. Dies beruht aber nur auf den Konventionen, denen nicht entsprochen werden muss und die auch relativ komplex sind. (Darauf beruhen Werkzeuge wie *gv* oder *pselect*.)

- Bei PDF ist über die Sprachdefinition genau geregelt, welche Informationen notwendig sind, um eine Seite zu generieren.
- All diese Informationen können gefunden werden, ohne das gesamte PDF-Dokument durchzulesen. Das ist möglich dank einem Index am Ende eines jeden PDF-Dokuments, über das alle einzelnen PDF-Objekte gefunden werden können.
- Ein globaler Zustand existiert nicht. Entsprechend beginnt jede Seitengenerierung mit einem vordefinierten Zustand.

- PDF kennt keine Schleifen und keine Rekursion.
- Entsprechend hängt der Ausführungsaufwand direkt linear ab von dem Umfang der Beschreibung einer Seite und deren referenzierter Ressourcen wie etwa Schriftschnitte oder gemeinsame (nicht-rekursive!) Prozeduren.
- Das kann allerdings zur Folge haben, dass der Umfang eines PDF-Dokuments volumemäßig im Vergleich zur äquivalenten PostScript-Quelle zunimmt.
- Grundsätzlich beschleunigt dies jedoch die Darstellung.

- Bei PostScript kam die Möglichkeit, Type-1-Schriften einzubetten, recht spät mit *Level 3*.
- GhostScript unterstützt dies bis heute nicht, d.h. bei einer Konvertierung von PDF auf PostScript gehen die eingebetteten Schriften verloren. Stattdessen werden diese für eine vorgegebene Auflösung gerastert. (Eine Alternative ist hier pdftops, das zum xpdf-Paket gehört.)
- Im Gegensatz dazu ist bei PDF das Einbetten von Schriften der Normalfall — selbst wenn es prinzipiell möglich ist, diese auch wegzulassen.
- Ziel der Portabilität ist es, dass jeder mit einem PDF-Leser die Dokumente betrachten kann, ohne eigene Kopien der Schriften auf seinem Rechner zu besitzen.
- Nur so ist auch eine gute Rasterung möglich, die am Bildschirm anspruchsvoller ist als auf dem Drucker.

- 1993 wurde das erste Referenzdokument zu PDF von Adobe veröffentlicht zusammen mit der ersten Version des Acrobat Readers.
- 1994 begann die Unterstützung von PDF durch GhostScript. 1995 kam *xpdf* hinzu.
- 1995 begann die Entwicklung von PDF/X, einer Teilmenge von PDF, die für Druckvorlagen geeignet ist.
- 1997 kam PostScript Level 3 heraus, d.h. erst dann konnten in PostScript Type-1-Schriften eingebettet werden.
- 2001 und 2002 wurde PDF/X in die ISO-Normen 15929 und 15930 übernommen.
- 2005 wurde mit PDF/A eine Teilmenge von PDF für Archivierungen als ISO 19005-1 standardisiert.
- 2006 kam PDF 1.7 heraus – das ist die zur Zeit aktuelle Fassung.
- 2008 wurde PDF 1.7 als ISO-Standard akzeptiert: ISO 32000-1:2008.

- PDF ist ein Datenformat von Programmen für Programmen. Es ist (anders als PostScript) nicht dafür vorgesehen, von Menschen erstellt, verändert oder gelesen zu werden.
- Im Prinzip ist PDF eine lineare Repräsentierung einer komplexen Datenstruktur, die in einzelne sich untereinander referenzierende PDF-Objekte zerlegt worden ist.

- PDF-Objekte haben einen Typ, wobei alle Typen der Sprache PostScript eingeschlossen sind: Boolean, ganze Zahlen, Gleitkommazahlen, Zeichenketten, Arrays und assoziative Arrays (*dictionaries*).
- Hinzu kommen Streams und das *null*-Objekt.
- PDF-Objekte können von anderen PDF-Objekten referenziert werden. Damit dies möglich ist, können PDF-Objekte mit einem Tupel bestehend aus einer ganzen Zahl und einer Versionsnummer (beginnend mit 0) definiert werden.
- Alle so referenzierbaren PDF-Objekte werden *indirekte Objekte* genannt. Sie müssen allesamt im Index am Ende der PDF-Datei aufgeführt werden.
- Die Versionsnummer erlaubt das spätere Ändern von PDF-Objekten, ohne die gesamte PDF-Datei neu erzeugen zu müssen.

- Die folgenden Seiten stellen ein minimales PDF-Beispiel vor, das nur eine Seite mit dem Text »Hello World« produziert.
- Das Beispiel beruht auf dem Beispiel G.2 aus dem PDF-Referenzmanual von Adobe. Im Vergleich zum Original wurden einige Teile eliminiert, die ab PDF-1.4 obsolet sind.
- Das Referenzmanual ist unter http://www.adobe.com/devnet/pdf/pdf_reference.html zu finden.

HelloWorld.pdf

```
5 0 obj
  << /Type /Font
    /Subtype /Type1
    /Name /F1
    /BaseFont /Helvetica
    /Encoding /MacRomanEncoding
  >>
endobj
```

- Mit obj und endobj wird die Definition eines PDF-Objekts geklammert.
- Vor obj stehen die Objekt- und die Versionsnummer. Erstere ist typischerweise fortlaufend, letztere im Normalfall 0. (Nur bei nachträglich veränderten PDF-Dateien entstehen Objekte mit höheren Versionsnummern.)
- Dieses PDF-Objekt definiert eine Schriftschnitt-Ressource, die bei einzelnen Seiten genutzt werden kann.

HelloWorld.pdf

```
3 0 obj
  << /Type /Page
    /Parent 2 0 R
    /MediaBox [0 0 612 792]
    /Contents 4 0 R
    /Resources << /Font << /F1 5 0 R >> >>
  >>
endobj
```

- Dieses PDF-Objekt ist die Datenstruktur für eine Seite.
- Seiten sind in PDF in einem Baum hierarchisch geordnet.
- Diese Hierarchie hat nichts mit Kapiteln oder Abschnitten zu tun, sondern dient nur der Vererbung von Einstellungen. Normalerweise ist dieser Baum ziemlich flach mit einer Wurzel, die alle einzelnen Seiten als Kinder hat.

HelloWorld.pdf

```
3 0 obj
  << /Type /Page
    /Parent 2 0 R
    /MediaBox [0 0 612 792]
    /Contents 4 0 R
    /Resources << /Font << /F1 5 0 R >> >>
  >>
endobj
```

- 2 0 R referenziert das PDF-Objekt, das mit der identifizierenden Sequenz 2 0 obj definiert ist.
- Das Feld MediaBox spezifiziert die Dimension der Seite (hier US-Letter-Format), Contents referenziert das PDF-Objekt mit der Seitenbeschreibung und Resources zählt die für die Seitengenerierung benötigten Ressourcen auf. Hier wird nur der zuvor deklarierte Schriftschnitt unter dem Namen F1 eingebunden.

HelloWorld.pdf

```
4 0 obj
  << /Length 68 >>
stream
  BT
    /F1 24 Tf
    100 100 Td
    (Hello World) Tj
  ET
endstream
endobj
```

- Eine Seitenbeschreibung besteht aus einem Stream-Objekt, das aus einem assoziativen Array, das mindestens die Länge in Bytes spezifizieren muss, und dem eigentlichen Stream-Inhalt besteht.
- Der Stream-Inhalt wird von den Schlüsselworten `stream` und `endstream` eingegrenzt.

HelloWorld.pdf

```
BT
  /F1 24 Tf
  100 100 Td
  (Hello World) Tj
ET
```

- Mit BT und ET wird ein Textobjekt geklammert. Textobjekte können nicht verschachtelt werden und verwalten einen lokalen Zustand mit beispielsweise dem aktuellen Schriftschnitt.
- Der Operator Tf entspricht dem `selectfont`-Operator in PostScript und wählt eine Schrift aus. Die Schrift muss in den zur Seite gehörenden Ressourcen explizit registriert sein.
- Der Operator Td entspricht in etwa dem `moveto`, bezieht sich aber nur auf die Positionierung von Texten.
- Tj entspricht dem `show`-Operator in PostScript. Anders als in PostScript berücksichtigt dieser aber mehr Statusparameter.

HelloWorld.pdf

```
2 0 obj
  << /Type /Pages
    /Kids [3 0 R]
    /Count 1
  >>
endobj
```

- Dieses PDF-Objekt ist die Wurzel der Seitenhierarchie.
- Unter `Kids` werden die einzelnen Seiten über entsprechende Referenzen (wie etwa hier `3 0 R`) aufgezählt.
- `Count` spezifiziert die Zahl der Blattknoten, die direkt oder indirekt über diesen Baum erreichbar sind.

HelloWorld.pdf

```
1 0 obj
  << /Type /Catalog
    /Pages 2 0 R
  >>
endobj
```

- Der Katalog verweist auf alle globalen Datenstrukturen des PDF-Dokuments.
- In diesem einfachen Beispiel findet sich nur der Verweis auf die Wurzel der Seitenhierarchie.
- Zusätzlich denkbar wären Navigationshierarchien, Hyperlinks, Benennungsregeln für die einzelnen Seiten, Voreinstellungen für Betrachter etc.

HelloWorld.pdf

```
xref
0 6
0000000000 65535 f
0000000009 00000 n
0000000071 00000 n
0000000147 00000 n
0000000304 00000 n
0000000425 00000 n
```

- Der Index beginnt mit dem Schlüsselwort `xref`, gefolgt von einer oder mehreren zusammenhängenden Bereichen.
- Jeder Bereich beginnt mit zwei ganzen Zahlen, die die niedrigste Objektzahl und die Zahl der Einträge spezifizieren. Dieser Bereich nennt dann fortlaufend die Byte-Positionen der entsprechenden PDF-Objekte. In diesem Beispiel beginnt es mit der Nummer 0 und es sind nur 6 PDF-Objekte deklariert.

HelloWorld.pdf

```
xref
0 6
0000000000 65535 f
0000000009 00000 n
0000000071 00000 n
0000000147 00000 n
0000000304 00000 n
0000000425 00000 n
```

- Der Eintrag zu einem PDF-Objekt besteht aus genau 20 Bytes, bestehend aus
 - ▶ einer 10-stelligen Byte-Position (mit führenden Nullen), einem Leerzeichen,
 - ▶ einer fünf-stelligen Versionsnummer, einem weiteren Leerzeichen,
 - ▶ dem Buchstaben `n` (belegt) oder `f` (frei) und
 - ▶ einem Zeilentrenner aus zwei Zeichen (CR LF).

HelloWorld.pdf

```
trailer
  << /Size 6
    /Root 1 0 R
  >>
startxref
564
%%EOF
```

- PDF-Dateien werden von hinten beginnend gelesen.
- Dort ist ganz am Ende bei `startxref` die Byte-Position des Index angegeben.
- Unmittelbar davor findet sich die `trailer`-Spezifikation, die die Zahl der PDF-Objekte insgesamt angeben und auf das Wurzel-Objekt (den Katalog) verweisen muss.