

# **Grundlagen Druckvorstufentechnik**

Skript zur Vorlesung BP103-DDV (DVT)

Fakultät 05, Druck- und Medientechnik,  
Hochschule München, Wintersemester 2017/2018

Langfassung einschließlich der Unterkapitel 7.2 bis 7.4 und 8.2 bis 8.4,  
die nicht zum Prüfungsstoff gehören



# Vorwort

Dies Skript steht kostenlos zur Vergütung. Es kann gedruckt und in digitaler Form weitergegeben werden. Auf dem Laufwerk „Unterricht“ (**Unterricht\DMT\Scharmer\Skripte**) liegt eine Kompakt- und eine Langfassung als PDF. Die Langfassung enthält zusätzlich die Unterkapitel 7.2–7.4 (SVG) und 8.2–8.4 (PostScript), die nicht zum Prüfungsstoff gehören.

Alle nicht mit einem Copyright-Hinweis versehenen Abbildungen sind von mir erstellt (teilweise unter Nutzung von Vorlagen auf Wikipedia) und dürfen gemäß WikiMedia Commons (<http://commons.wikimedia.org>) frei genutzt werden.

Bei vielen Begriffen werden die englischen Entsprechungen genannt.

Ich danke Prof. Dr.-Ing. Klaus Kreulich für die Überlassung seiner Vorlesungsunterlagen, die die Basis für dieses Skript bilden. Darüber hinaus habe ich folgende Quellen verwendet:

- ▶ Druck- und Medientechnik allgemein:
  - ▷ Helmut Kipphan: Handbuch der Printmedien. Springer, 1. Auflage (2000)  
Die englische Ausgabe des Buchs ist als PDF frei verfügbar unter:  
<http://ecolor.ir/books/Print%20Media.pdf>
  - ▷ Helmut Teschner: Druck- & Medientechnik. Christiani, 13. Auflage (2010)  
(14. Auflage 2017 erschienen)
- ▶ Grundlagen Druckvorstufentechnik:
  - ▷ Hans-Peter Schneeberger: PDF in der Druckvorstufe. Galileo, 1. Auflage (2008)  
(neue Auflage im September 2014 erschienen)
  - ▷ Thomas Merz: Die PostScript & PDF-Bibel. Dpunkt Verlag, 2. Auflage, (2001)
  - ▷ Adobe: PostScript Language Reference, 3rd Edition, 1999:  
[www.adobe.com/products/postscript/pdfs/PLRM.pdf](http://www.adobe.com/products/postscript/pdfs/PLRM.pdf)
  - ▷ Thomas Hoffmann-Walbeck: Digitale Druckformherstellung. Dpunkt Verlag,  
1. Auflage (2004)
  - ▷ Einführung in die Rastertechnologie: [www.heidelberg.com](http://www.heidelberg.com) > Downloads
  - ▷ Reiner Malaka, Andreas Butz, Heinrich Hußmann: Medieninformatik.  
Pearson Studium IT, Addison-Wesley Verlag, 1. Auflage (2009)

Das Skript deckt den prüfungsrelevanten Stoff komplett ab, geht aber teilweise darüber hinaus. Stoff, der eindeutig nicht prüfungsrelevant ist, steht zwischen folgenden Tags:

<Ø>

Nicht prüfungsrelevanter Stoff

</Ø>

---

Am Ende der meisten Kapitel wird das prüfungsrelevante Kernwissen stichpunktartig zusammengefasst:

<!> **Zusammenfassung**

Für die Praxis wichtige „Merksätze“ sind folgendermaßen gekennzeichnet:

- ⌘ Bei der Beurteilung von Druckvorlagen oder Druckergebnissen nie auf den optischen Eindruck verlassen!

Für Hinweise auf Tipp- und sonstige Fehler sowie Vorschläge zur Verbesserung/Erweiterung bin ich jederzeit dankbar und am besten unter **scharmer@hm.edu** erreichbar. An diese E-Mail-Adresse können auch alle anderen Fragen zur Vorlesung und zur Prüfung an mich gerichtet werden. Ich antworte kurzfristig.

München, März 2015 und Oktober 2017  
Ulrich Scharmer

# Inhalt

|                                                                                                                    | Seite |
|--------------------------------------------------------------------------------------------------------------------|-------|
| 1 Einführung: Der Workflow von der Idee zum (Print-)Produkt; Zeichencodierung . . .                                | 3     |
| 2 Lichttechnische Grundlagen (Strahlungsleistung, Helligkeit; Fotometrie) . . . . .                                | 12    |
| 3 Lichttechnik in der Fotografie (Blende, Belichtung, Schärfentiefe usw.) . . . . .                                | 23    |
| 4 Digitalisieren von Bildern (Scanner, Kodierung als Bytemap) . . . . .                                            | 43    |
| 5 Bildbearbeitung für die Druckvorstufe (Gradation, Tonwertkurve, Histogramm, Farbstichkorrekturen usw.) . . . . . | 53    |
| 6 Grafikformate (Bitmap, Bytemap, Vektor) . . . . .                                                                | 68    |
| 7 Scalable Vector Graphics (SVG) . . . . .                                                                         | 79    |
| 8 PostScript . . . . .                                                                                             | 92    |
| 9 Portable Document Format (PDF) . . . . .                                                                         | 105   |
| 10 Rastertechnik . . . . .                                                                                         | 118   |



# 1 Einführung: Der Workflow von der Idee zum (Print-)Produkt; Zeichencodierung

## Übersicht

- 1.1 Einordnung der Druckvorstufe im Medienproduktionsprozess
- 1.2 Produktionsschritte der Druckvorstufe
- 1.3 Zeichencodierung

### 1.1.1 Themen der Druckvorstufe/Druckvorstufentechnik

Abbildung 1.1.1.1 zeigt die technischen Themen (blau unterlegt) der Vorstufentechnik im Studiengang Druck- und Medientechnik an der HM (Stand: Studienordnung von 2016).

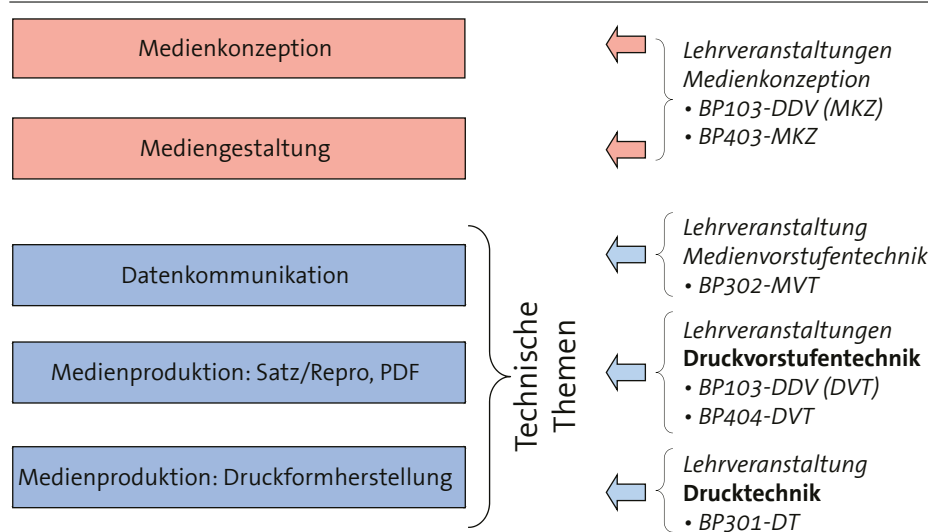
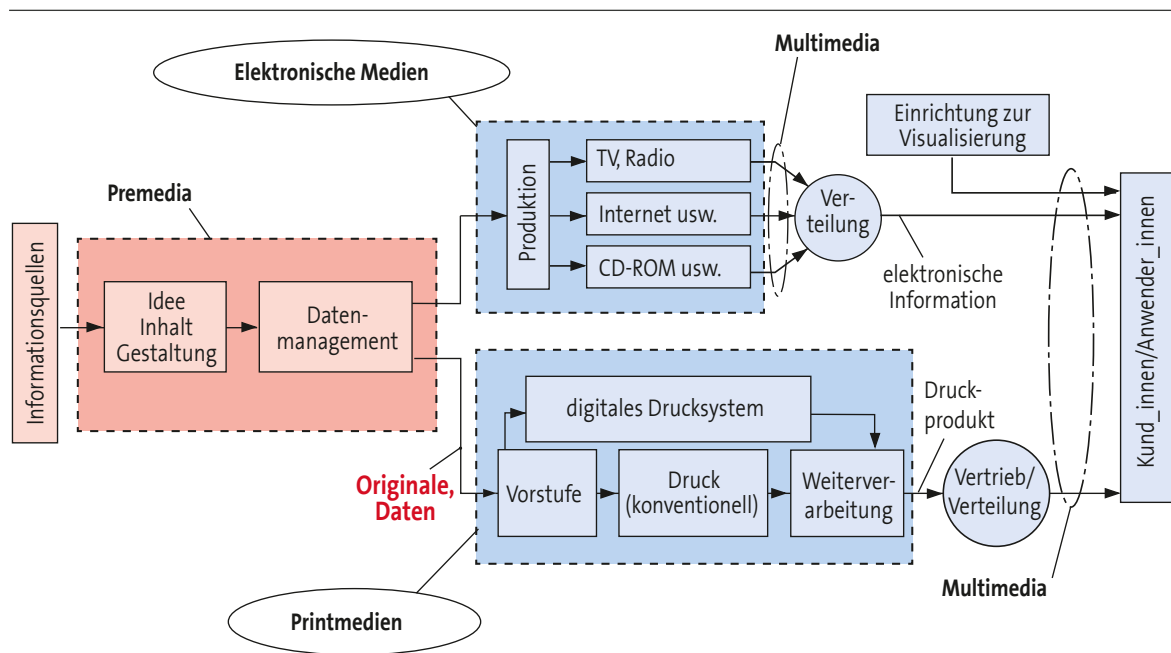


Abb. 1.1.1.1 | Stellung der Druckvorstufentechnik im Studium.

### 1.1.2 Erweiterter Produktionsprozess für Print- und elektronische Medien

Abbildung 1.1.1.2 (► Seite 4) zeigt den Prozess von der Idee bis zum fertigen Produkt. Rot unterlegt sind die Schritte, die dem Druck und der Erstellung elektronischer Medien vorausgehen („Premedia“). Die Nahtstelle ist mit „Originale, Daten“ gekennzeichnet. Dort beginnt im unteren Teil der Abbildung die Druckvorstufe (gefolgt vom Druck), im oberen Teil die Erstellung elektronischer Medien (beides blau unterlegt).



**Abb. 1.1.2.1 | Produktionsprozesse bei der Herstellung von elektronischen Medien, Printmedien und Multimedia-dokumenten; die Schritte vor der Druckvorstufe („Premedia“) sind rot unterlegt.**  
Quelle: Kipphan, Handbuch der Printmedientechnik

Die Grenze zwischen „Printmedien“ und „elektronischen Medien“ nach der Schnittstelle „Originale, Daten“ (in der Abbildung in roter Schrift) ist durchlässig geworden und wird sich weiter verwischen. So sollten Teilprodukte der Druckvorstufe, z. B. fertig layoutete Seiten einer Zeitschrift und das Bildmaterial dazu, direkt in elektronische Medien einfließen können, etwa in das Content-Management-System (CMS) für eine Website oder die Produktion eines e-Book (e-Publishing, heute wichtiger als die CD-ROM).

Diese Aufhebung der Grenzen betrifft auch schon den Bereich „Premedia“. Datenbanken sollten z. B. so angelegt sein, dass die XML-Daten direkt für die Erstellung von Printkatalogen (Database Publishing) und Web-Shops genutzt werden können.

<Ø> Ein weiteres Stichwort ist medienneutrale Datenhaltung (↻ Seite 54). Es wird teilweise empfohlen, Bilddaten z. B. nicht in einem CMYK-Farbraum (d. h. farbsepariert) zu halten, sondern in einem möglichst großen RGB-Farbraum (z. B. ECI-RGBv2, siehe [www.eci.org](http://www.eci.org)). Werden die Bilddaten für einen bestimmten Druckprozess separiert, engt das die spätere Verwendung (z. B. anderes Druckverfahren mit größerem Farbraum) ein, weil der Farbraumverlust bei der Separation der Bilddaten irreversibel ist.

</Ø>

### 1.1.3 Produktionsprozess für Printmedien

Abbildung 1.1.3.1 zoomt auf den Bereich Print. Schnittstelle zu „Premedia“ ist wieder die Übergabe „Originale, Daten“. Auch hier gilt das in 1.1.2 zur Durchlässigkeit zwischen den Bereichen Print- und elektronische Medien Gesagte.

Letzter Schritt der Druckvorstufe ist im Allgemeinen die Film- bzw. Plattenbelichtung (Computer to Plate, CTP, oft auch als CtP abgekürzt) oder die Übergabe einer für den Druck optimierten PDF-Datei.

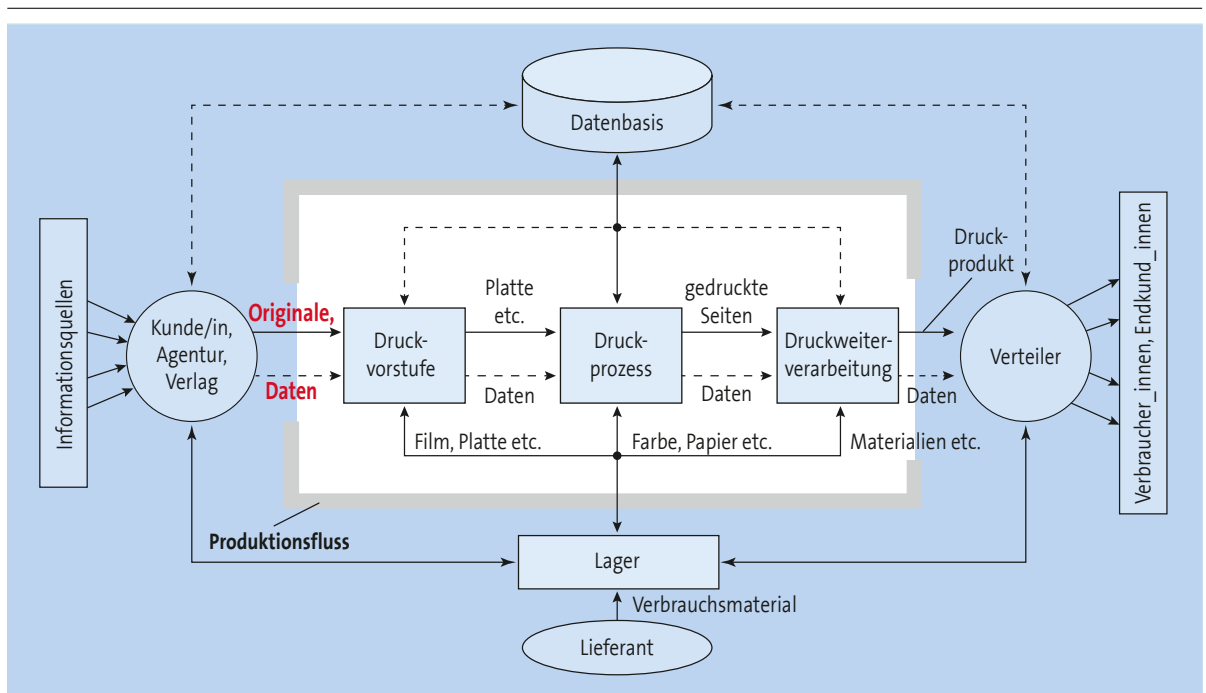


Abb. 1.1.3.1 | Produktionsprozess bei Printmedien.  
Quelle: Kipphan, Handbuch der Printmedientechnik

## 1.2 Produktionsschritte der Druckvorstufe

### 1.2.1 Die klassischen Prozesse der Druckvorstufe

Die Abbildung 1.2.1.1 zeigt die „klassischen“ Prozesse der Druckvorstufe, die Mitte der 1960er-Jahre mit den sog. Lichtsatztechniken (im Gegensatz zum Hand- oder Maschinensatz mit Bleiletttern) eingeführt worden waren und bis zum Ende der 1980er-Jahre Standard waren. Text, z. B. für eine Zeitschriftenseite, wurde als endloser Spaltentext belichtet. Bild- und Grafikelemente wurden separat reproduziert und belichtet. Erst bei der Seitenmontage auf einem Standbogen wurden Text, Bild und Grafik zusammengeführt. Von dem geklebten Bogen wurde in der Regel eine Kontaktkopie gezogen (sog. Glattfilm). Davon wurden schließlich die Druckformen erstellt.

Die klassischen Prozesse der Druckvorstufe

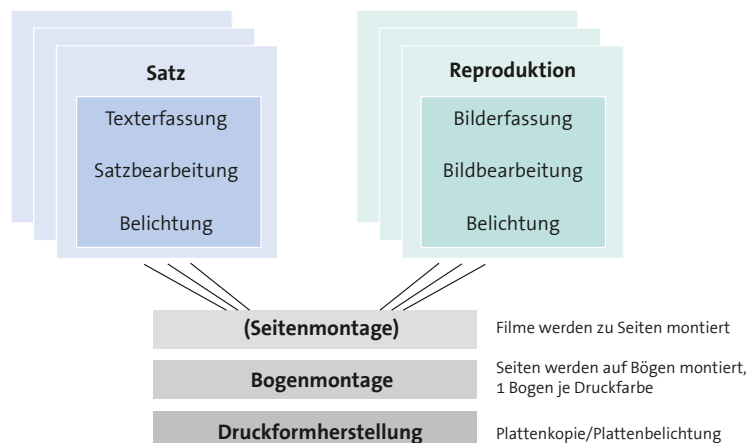


Abb. 1.2.1.1 | „Klassischer“ Ablauf der Druckvorstufe (heute kaum noch zu finden).

## 1.2.2 Digitaler Druckvorstufen-Workflow

Der Workflow von der Idee bis zur Druckform lässt sich in zwei Abschnitte gliedern:

- ▶ von der Informationsidee zum digitalen Datenbestand („Premedia“ in Abb. 1.1.2.1),
- ▶ vom digitalen Datenbestand zur Druckform (Druckvorstufe im engeren Sinn).

Abbildung 1.2.2.1 zeigt den Abschnitt „Premedia“, Abbildung 1.2.2.2 die Druckvorstufe im engeren Sinn.

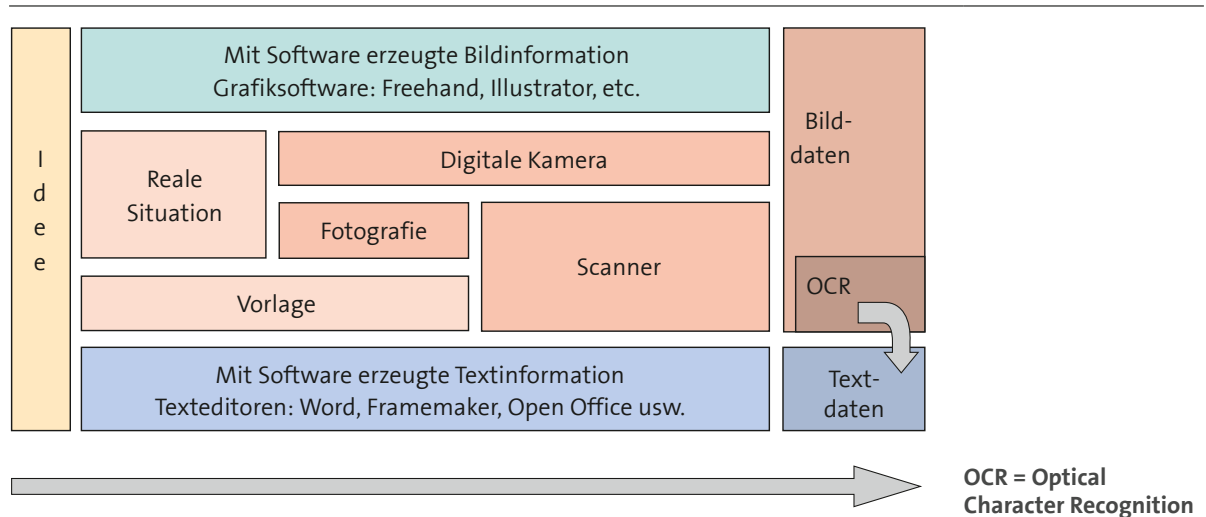


Abb. 1.2.2.1 | Teil 1: von der Idee zum digitalen Datenbestand („Premedia“).

Die Druckvorstufe im engeren Sinn endet mit der Herstellung der Druckformen. Hierzu kann entweder erst ein Film belichtet werden, von dem eine Platte belichtet wird. Die Schritte bis auf Filmbelichtung sind in Abbildung 1.2.2.2 dargestellt.

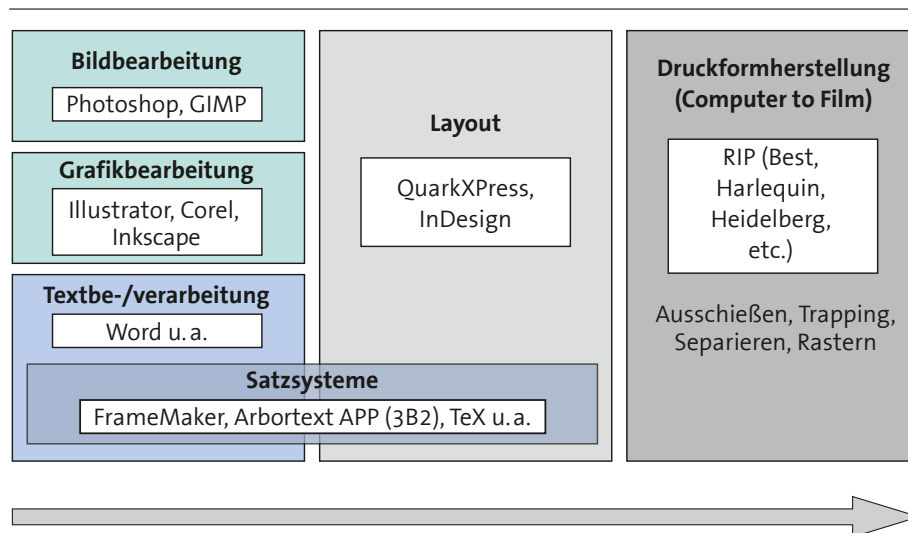
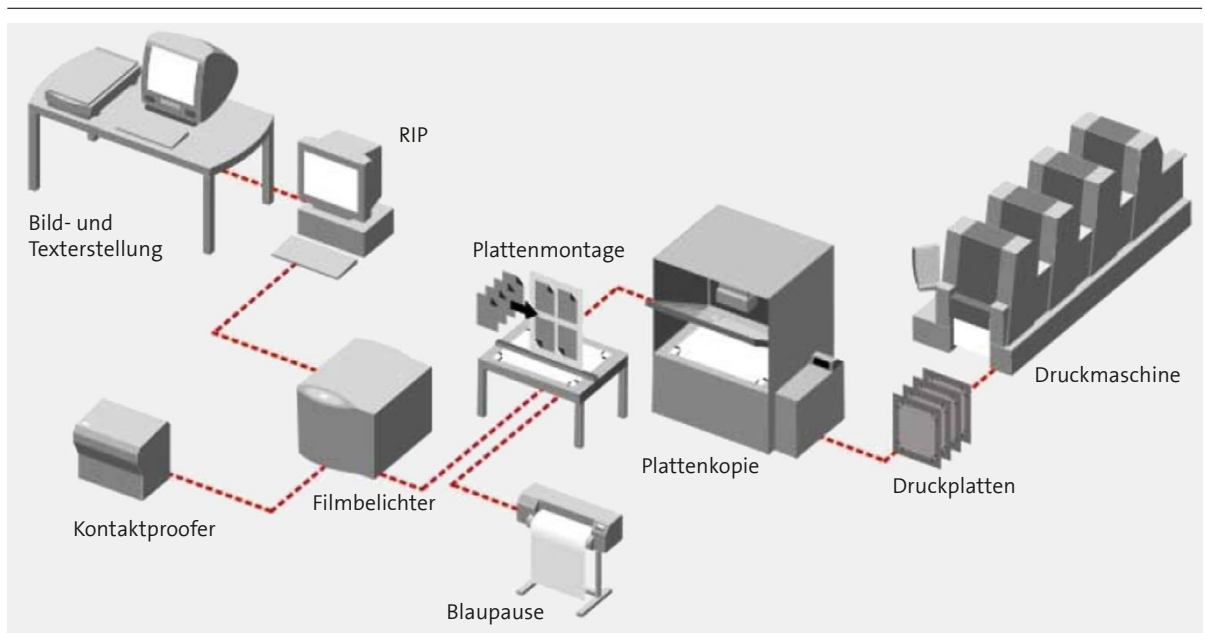


Abb. 1.2.2.2 | Teil 2: vom digitalen Datenbestand zur Druckform (hier: bis zur Filmbelichtung, vgl. Abb. 1.2.2.3).

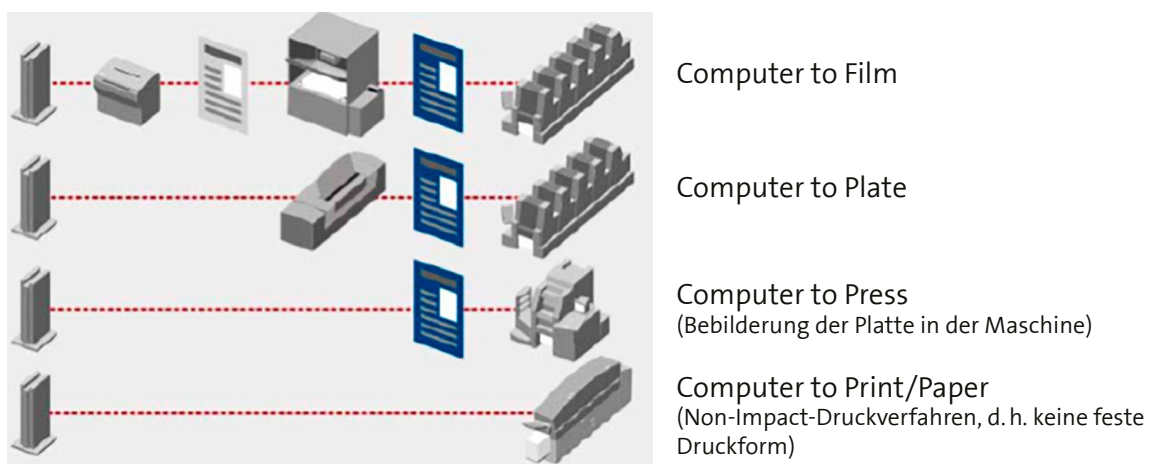
Schließt der Workflow die Belichtung von Filmen als Vorlage für die Druckplattenherstellung ein, wird dies als Computer to Film bezeichnet. Abbildung 1.2.2.3 zeigt solch einen Workflow.



**Abb. 1.2.2.3 | Produktionsprozess bei Printmedien (hier: Druckplattenherstellung über den Zwischenschritt der Filmbelichtung).**  
Quelle: Heidelberger Druckmaschinen AG

Werden die Druckplatten ohne den Zwischenschritt über Film direkt belichtet, wird dies als Computer to Plate (CTP) bezeichnet. CTP verbessert das Druckergebnis insbesondere im Bereich sehr niedriger und sehr hoher Tonwertstufen. Während etwa bei Film Tonwerte unter etwa 1 % gekappt werden (d. h. nicht gedruckt), sind bei CTP auch Tonwerte unter 1 % umsetzbar. Das erhöht zugleich die Anforderungen an die Bildbearbeitung.

Abbildung 1.2.2.4 gibt einen Überblick über die Computer-to-...-Technologien



**Abb. 1.2.2.4 | Die verschiedenen Computer-to-...-Technologien.**  
Quelle: Heidelberger Druckmaschinen AG

## 1.3 Zeichenkodierung

In diesem Kapitel wird kurz auf die Zeichenkodierung für Texte eingegangen. Stellvertretend werden erwähnt:

- ASCII (American Standard Code for Information Interchange)
- ISO 8859-X (mit X von 1 bis 16)
- UTF-8 (8-Bit Unicode Transformation Format)

### Vorbemerkung

Grundlagen von Zeichensatzsystem sind das Dual-, Dezimal- und Hexadezimalsystem. Zeichensatzsysteme ordnen jedem Zeichen einen eindeutigen Wert zu, den *Code Point*. In Unicode-Tabellen (siehe unten) werden Code Points in der Regel als Hexadezimalwert gelistet (z. B. U+0041 für den Buchstaben „A“). Die Berechnung des UTF-8-Codes aus einem bestimmten Unicode Code Point (siehe Seite 9) benötigt als Zwischenschritt die Umwandlung in das Dualsystem.

#### 1.3.1 ASCII

Dies ist das älteste Kodierungssystem. Es beschränkt sich auf die im Englischen vorkommenden Groß- und Kleinbuchstaben, Ziffern sowie einigen Interpunktions- und Sonderzeichen sowie einigen Steuerzeichen. Es ist auf 128 Zeichen limitiert (0 bis 127) und kann daher mit einem 7-Bit-Code dargestellt werden. In 8-Bit-Systemen wird eine führende Null ergänzt.

#### 1.3.2 ISO 8859-X

Der nächste Schritt war die Weiterentwicklung zu 8-Bit-Codes, mit denen sich 256 Zeichen darstellen lassen. 256 Zeichen reichen aber bei Weitem nicht, um alle auf lateinischen Buchstaben basierenden Sprachen komplett abzubilden. Von ISO 8859 gibt es daher 15 Varianten. Die Zahl wird mit einem Bindestrich an den Namen ISO 8859 angehängt:

| -1                         | -2                           | -3                        | -4                         | -5         | -6       | -7         | -8        | -9                | -10               | -11  | -12               | -13               | -14               | -15                          | -16                           |
|----------------------------|------------------------------|---------------------------|----------------------------|------------|----------|------------|-----------|-------------------|-------------------|------|-------------------|-------------------|-------------------|------------------------------|-------------------------------|
| Latin-1,<br>Westeuropäisch | Latin-2,<br>Mitteleuropäisch | Latin-3,<br>Südeuropäisch | Latin-4,<br>Nordeuropäisch | Kyrillisch | Arabisch | Griechisch | Hebräisch | Latin-5, Türkisch | Latin-6, Nordisch | Thai | (existiert nicht) | Latin-7, Baltisch | Latin-8, Keltisch | Latin-9, Westeuro-<br>päisch | Latin-10,<br>Südosteuropäisch |

#### 1.3.3 Unicode

Die (noch laufende) Entwicklung von Unicode (= *Unicersal Character Set, UCS*) hat das Ziel, sämtliche Schriftzeichen, d. h. auch z. B. chinesische, japanische und koreanische (CJK) sowie Symbole, musikalische Notationszeichen, Dominosteine, Emojis usw. zu kodieren.

Unicode hat 17 Ebenen. Pro Ebene (*Plane*) steht der Adressraum von 0000 bis FFFF (Hexadezimalcode,  $16^4 = 65\,536$  Zeichen) zur Verfügung. 17 Ebenen (nummeriert von 0 bis 16) erlauben daher insgesamt 1 114 112 Zeichen. Korrekt wird der Adressraum für Ebene 0 mit 0000 0000 – 0000 FFFF angegeben. Für Ebene 16 lautet er folglich 0010 0000 – 0010 FFFF.

Ebene 0 ist die *Basic Multilingual Plane* (BMP), in der alle geläufigen Zeichen der größeren Sprachen der Welt zu finden sind. Die anderen 16 Ebenen sind *Supplementary Planes*. Ebene 1 ist die *Supplementary Multilingual Plane* (SMP), die u. a. mathematische Zeichen, Ornamente, Spielkarten, Dominosteine und Emojis enthält. In der aktuellen Version 10.0 von Unicode sind knapp 140 000 Zeichen zugewiesen.

Unicode-Zeichen werden nach folgendem Prinzip benannt:

- U+XXXX für Zeichen der *Ebene 0* (BMP) (z. B. das 2017 in den Duden aufgenommene versale ß: U+1E9E); XXXX ist der Code Point in Hexadezimalschreibweise.
- U+1XXXX für Zeichen der *Ebene 1* (z. B. der Violinschlüssel : U+1D11E)
- U+2XXXX für Zeichen der *Ebene 2* usw.

## Von Unicode zu UTF-8

Man liest gelegentlich, UTF-8 und Unicode seien „weitgehend identisch“. Das ist insofern Unsinn, als Unicode und UTF-8 unterschiedlichen Begriffsebenen angehören: Unicode ist ein *Zeichensatz*, UTF-8 dagegen eines (von mehreren) *Transformationsformaten* für die Zeichen in Unicode.

Für Sprachen mit lateinischen Buchstaben ist UTF-8 (exakte Bezeichnung: *8-Bit Unicode Transformation Format*) am weitesten verbreitet. Weitere Transformationsformate sind u. a. UTF-16 (16 Bit) und UTF-32 (32 Bit). Alle UTF-Varianten können alle Unicode-Zeichen aller Ebenen darstellen, allerdings mit unterschiedlichem Speicherbedarf.

UTF-8 kann aus Folgen von 1 bis 4 Byte bestehen. Vorteile: Es ist abwärtskompatibel zu ASCII und benötigt kein Zeichen, das die Byte-Reihenfolge festlegt (Byte Order Mark, BOM). Ein Nachteil ist, dass Texte in nichtlateinischen Sprachen in UTF-8 mehr Speicherplatz einnehmen als in UTF-16. Der Grund liegt darin, dass bei UTF-8 von 2 Byte (zusammen 16 Bit) nur 11 Bit (= 2048 Zeichen) frei verfügbar sind (Erklärung weiter unten), während bei UTF-16 alle 65 536 Zeichen der Ebene 0 (BMP) mit nur *einer* 16-Bit-Code-Unit beschreibbar sind. Die zweite 16-Bit-Code-Unit wird nur für Zeichen aus den anderen 16 Unicode-Ebenen benötigt.

Für reinen ASCII kommt UTF-8 mit 1 Byte aus. Für die häufigsten Zeichen vieler Sprachen genügen 2 Byte. Nichtlateinische Sprachen benötigen jedoch häufig 3 Byte (siehe oben).

Warum sind überhaupt *Transformationsformate* wie UTF-8 nötig? Der Grund: Um eine variable Anzahl von *Code Units* (bei UTF-8 die erwähnten 1 bis 4 Byte) ohne Begrenzungszeichen zu ermöglichen, muss das erste Byte eines Zeichens zusätzlich zum Zeichencode Information darüber enthalten, aus wie vielen Byte das Zeichen besteht.

<Ø> Das nachfolgend beschriebene Transformationsprinzip für UTF-8 ist **nicht prüfungsrelevant!**

Um einen Unicode *Code Point* in UTF-8 zu transformieren, benötigt man zunächst den Binärcode des Zeichens. Je nachdem, aus wie vielen Bit er besteht (eventuelle führende Nullen aus Umrechnungshilfsprogrammen werden nicht mitgezählt) werden die Einsen und Nullen (in der Tabelle grün und überstrichen) nach einem festen Schema in eine Tabelle eingefügt. Diese Tabelle enthält je nach Anzahl der Byte für das Zeichens unterschiedliche feste Einträge (rot und unterstrichen).

|                                                                                                                                           |        |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 7 Bit (= ASCII): <u>0xxx xxxx</u>                                                                                                         | 1 Byte |
| 8 bis 11 Bit ( $2^{11} = 2048$ Zeichen): <u>110x xxxx</u> <u>10xx xxxx</u>                                                                | 2 Byte |
| 12 bis 16 Bit ( $2^{16} = 65536$ Zeichen): <u>1110 xxxx</u> <u>10xx xxxx</u> <u>10xx xxxx</u>                                             | 3 Byte |
| 17 bis 21 Bit ( $2^{21} = \text{theoretisch } 2.097.152$ Zeichen):<br><u>1111 0xxx</u> <u>10xx xxxx</u> <u>10xx xxxx</u> <u>10xx xxxx</u> | 4 Byte |

- Beginnt das erste Byte mit einer 0, bedeutet das 7 Bit ASCII und legt fest, dass das ganze Zeichen nur aus einem Byte besteht.
- Beginnt das erste Byte mit 110, bedeutet das, dass das Zeichen aus 2 Byte besteht. Das Folge-Byte beginnt mit 10 und kann daher weder mit 7-Bit-ASCII noch mit einem anderen Start-Byte verwechselt werden.
- Beginnt das erste Byte mit 1110, kündigt es 3 Byte an,
- beginnt es mit 11110, 4 Byte.



## Beispiele für die Transformation von Code Points aus Unicode zu UTF-8:

### ■ Beispiel 1: Das Unicode-Zeichen A (Ebene 0)

Code Point: U+0041

Das Zeichen liegt im Bereich für ASCII (d. h. zwischen 0000 und 007F), also ist es ein 7-Bit-Zeichen. Es gilt die Regel für 7-Bit-Zeichen:

0 plus 7 Bit für das Zeichen ergibt 1 Byte.

0041 ist in Binärcode 1000001;

ergibt als UTF-8: 01000001 (Hex: 41)

**Beachte:** Der Hex-Wert in UTF-8 ist von 0000 bis 007F *identisch* mit dem Hex-Wert in ASCII.

### ■ Beispiel 2: Das Unicode-Zeichen Ω (Ebene 0)

Code Point: U+03A9

03A9 hat den Binärcode 1110101001 (10 Bit)

Das Zeichen hat also mehr als 7, aber nicht mehr als 11 Bit. Somit gilt die Regel für 8- bis 11-Bit-Zeichen (d. h. 2 Byte): 110x xxxx 10xx xxxx

Einfügen in das Schema ergibt: (die letzten 6 Stellen von rechts in das zweite Byte, die restlichen 4 von rechts in das erste Byte, die verbleibende freie Stelle wird mit 0 gefüllt)

UTF-8: 1100 1110 1010 1001 (Hex: CEA9)

**Beachte:** Der Hex-Wert in UTF-8 ist hier *nicht identisch* mit dem Hex-Wert des Code Point.

### ■ Beispiel 3: Violschlüssel: (Ebene 1)

Code Point: U+1D11E

1D11E entspricht den Binärcode 11101000100011110 (17 Bit)

Da das Zeichen mehr als 16 Bit hat, gilt die Regel für 17- bis 21-Bit-Zeichen (d. h. 4 Byte):

1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx

Einfügen in das Schema ergibt: (die letzten 6 Stellen von rechts in das vierte Byte, die nächsten 6 Stellen von rechts in das dritte Byte und die restlichen 5 in das zweite Byte; die verbleibende freie Stelle in Byte 2 wird mit 0 gefüllt, ebenso alle freien Stellen in Byte 1)

UTF-8: 1111 0000 1001 1101 1000 0100 1001 1110 (Hex: F09D849E)

Auch hier ist der Hex-Wert des Code Point nicht identisch mit dem Hex-Wert in UTF-8.

## Anmerkung

HTML-Seiten sollten heute mit UTF-8 als Zeichensatz angelegt werden (charset="utf-8"). Alle auf den Standardtastaturen erreichbaren Zeichen können dann direkt eingegeben werden, d. h. sie müssen nicht mühsam als *Entitäten* (z. B. `&uuml;` für „ü“) oder Sonderzeichen (z. B. `&#8364;` für „€“; kann auch als Entität kodiert werden: `&euro;`) kodiert werden. Jeder zeitgenössische Browser kann UTF-8 korrekt darstellen.

Unicode *Code Points* können in HTML und XML als Dezimal- oder Hexadezimalwert eingegeben werden. `&#7838;` steht für das versale `ß` in Dezimalcode. Will man den Code Point als Hexadezimalwert eingeben, muss nach der Raute ein **kleines x** eingefügt werden: `&#x1E9E;` (für den Hex-Code sind auch Kleinbuchstaben erlaubt, aber nicht empfohlen).



## Garbage Characters

„Zeichenmüll“ entsteht, wenn Zeichen nicht mit der beabsichtigten Kodierung gelesen werden, etwa weil die Kennzeichnung fehlt oder fehlerhaft ist.

- ▶ *Als UTF-8 geschrieben:*  
Ärger über blöden Text. €
- ▶ *Als ISO 8859-1 interpretiert:*  
Ärger Ä¼ber blÄ¼den Text. â¬

### Erklärung:

- ▷ Der Buchstabe Ä wird in UTF-8 als C384 kodiert. ISO 8859-1 liest C3 als den Buchstaben Ä, der Code 84 steht für ein Steuerzeichen, wird also nicht durch ein Textzeichen angezeigt.
- ▷ Der Buchstabe ü wird in UTF-8 als C3BC kodiert. ISO 8859-1 liest C3 als Ä und BC als ¼.
- ▷ In UTF-8 besteht das Zeichen ö aus den beiden Bytes C3 und B6, die in ISO 8859-1 den Zeichen Ä und ¶ entsprechen.
- ▷ Das Euro-Zeichen schließlich ist in UTF-8 als 3-Byte-Zeichen E2 82 AC kodiert. ISO 8859-1 liest E2 als â, 82 ist ein Steuerzeichen (bleibt stumm), und AC als das logische Negationszeichen (¬).

### Hinweis:

ISO 8859-1 ist zwar identisch mit den ersten 256 Zeichen von Unicode; die Transformation von Unicode in UTF-8 liefert aber nur für die ersten 128 Zeichen (= ASCII) unveränderte Werte. Dies erklärt, warum die Zeichen 128 bis 255 aus ISO 8859-1 bei Interpretation als UTF-8 als Garbage Characters ankommen.

Das Fragezeichen in der Raute (❖) ist in Unicode das Replacement-Zeichen (U+FFFD). Es wird von Browsern ausgegeben, wenn ein Zeichen nicht identifiziert werden kann (z. B. beim Anzeigen bestimmter Zeichen einer mit ISO 8859-1 geschriebenen Seite als UTF-8).

### ■ Beispiel:

- ▶ *Als ISO 8859-1 geschrieben (das €-Zeichen ist in ISO 8859-1 nicht enthalten)*  
Ärger über blöden Text.
- ▶ *als UTF-8 interpretiert:*  
❖ger ❖ber bl❖den Text.

### Erklärung für das erste Replacement-Zeichen:

Ä und r sind in ISO 8859-1 als C4 bzw. 72 codiert. In UTF-8 gibt es kein Zeichen mit C4 und auch keines mit C472. Der Binärcode beider Bytes ist 11000100 01110010. Das erste Byte beginnt mit 110, d.h. es legt fest, dass das Zeichen aus 2 Byte besteht. Das zweite Byte beginnt mit 01 und verstößt damit gegen die Regel, dass ein Folge-Byte mit 10 beginnt. Die beiden Byte können daher nicht interpretiert werden.

</Ø>

---

## 2 Lichttechnische Grundlagen

### Übersicht

#### 2.1 Einführung

#### 2.2 Physik und Sinneswahrnehmung Licht, Farbe, Helligkeit

#### 2.3 Fotometrische Größen Lichtstrom, Lichtstärke, Leuchtdichte, Beleuchtungsstärke, Belichtung

---

#### 2.1 Warum wird zwischen Physik und Sinneswahrnehmung unterschieden?

Licht lässt sich physikalisch charakterisieren (→ 2.2). Was der Betrachter letztlich wahrnimmt, wird aber nicht nur von den physikalischen Eigenschaften einer Lichtquelle bestimmt, sondern auch von den Prozessen der Sinneswahrnehmung. Diese sind das Resultat der optischen Eigenschaften des Auges (Hornhaut, Linse), der Interaktion des auf der Netzhaut einfallenden Lichts mit den lichtempfindlichen Molekülen (Sehpurpur) der Netzhaut und der Verarbeitung der Signale aus der Netzhaut im Gehirn. Die Prozesse der Sinneswahrnehmung können, wie die folgenden Beispiele zeigen, dazu führen, dass die Wahrnehmung stark von dem abweicht, was die physikalischen Gegebenheiten erwarten lassen.

⚡ Bei der Beurteilung von Druckvorlagen oder Druckergebnissen nie auf den optischen Eindruck verlassen!

#### Simultankontrast (= Umfeldkontrast)

Simultankontrast bedeutet, dass bei aneinandergrenzenden Flächen die Komplementärfarbe der größeren Fläche mit der Farbe der kleineren Fläche verrechnet wird. Abbildung 2.1.1 zeigt dies anhand zweier identischer grauer Ringe (= kleinere Fläche) vor hellem bzw. dunklem Hintergrund (= größere Fläche). Die Komplementärfarbe zu dem schwarzen Hintergrund im rechten Teil der Abbildung ist Weiß. Der Simultankontrast bewirkt, dass Weiß zu dem grauen Kreis „addiert“ wird, sodass der graue Kreis heller wahrgenommen wird als der identische graue Kreis vor hellgrauem Hintergrund im linken Teil der Abbildung. Im linken Teil der Abbildung ist die Komplementärfarbe zum grauen Hintergrund ein sehr dunkles Grau, das zum Grau des Rings addiert wird und diesen daher dunkler erscheinen lässt.



2 Identische graue Kreise werden vor unterschiedlichem Hintergrund verschieden hell wahrgenommen.

Abb. 2.1.1 | Simultankontrast bei Graustufenobjekten.

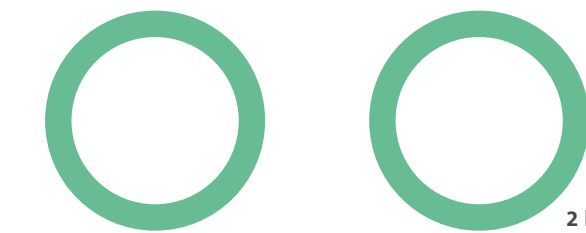
Abbildung 2.1.2 zeigt, wie ein Gradient Simultankontrast erzeugt.



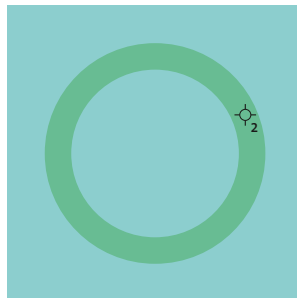
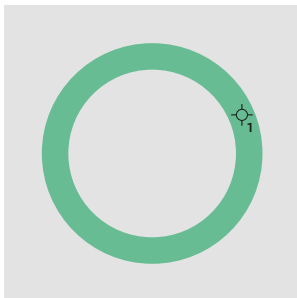
Die große Hintergrundfläche weist einen Gradienten von etwa 90 % Grau (linker Rand) bis etwa 30 % (rechter Rand) auf. Der Balken weist durchgehend denselben Grauwert von rund 65 % auf. Der vom Betrachter wahrgenommene Helligkeitsverlauf im Balken wird durch den Simultankontrast erzeugt.

Abb. 2.1.2 | Simultankontrast durch einen Gradienten (Helligkeitsverlauf).

Simultankontrast tritt auch bei farbigen Objekten auf, wie Abbildung 2.1.3 zeigt.



2 Identische grüne Kreise vor gleichem Hintergrund



Unterschiedliche Hintergründe verändern die Wahrnehmung der Kreisfarbe, obwohl die Messpipetten identische Farbwerte zeigen.

Abb. 2.1.3 | Simultankontrast bei farbigen Objekten.

## <Ø> Weitere Beispiele für optische Täuschungen

Neben dem Simultankontrast können weitere Mechanismen zu verfälschter Farb- und Helligkeitswahrnehmung führen, z. B. die Erwartungshaltung des Betrachters, die durch einen Schlagschatten hervorgerufen wird (Abb. 2.1.4).

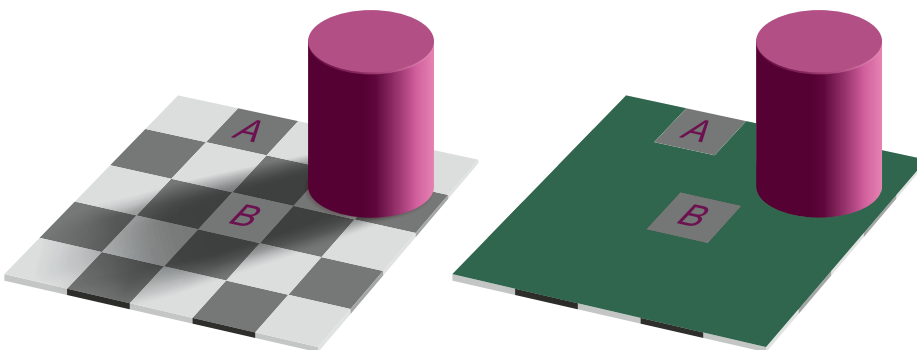


Abb. 2.1.4 | Im linken Teil der Abbildung wird die Fläche A des Schachbretts als deutlich dunkler wahrgenommen als die Fläche B, obwohl beide gleich hell sind. Maskiert man (rechter Teil der Abbildung) alle Felder des Schachbretts außer A und B mit Grün (der Schlagschatten ist noch vorhanden, wird aber vom Auge nicht mehr als solcher wahrgenommen), werden die Flächen A und B als gleich hell empfunden. Quelle: Nach Edward H. Adelson

Abbildung 2.1.5 zeigt Beispiele für optische Täuschungen, die sich nicht auf die Farb- bzw. Helligkeitswahrnehmung beschränken.

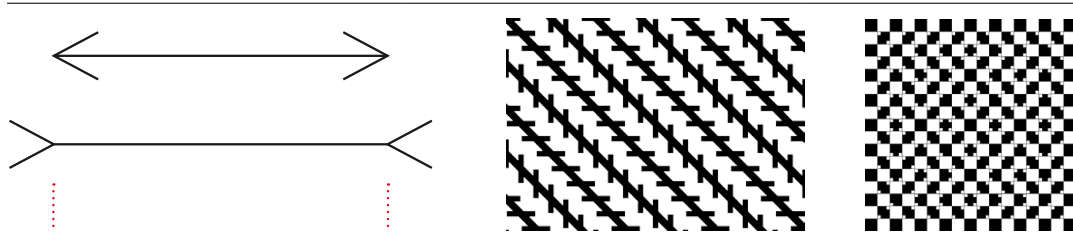


Abb. 2.1.5 | Links: Die beiden Linienabschnitte (zwischen den rot gepunkteten Linien) sind gleich lang, der untere erscheint aber aufgrund der umgedrehten Pfeilspitzen deutlich länger (Müller-Lyer-Illusion). Mitte: Die diagonalen Linien laufen parallel, obwohl die kurzen vertikalen und horizontalen Linien-segmente unterschiedliche Winkel vortäuschen (Zöllner-Illusion). Rechts: Alle Flächen sind Quadrate, die Linien erscheinen aber wellenförmig (Münsterberg-Illusion). Quelle: Wikipedia

</Ø>

## 2.2 Physik und Sinneswahrnehmung

### 2.2.1 Was ist Licht?

Zur physikalischen Beschreibung von Licht dienen 2 Modelle:

- ▶ Photonen („Teilchennatur“ des Lichts)
- ▶ Elektromagnetische Wellen („Wellennatur“ des Lichts)

⌘ Licht ist definitionsgemäß der Teil aus dem Spektrum\* der elektromagnetischen Strahlung, der für das menschliche Auge sichtbar ist. Das ist der Bereich der Wellenlängen zwischen etwa 380 und 780 nm. Manche Tiere (z. B. Bienen, Schmetterlinge) können auch Wellenlängen im angrenzenden IR- und/oder UV-Bereich sehen.

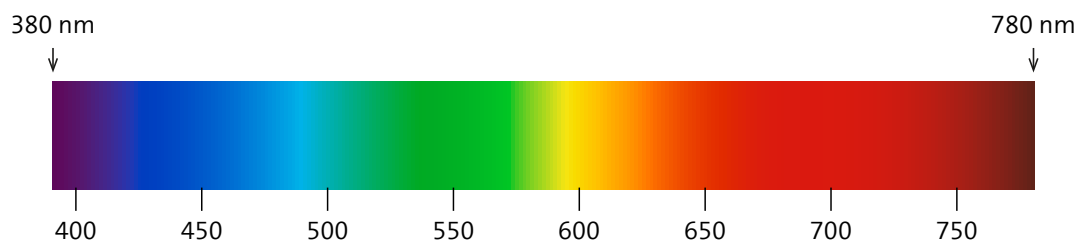


Abb. 2.2.1 | Sichtbarer Teil des elektromagnetischen Spektrums. Nach links ( $\lambda < 380$  nm) schließt sich der ultraviolette (UV) Bereich an, nach rechts ( $\lambda > 780$  nm) der infrarote (IR) Bereich.

Die Wahrnehmung von Licht umfasst die Vorgänge im Auge (Netzhaut) bis hin zur hochkomplexen Verarbeitung im Gehirn. Sie hängt somit ab von:

- ▶ der Physiologie des Auges (z.B. Sehpurpur in den Stäbchen und Zapfen der Netzhaut),
- ▶ Prozessen bei der Verarbeitung des Primärsignals auf der Netzhaut in den nachgeschalteten Neuronenverbänden und im Gehirn, bei denen u. a. auch psychologische Aspekte relevant sind (vgl. z. B. die diversen Farbenlehren).

\* Die in diesem Skript wiedergegebenen Spektren dienen generell nur zur Orientierung und können nicht als Referenz verwendet werden.

In einfacher Näherung kann der Sinneseindruck „Licht“ durch 3 Teilaspekte beschrieben werden:

- ▶ Buntton (= Farbton)
  - ▷ z. B. rot, grün, blau, violett
  - ▷ Welche Wellenlängen sind im Licht enthalten?
- ▶ Buntheit
  - ▷ Stärke des Farbtons: z. B. schwaches oder kräftiges Rot
  - ▷ Welche störenden (verschmutzenden) Wellenlängen sind im Licht enthalten?
- ▶ Helligkeit
  - ▷ Intensität des Lichts
  - ▷ Wie viele Photonen kommen pro Zeit im Auge an?

### Anmerkung und Ausblick

Das CIE-Lab-System, das die Farbwahrnehmung mithilfe von Buntton, Buntheit und Helligkeit beschreibt, sowie die Grundlagen der Farbmessstechnik in der Druckvorstufe werden in der Vorlesung Druckvorstufentechnik (4. Semester) behandelt. Im Folgenden wird nur der Teilaspekt *Helligkeit* erläutert.

## 2.2.2 Was ist Helligkeit?

Die Netzhaut des Auges registriert Licht über Zapfen und Stäbchen.

Merkhilfe: Zapfen sind für das Farbsehen bei Tag zuständig,  
die Stäbchen für das Sehen in der Dämmerung.

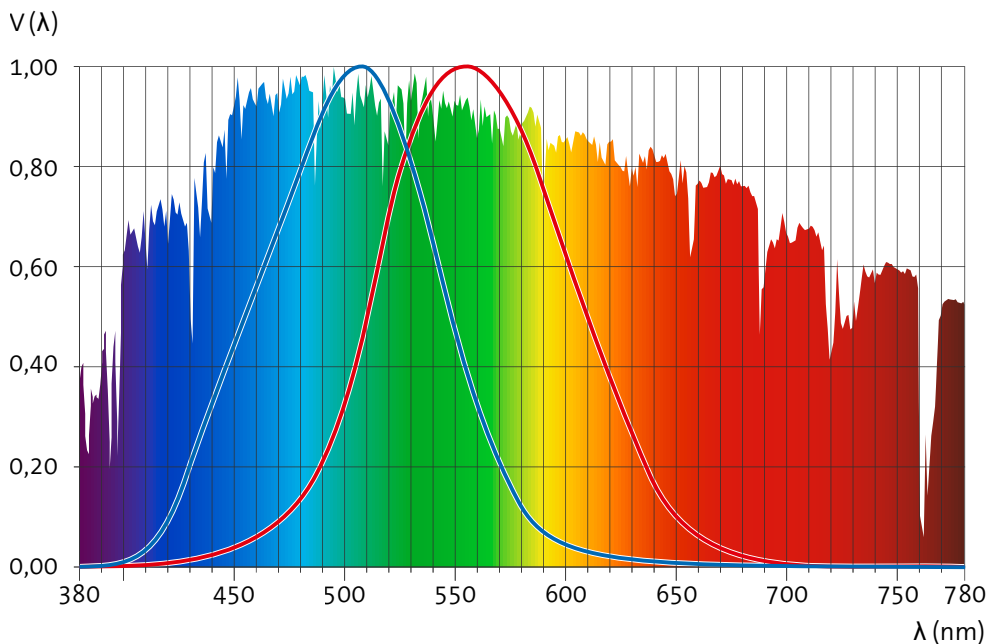
- ▶ Die 3 Typen von Zapfen (S, M und L, für short [Blau], middle [Grün] und long [Rot]) reagieren in Abhängigkeit von der Helligkeit des einfallenden Lichts und den enthaltenen Wellenlängenteilen (Farbton, Sättigung).
- ▶ Stäbchen sind nicht farbbempfindlich und nehmen nur die Helligkeit wahr.
- ▶ Zapfen sind besonders dicht an der Stelle des schärfsten Sehens (gelber Fleck auf der Netzhaut), die Stäbchen hauptsächlich in der Peripherie der Netzhaut, d. h. außerhalb des gelben Flecks. Das erklärt, warum der Mensch nachts unabhängig von der individuellen Augenometrie weniger scharf sieht.

Helligkeit ist ein Maß für die Stärke einer Lichtempfindung. Sie wird von den physikalischen Eigenschaften und physiologischen Vorgängen bestimmt:

- ▶ Bei Tag (photopisches Sehen, Farbwahrnehmung) wird die Wellenlänge 555 nm (Grün) am hellsten wahrgenommen, bei Dämmerung und Nacht (skotopisches Sehen, nur mithilfe der Stäbchen) liegt die maximale Empfindlichkeit bei 505 nm (Blau; ➤ 2.2.3).
- ▶ Simultankontrast: Licht vor dunklem Hintergrund erscheint heller als vor hellem Hintergrund (➤ 2.1).
- ▶ Helladaptation: Nach einer Einstellphase wird das hellste Licht einer Szene als Weiß wahrgenommen.
- ▶ Chromatische Adaptation (Farbkonstanz): Nach einer bestimmten Dauer der Betrachtung wird eine helle Fläche auch dann als Weiß wahrgenommen, wenn sie einen Farbstich hat. Beispiel: Ein weißes Blatt Papier erscheint sowohl bei Sonnenlicht als auch bei Kerzenlicht als weiß, obwohl im Sonnenlicht der Blauanteil viel höher ist, bei Kerzenlicht der Gelbanteil. Die chromatische Adaptation kann als eine Art Weißabgleich des Auges bezeichnet werden.

### 2.2.3 Hellempfindlichkeitskurve V des Auges

Abbildung 2.2.3.1 zeigt das Sonnenlichtspektrum auf Meeresspiegelhöhe zwischen 380 und 780 nm, d.h. im wahrnehmbaren Bereich (→ Abb. 2.2.1). Überlagert sind die *Hellempfindlichkeitskurven* V (Helladaptationskurven) für das Tag- und das Nachtsehen.



**Abb. 2.2.3.1** | Spektrum des Sonnenlichts mit überlagerten Hellempfindlichkeitskurven V des Auges bei Tagsehen (= rote Kurve) und Nachtsehen (= blaue Kurve).  
Hinweis: Unterhalb 400 nm und oberhalb 700 nm ist die Empfindlichkeit des Auges extrem gering. Für die Farbwahrnehmung spielen diese Bereiche keine Rolle.

V ist ein dimensionsloser Faktor zwischen 0 und 1 (entsprechend einer Empfindlichkeit von 0 bzw. 100 %). Das Maximum der Hellempfindlichkeitskurve liegt beim Tagsehen (photopisches Sehen) bei etwa 555 nm, beim Nachtsehen (skotopisches Sehen) bei etwa 505 nm.

Die Hellempfindlichkeitskurve besagt u.a.:

- ▶ Beim Tagsehen wird eine grüne Strahlungsquelle bei gleicher Strahlungsleistung (→ 2.3) als heller wahrgenommen als eine rote.
- ▶ Werden ein bestimmtes Grün und ein bestimmtes Blau bei Tag mit derselben Helligkeit wahrgenommen, erscheint das Blau bei Nacht heller.

## 2.3 Lichtstrom, fotometrisches Strahlungsäquivalent, Lichtstärke

Licht kann wie jede elektromagnetische Strahlung physikalisch durch verschiedene *radiometrische* Größen beschrieben werden. Wird die Hellempfindlichkeit des Auges (→ 2.2.3) als Gewichtung einbezogen, ergeben sich aus den radiometrischen Größen die zugehörigen *fotometrischen* Größen.

Die grundlegende radiometrische Größe ist die *Strahlungsleistung*.

Die zugehörige fotometrische Größe ist der *Lichtstrom*  $\Phi_v$  (Luminous Flux):

- ▶ Die Strahlungsleistung  $\Phi_e$  (Einheit: Watt, abgekürzt W) ist die Leistung (= Energie pro Zeit), die von einer elektromagnetischen Welle transportiert wird.  $\Phi_e$  wird auch als Strahlungsfluss bezeichnet. Der Index  $_e$  steht für *energetische* Messgröße.
- ▶ Den Lichtstrom  $\Phi_v$  (Einheit: Lumen, abgekürzt lm; der Index  $_v$  steht für *visuelle* Messgröße) erhält man, wenn man die Strahlungsleistung mit dem zugehörigen Wert auf der Hellempfindlichkeitskurve  $V(\lambda)$  des menschlichen Auges gewichtet. Als Faktor kommt das *fotometrische Strahlungsäquivalent*  $K(\lambda)$  hinzu:

▷ Für monochromatisches Licht gilt:

$$\Phi_v(\lambda) = K(\lambda) \cdot \Phi_e$$

bzw.

$$K(\lambda) = \frac{\Phi_v(\lambda)}{\Phi_e}$$

K hat die Einheit lm/W. Das Maximum ( $K_m$ ) liegt bei einer Wellenlänge  $\lambda$  von 555 nm.

Bei dieser Wellenlänge gilt:  $V(\lambda) = 1$  (☛ 2.2.3).

$K_m$  wird als *maximales fotometrisches Strahlungsäquivalent* bezeichnet.

Es gilt:

$$K(\lambda) = K_m \cdot V(\lambda)$$

Somit kann  $\Phi_v$  auch dargestellt werden als:

$$\Phi_v(\lambda) = K_m \cdot V(\lambda) \cdot \Phi_e$$

▷ Für Licht, das sich aus verschiedenen Wellenlängen zusammensetzt, gilt:

$$\Phi_v = \int_{380 \text{ nm}}^{780 \text{ nm}} \Phi_v(\lambda) d(\lambda)$$

⌘ Der Lichtstrom  $F$  ist ein Maß für die vom Auge wahrnehmbare Leistung einer elektromagnetischen Strahlung. Er ist keine *physikalische*, sondern eine *physiologische* Größe, die durch Gewichtung der physikalischen Größe *Strahlungsleistung* entsteht.

⌘ Hinweis: Bei Begriffen, die mit *Licht* oder *Leucht* beginnen, handelt es sich um fotometrische Größen. Beispiele: Leuchtdichte, Lichtstärke, Belichtung.

### 2.3.1 Die SI-Einheit Candela

Wie in 2.3 dargestellt, ist das fotometrische Strahlungsäquivalent der Quotient aus Strahlungsleistung  $\Phi_e$  und Lichtstrom  $\Phi_v$ .

Die Einheit Lumen (lm) für  $\Phi_v$  ist von der Einheit *Candela* (cd) abgeleitet (*Hinweis*: Betonung auf der zweiten Silbe). Candela bezeichnet die *Lichtstärke*  $I_v$  (Luminous Intensity) und ist eine der 7 Basiseinheiten des SI.

[Neben Candela sind das: Länge  $L$  in Meter (m), Masse  $M$  in Kilogramm (kg), Zeit  $T$  in Sekunden (s), Stromstärke  $I$  in Ampere (A), thermodynamische Temperatur  $T$  in Kelvin (K) und die Stoffmenge  $N$  in Mol (mol).]

Die Lichtstärke [cd] ist der Lichtstrom, der von einer (idealisierten punktförmigen) Lichtquelle pro Raumwinkel  $\Omega$  [Sterad(iant), abgekürzt sr, siehe unten]) ausgeht:  $\text{cd} = \text{lm}/\text{sr}$ .

Die Definition von Candela („Kerze“) ist historisch bedingt. Für das maximale fotometrische Strahlungsäquivalent  $K_m$  ergibt sich daraus:

Eine monochromatische Lichtquelle, die bei der Wellenlänge  $\lambda = 555 \text{ nm}$  mit einer Strahlungsleistung  $\Phi_e$  von 1 Watt strahlt, erzeugt einen Lichtstrom  $\Phi_v$  von 683 lm, d. h.:

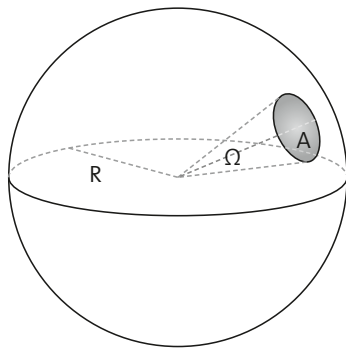
$$K_m = 683 \text{ lm/W}$$

### <Ø> Erläuterung zum Raumwinkel

Der Raumwinkel  $\Omega$  ist eine Übertragung des ebenen Winkels (2-D) auf den dreidimensionalen Raum.  $\Omega$  wird definiert über die Teilfläche  $A$ , die ein Kegel mit dem Raumwinkel  $\Omega$  aus der Oberfläche einer Kugel mit dem Radius  $R$  ausschneidet (☛ Abb. 2.3.1.1):

$$\Omega = A/R^2$$

$\Omega$  ist dimensionslos. Als Einheit wird Sterad(iant) (sr) angegeben.



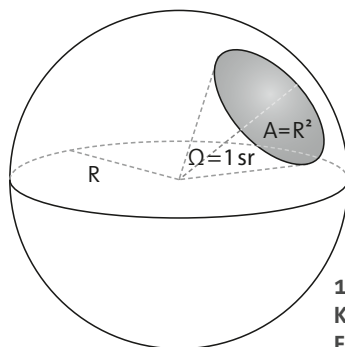
Der Raumwinkel  $\Omega$  kann definiert werden als Teilfläche  $A$  einer Kugel, dividiert durch das Quadrat des Radius  $R$  der Kugel:

$$\Omega = A/R^2$$

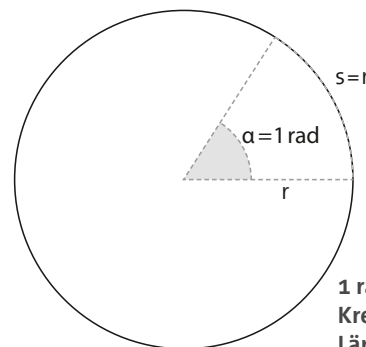
Abb. 2.3.1.1 | Definition des Raumwinkels  $\Omega$ .

Abbildung 2.3.1.2 zeigt die Definition von Steradian:  
Auf einer Kugel mit dem Radius  $R$  schneidet ein Steradian eine Kugelkalotte mit der Fläche  $A = R^2$  aus. Auf die Einheitskugel mit  $r = 1$  m übertragen hat die Kalotte die Fläche  $1 \text{ m}^2$ .

Mit der SI-Längeneinheit m für Meter gilt:  
 $1 \text{ sr} = 1 \text{ m}^2/\text{m}^2$  (Analogie: Radiant [rad] des ebenen Winkels:  $1 \text{ rad} = 1 \text{ m/m}$ )



1 sr schneidet eine Kugelkalotte der Fläche  $A = R^2$  aus.



1 rad schneidet ein Kreissegment der Länge  $s = r$  aus.

Abb. 2.3.1.2 | Definition von Steradian (links) und rad (rechts).

Der volle Raumwinkel ist gleich der Oberfläche einer Kugel geteilt durch  $R^2$ :

$$\Omega_{\text{voll}} = 4\pi R^2/R^2 = 4\pi \text{ sr} \approx 12,57 \text{ sr}$$

## Anmerkungen zum Lichtstrom

Wie entsteht Licht durch eine LED-Lampe mit 11 Watt Leistungsaufnahme?

Bei 230 V Netzspannung und 11 W Leistung fließt gemäß der Formel  $P = U \cdot I$  ein Strom von etwa 50 mA. Auch bei einer LED-Lampe wird nur ein Teil der Energie als elektromagnetische Strahlung (Strahlungsleistung  $\Phi_v$ ) abgegeben, der Rest wird in Wärme umgewandelt.

Gäbe die LED-Lampe nur Licht der Wellenlänge 555 nm ab und hätte einen Wirkungsgrad von 100 %, würde sie einen Lichtstrom von  $11 \cdot 683 \text{ lm} \approx 7500 \text{ lm}$  erzeugen. Ein Hersteller nennt für eine 11-W-Lampe (warmweiß) einen Lichtstrom von 630 lm. Der „Lichtausbeute“ beträgt somit  $630 \text{ lm}/11 \text{ W} \approx 57 \text{ lm/W}$  (8,4 % des theoretischen Maximums von 683 lm/W für monochromatisches Licht mit 555 nm). Herkömmliche Glühlampen erreichen 10–15 lm/W, Halogenlampen 15–20 lm/W und Leuchtstofflampen 70–90 lm/W.

- Die *Strahlungsleistung* der Lampe ist derjenige Anteil der Leistungsaufnahme, der in Form von elektromagnetischen Wellen abgestrahlt wird.
- Der *Lichtstrom*  $F$  der Lampe ist derjenige Teil der Strahlungsleistung, der als *sichtbares Licht* abgestrahlt wird. Die Gewichtung erfolgt mit  $K_m \cdot V(\lambda)$ .
- Eine Lampe, die ausschließlich unsichtbare UV-Strahlung erzeugt, kann zwar eine hohe Strahlungsleistung haben, erzeugt aber keinerlei Lichtstrom ( $V = 0$ )!



### Berechnungsbeispiel: Lichtstrom einer grünen Laserlichtquelle

- Angenommen, eine monochromatische Lichtquelle strahlt mit der Wellenlänge  $\lambda = 555 \text{ nm}$  eine physikalische Strahlungsleistung von 1 Watt ab. Dann gilt:

$$\Phi_e(\lambda) = 0 \text{ W} \quad \text{für } \lambda \neq 555 \text{ nm}$$

$$\Phi_e(\lambda) = 1 \text{ W} \quad \text{für } \lambda = 555 \text{ nm}$$

Mit

$$V(\lambda) = 1 \quad \text{für } \lambda = 555 \text{ nm}$$

und

$$K_m = 683 \text{ lm/W}$$

folgt für den Lichtstrom:

$$\Phi_v = K_m \cdot \int_{380 \text{ nm}}^{780 \text{ nm}} \Phi_e(\lambda) V(\lambda) d\lambda = K_m \cdot 1 \text{ W} = 683 \text{ lm/W} \cdot 1 \text{ W} = 683 \text{ lm}$$

- Wenn eine Lampe mehrere Wellenlängen abstrahlt, tragen definitionsgemäß nur die sichtbaren Anteile (d. h. zwischen 380 und 780 nm) zum Lichtstrom bei.

</Ø>

## 2.3.2 Weitere fotometrische Größen

*Lichtstärke*  $I_v$  (Luminous Intensity) = Lichtstrom pro Raumwinkel [ $\text{cd} = \text{lm/sr}$ ]  
(korrespondierende Größe aus der Radiometrie: Strahlstärke [ $\text{W/sr}$ ])

Der *Lichtstrom*  $F$  erfasst sämtliches Licht, das von einer Strahlungsquelle ausgeht, d. h. den vollen Raumwinkel. Der Lichtstrom, der in einem bestimmten Raumwinkel abgegeben wird, (Lichtstrom pro Raumwinkel, ► 2.3.1) wird als Lichtstärke bezeichnet.

Wie bei allen fotometrischen Größen wird die spektrale Wahrnehmungsfähigkeit (Hellempfindlichkeitskurve) des menschlichen Auges berücksichtigt. Unsichtbare Strahlung trägt nichts zur Lichtstärke bei. Daher ist die Lichtstärke (ebenso wie der Lichtstrom) einer reinen UV-Lichtquelle gleich null, da sie für das menschliche Auge unsichtbar ist.

- ⌘ Die Lichtstärke ist eine Eigenschaft der Lichtquelle, sie hängt nicht vom Abstand eines Beobachters ab.

Anmerkung:  $I_v$  ist kein geeignetes Maß für die Helligkeit einer Lichtquelle. Da die wahrgenommene Helligkeit einer Lichtquelle durch deren Kontrast zur Umgebung (► Simultankontrast, Seite 12) beeinflusst wird, wird z. B. eine Lichtquelle mit einer kleinen Oberfläche als heller (oder blendender) empfunden als eine Lichtquelle mit gleicher Lichtstärke und einer größeren Oberfläche.

*Leuchtdichte*  $L$  (Luminance) = Lichtstärke pro Fläche [ $\text{cd/m}^2$ ]  
(korrespondierende Größe aus der Radiometrie: Strahldichte [ $\text{W}/(\text{m}^2 \cdot \text{sr})$ ])

$L$  ist ein Maß für die Helligkeit einer Lichtquelle (z. B. Monitor).

$L$  bewertet die Lichtstärke einer Lichtquelle bezogen auf die vom Auge gesehene Fläche der Lichtquelle.

Für Flächen, die nicht senkrecht zur Blickrichtung liegen, ist der auf die senkrechte Ebene projizierte Flächenanteil (= die vom Auge gesehene Fläche) maßgebend.

Für  $L$  ist es unerheblich, ob eine Fläche selbstleuchtend oder angestrahlt ist.

Tab. 2.3.2.1 | Typische Werte für L

|                              |                                  |
|------------------------------|----------------------------------|
| Wahrnehmungsschwelle:        | $3 \cdot 10^{-6} \text{ cd/m}^2$ |
| Nachtsehen:                  | $< 10^{-4} \text{ cd/m}^2$       |
| Sonnenscheibe:               | $5 \cdot 10^9 \text{ cd/m}^2$    |
| Röhrenmonitor (CRT) Weiß:    | 80–200 $\text{cd/m}^2$           |
| Röhrenmonitor (CRT) Schwarz: | $< 0,01 \text{ cd/m}^2$          |
| Flachbildschirm (TFT) Weiß:  | 150–500 $\text{cd/m}^2$          |
| Flachbildschirm TFT Schwarz: | 0,15–0,8 $\text{cd/m}^2$         |

Anmerkung: Die Anzahl der Graustufen, die gleichzeitig in einer Szene unterschieden werden können, liegt bei 50 bis 100. Wenn das Auge z.B. auf Sonnenlicht adaptiert ist, kann zwischen Röhrenmonitor Schwarz und TFT Schwarz nicht mehr unterschieden werden.

*Beleuchtungsstärke  $E$*  (Luminous  $E$ mittance, Luminous  $E$ xitance) =  
Lichtstrom pro Empfängerfläche [ $\text{Lux} = \text{lx} = \text{lm/m}^2$ ]  
(korrespondierende Größe aus der Radiometrie: Bestrahlungsstärke [ $\text{W/m}^2$ ])

$E$  ist ein Maß für die Helligkeit, mit der ein Objekt ausgeleuchtet wird:

- ▷ Welcher Teil des Lichtstroms kommt beim Lichtempfänger an?
- ▷ Wie stark wird eine Fläche beleuchtet?
- ▷ Beispielanwendung: Ist der Schreibtisch genügend ausgeleuchtet?

Wird ein Objekt mit konstanter Lichtstärke von einer punktförmigen Lichtquelle beleuchtet, dann nimmt  $E$  mit dem Quadrat der Entfernung zur Lichtquelle ab (quadratisches Abstandsgesetz):

$E = \text{Lichtstrom}/\text{Fläche}$ ; mit  $\text{Lichtstärke} = \text{Lichtstrom}/\text{Raumwinkel}$  gilt:

$E = \text{Lichtstärke} \cdot \text{Raumwinkel}/\text{Fläche}$ ; mit  $\text{Raumwinkel} = \text{Fläche}/(\text{Radius})^2$  gilt:

$E = \text{Lichtstärke}/(\text{Abstand})^2$  [der Radius entspricht hier dem Abstand von der Lichtquelle]

Tab. 2.3.2.2 | Typische Werte für E

|                            |            |
|----------------------------|------------|
| Helles Sonnenlicht         | 100.000 lx |
| Schatten im Sommer         | 10.000 lx  |
| Operationssaal             | 10.000 lx  |
| Beleuchtung TV-Studio      | 1.000 lx   |
| Bürobeleuchtung            | 800 lx     |
| Straßenbeleuchtung         | 10 lx      |
| Kerze ca. 1 Meter entfernt | 1 lx       |
| Bewölkter Nachthimmel      | 0,0001 lx  |

*Belichtung  $H_v$*  (Exposure):

Das zeitliche Integral der Beleuchtungsstärke ergibt die Belichtung (Lichtmenge pro Fläche,  3.7, Seite 36). Da die Beleuchtungsstärke während der Belichtungsdauer im Allgemeinen konstant ist, gilt: Belichtung = Beleuchtungsstärke • Belichtungszeit [ $\text{lx} \cdot \text{s}$ ].

## Fotometrische Größen in der Druckvorstufe

| Lichtstrom F [lm]                                                                                                                                                                                                                                                                                                                                                                                                                                       | Lichtstärke I [lm/sr = cd]                                                                                                                                                                                                                                                                                                                                                                                                              | Leuchtdichte L [cd/m²]                                                                                                                                                                                                                     | Beleuchtungsstärke E [lx = lm/m²]                                                                                                                                                                                                                                                                        | Belichtung H [lx·s]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div data-bbox="539 1899 679 2033"></div> <p>Ideale punktförmige Lichtquelle strahlt gleichmäßig in alle Raumrichtungen.</p> <p><i>Physikalisch:</i><br/><b>Strahlungsleistung [W]</b></p> <p><i>Fotometrisch:</i><br/><b>Lichtstrom F [lm]</b><br/>(Lichtstrom ist Strahlungsleistung, gewichtet mit der Hellempfindlichkeit V des Auges, multipliziert mit dem maximalen fotometrischen Strahlungsäquivalent <math>k_m = 683 \text{ lm/W}</math>)</p> | <p>Lichtstrom F, kleineres <math>\Omega</math></p> <p>Lichtstrom F, größeres <math>\Omega</math></p> <div data-bbox="539 1451 619 1697"></div> <p>Wenn ein gegebener <b>Lichtstrom F</b> (symbolisiert durch 3 Pfeile) auf einen kleinen <i>Raumwinkel</i> gebündelt ist (links), resultiert eine höhere <b>Lichtstärke</b>, als wenn sich derselbe Lichtstrom auf einen größeren Raumwinkel <math>\Omega</math> verteilt (rechts).</p> | <p>Lichtstärke I<br/>Fläche</p> <p>Bezieht man die <b>Lichtstärke I</b> auf die <i>abstrahlende Fläche</i>, erhält man die <b>Leuchtdichte L</b>. Die Leuchtdichte dient z.B. dazu, die Helligkeit eines Monitors zu charakterisieren.</p> | <p>Lichtstrom F<br/>(Empfänger-)Fläche</p> <p>Wird der <b>Lichtstrom</b> auf die <i>Empfängerfläche</i> bezogen, erhält man die <b>Beleuchtungsstärke E</b>. Sie charakterisiert z.B. die Helligkeit an einem Arbeitsplatz oder die Helligkeit des Normlichts zur Begutachtung von Druckergebnissen.</p> | <p>Sensor, Film</p> <p>Welche <b>Beleuchtungsstärke</b> auf dem Sensor bzw. dem Film einer Kamera herrscht, hängt von der <i>Helligkeit des Motivs</i> und dem Öffnungsgrad der <i>Blende</i> ab.</p> <p>Wie stark die lichtempfindlichen Zellen des Sensors angeregt bzw. wie stark Film geschwärzt wird, hängt auch davon ab, <i>wie lange</i> eine bestimmte Beleuchtungsstärke einwirkt = Belichtung (= Integral der Beleuchtungsstärke E über die Zeit t). Diese Zeit t entspricht der <i>Verschlusszeit</i> beim Fotografieren. Da E für kurze Verschlusszeiten (z.B. 1/100 s) meist konstant ist, vereinfacht sich das Integral zum Produkt <math>E \cdot t</math>.</p> |

Abb. 2.3.1.3 | Fotometrische Größen im Überblick.

## SI-Einheiten der Fotometrie

(SI = Internationales Einheitensystem, Système international d'unités)

- Candela [ $\text{cd} = \text{lm}/\text{sr}$ ]: *SI-Basiseinheit* für die Lichtstärke  
Candela ist definiert als diejenige Lichtstärke in einer bestimmten Richtung einer Strahlungsquelle, die monochromatische Strahlung der Frequenz  $540 \cdot 10^{12}$  Hz (entspricht einer Wellenlänge  $\lambda = 555$  nm) aussendet und deren Strahlstärke in dieser Richtung  $1/683$  Watt pro Steradian beträgt.
- Lumen [ $\text{lm} = \text{cd} \cdot \text{sr}$ ]: *abgeleitete* SI-Einheit für den Lichtstrom.  
Beispiel: Wenn eine isotrop (in alle Richtungen gleich) strahlende Lichtquelle mit 1 cd leuchtet, also  $1 \text{ lm}/1 \text{ sr}$  aussendet, ergibt sich der Lichtstrom durch die Summierung (Integral) der Lichtstärke über den gesamten Raum (voller Raumwinkel =  $4\pi \text{ sr}$ ), der die Lichtquelle umgibt:  
 $4\pi \text{ sr} \cdot 1 \text{ lm}/1 \text{ sr} = 4\pi \text{ lm} \approx 12,57 \text{ lm}$
- Lux [ $\text{lx} = \text{lm}/\text{m}^2$ ]: *abgeleitete* SI-Einheit für die Beleuchtungsstärke  
Maß für die Helligkeit, mit der ein Objekt, z. B. eine Arbeitsfläche, ausgeleuchtet wird.  
Beispiel: Bei Druckabmusterung gilt  $2000 \pm 500 \text{ lx}$  als Norm.  
Anmerkung: Für monochromatisches Licht mit 555 nm entspricht die radiometrische Bestrahlungsstärke von  $1 \text{ W}/\text{m}^2$  der fotometrischen Beleuchtungsstärke von 683 lux.

## <!> Zusammenfassung „Lichttechnische Grundlagen“

- Die Summe der elektromagnetischen Strahlung eines Strahlers (unabhängig von der Sichtbarkeit) ist die radiometrische Strahlungsleistung  $\Phi_e$  [W].
- Der Anteil der elektromagnetischen Strahlungsleistung, der in dem für den Menschen sichtbaren Bereich zwischen 380 und 780 nm Wellenlänge liegt, wird als Lichtstrom  $F$  ( $\Phi_v$  [lm]) bezeichnet (Luminous Flux). In  $\Phi_v$  geht die Hellempfindlichkeit  $V$  (zwischen 0 und 1) des Auges für unterschiedliche Wellenlängen ein. Das Maximum beim Tagsehen liegt bei  $\lambda = 555$  nm, beim Nachtsehen bei 505 nm. Die Konstante in der Beziehung von  $\Phi_v$  und  $\Phi_e$  ist das maximale fotometrische Strahlungsäquivalent  $K_m$ .  
Es gilt:  
 $\Phi_v(\lambda) = K_m \cdot V(\lambda) \cdot \Phi_e = K(\lambda) \cdot \Phi_e$   
Formt man die Gleichung um, ergibt sich:  
$$\frac{\Phi_v(\lambda)}{\Phi_e} = K(\lambda)$$
  
Für  $V = 1$  (d. h.  $\lambda = 555$  nm) erreicht  $K(\lambda)$  das Maximum  $K_m$   
 $K_m = 683 \text{ lm}/\text{W}$ .
- $\Phi_v$  ist der gesamte Lichtstrom  $F$ , den eine elektromagnetische Strahlungsquelle aussendet. Bezieht man den Lichtstrom auf den Raumwinkel  $\Omega$ , erhält man die Lichtstärke  $I$  (Luminous Intensity) in Candela [ $\text{cd} = \text{lm}/\text{sr}$ ].
- Bezieht man die Lichtstärke  $I$  auf die abstrahlende Fläche (z. B. eines Monitors), erhält man die Leuchtdichte  $L$  (Luminance) [ $\text{cd}/\text{m}^2$ ].
- Bezieht man den Lichtstrom  $F$  auf die Empfängerfläche, erhält man die Beleuchtungsstärke  $E$  (Luminous Emittance, Luminous Exitance) [ $\text{Lux} = \text{lx} = \text{lm}/\text{m}^2$ ].
- Das zeitliche Integral des Lichtstroms  $F$  entspricht der abgestrahlte Lichtmenge [ $\text{lm} \cdot \text{s}$ ].
- Das zeitliche Integral der Beleuchtungsstärke  $E$  (Lichtstrom/Fläche [ $\text{lx} = \text{lm}/\text{m}^2$ ]) beschreibt die Belichtung  $H_v$  [ $\text{lx} \cdot \text{s} = \text{lm} \cdot \text{s}/\text{m}^2$ ]. Belichtung ist die Lichtmenge (Lichtstrom  $\cdot$  Zeit, siehe oben), die auf einer bestimmten Fläche (z. B. auf dem Film oder auf dem Sensor einer Kamera) auftrifft.

## 3 Lichttechnik in der Fotografie

### Übersicht

Ein Foto gilt als gut, wenn

- ▶ das Motiv und der Bildausschnitt geschickt gewählt werden,
- ▶ die Schärfe bzw. Unschärfe passend gesetzt sind,
- ▶ die optimale Lichtmenge auf den Film bzw. Sensor\* der Kamera trifft.

In diesem Kapitel geht es in erster Linie um die korrekte Belichtung. Um die Schärfe geht es zum einen bei der Erklärung der allgemeinen Abbildungsgleichung (Linsengleichung), zum anderen bei der Wahl der Blende, die Einfluss auf die Schärfentiefe hat.

- 3.1 Brennweite und Sensorformat
  - 3.2 Blende und Blendenstufe
  - 3.3 Schärfentiefe
  - 3.4 Fotografische Lichtstärke eines Objektivs
  - 3.5 Belichtungszeit
  - 3.6 Lichtempfindlichkeit
  - 3.7 Lichtwert und Belichtung
  - 3.8 Weißabgleich
- 

### 3.1 Brennweite (Focal Length) und Sensorformat

<Ø>

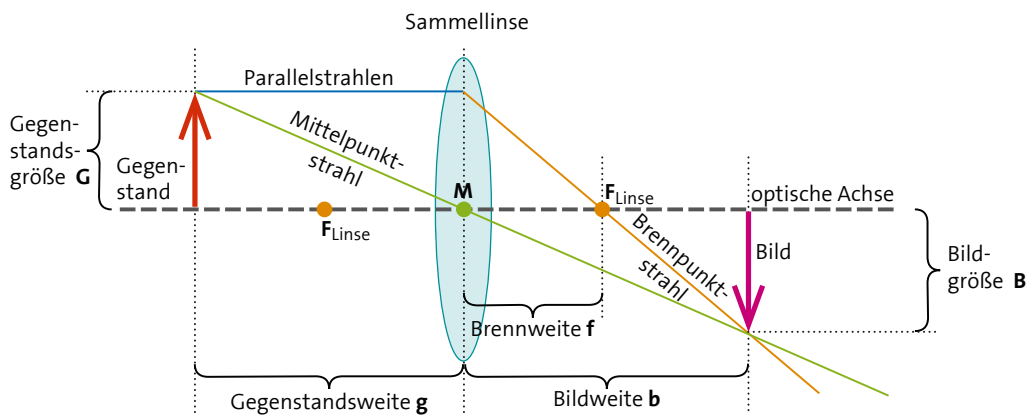
#### 3.1.1 Abbildungsgleichung

Abbildung 3.1.1 zeigt vereinfacht den Verlauf von Lichtstrahlen an einer dünnen Sammellinse.

Eine Linse ist charakterisiert durch die *Brennweite*  $f$ . Brennweite ist der Abstand des *Brennpunkts*  $F$  von der Linsenmitte  $M$  auf der optischen Achse, auf dem ein parallel einfallender Lichtstrahl (Parallelstrahl) nach Durchlaufen der Linse die optische Achse schneidet. Wenn man eine Sammellinse in das parallel einfallende Sonnenlicht hält und ein Papier im Abstand  $f$  parallel zur Linse liegt, dauert es nicht lange, bis das Papier brennt. Daher der Name Brennpunkt. Es gibt einen gegenstandsseitigen und einen bildseitigen Brennpunkt.

Damit ein Gegenstand scharf abgebildet wird, müssen sich Mittelpunkt- und Parallelstrahlen treffen. Wenn ein Gegenstand sehr weit entfernt ist (Gegenstandsweite  $g \rightarrow \infty$ ), nähert sich die Abbildungsebene der Ebene des Brennpunkts ( $b \approx f$ ). Je näher der Gegenstand rückt (d. h. je kleiner  $g$  wird), desto größer muss der Abstand zwischen Linse und Abbildungsebene (= Ebene des Sensors) sein, d. h. die Bildweite  $b$  wird größer.

\* Die meisten Inhalte dieses Kapitels gelten sowohl für die analoge Fotografie auf Film als auch die Digitalfotografie. Wenn im Folgenden von Sensor oder Sensorfläche geschrieben wird, ist damit auch Film gemeint, wenn nicht anders angegeben.



**Brennweite  $f$**  = Abstand des Brennpunkts F zur Linsenmitte M

**Brennpunkt F** = Punkt hinter/vor der Sammellinse, an dem parallel zur optischen Achse einfallende Strahlen die optische Achse schneiden

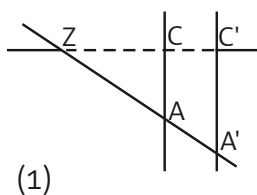
Abb. 3.1.1 | Vereinfachte Darstellung des Strahlenverlaufs an einer dünnen Sammellinse.

Die Abbildungsgleichung (Linsengleichung) lässt sich mithilfe der beiden Strahlensätze der Geometrie herleiten. Dies zeigt Abbildung 3.1.2.

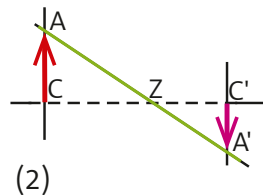
1. und 2. Strahlensatz der Geometrie (ähnliche Dreiecke):

(1)  $\overline{ZA} : \overline{AA'} = \overline{ZC} : \overline{CC'}$  und  $\overline{ZA} : \overline{ZA'} = \overline{ZC} : \overline{ZC'}$  (die parallelen Geraden liegen auf der gleichen Seite von Z, „Y-Figur“)

(2)  $\overline{AC} : \overline{A'C'} = \overline{ZA} : \overline{ZA'}$  und  $\overline{AC} : \overline{A'C'} = \overline{ZC} : \overline{ZC'}$  (Z liegt zwischen den parallelen Geraden, „X-Figur“)



(1)



(2)

An einer Sammellinse gilt (2. Strahlensatz):

$$\left. \begin{array}{l} \text{(I)} \quad \frac{B}{G} = \frac{b}{g} \\ \text{(II)} \quad \frac{B}{G} = \frac{b-f}{f} \end{array} \right\} \Rightarrow \frac{b}{g} = \frac{b-f}{f} \Rightarrow$$

$$\Rightarrow \frac{b}{g \cdot b} = \frac{b-f}{f \cdot b} \Rightarrow \frac{1}{g} = \frac{b-f}{f \cdot b} = \frac{b}{f \cdot b} - \frac{f}{f \cdot b} \Rightarrow$$

$$\Rightarrow \frac{1}{g} = \frac{1}{f} - \frac{1}{b} \Rightarrow \boxed{\frac{1}{f} = \frac{1}{b} + \frac{1}{g}}$$

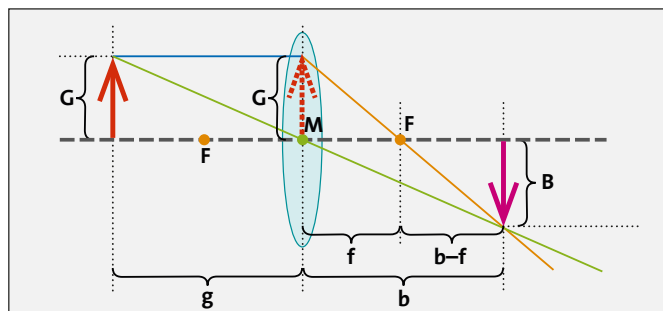


Abb. 3.1.2 | Herleitung der Abbildungsgleichung (rot umrandet) mithilfe der Strahlensätze der Geometrie. Oben: Erster (1) und zweiter (2) Strahlensatz. Die beiden Darstellungen bei (2) sind identisch, die unterschiedlichen Farben erleichtern die Übertragung auf den Strahlenverlauf an einer Sammellinse (unterer Teil der Abbildung). Unten: Übertragung der Strahlensätze auf den Strahlenverlauf an einer dünnen Sammellinse (grau hinterlegt, vgl. Abb. 3.1.1).

(Die Abbildungsgleichung gehört zum Grundlagenwissen Optik, ist aber *nicht* Gegenstand der Prüfung GDVT.)

</>

### 3.1.2 Auswirkungen der Brennweite

Die Brennweite eines Objektivs bestimmt zusammen mit dem Aufnahmeformat (Sensorformat) den Bildwinkel und damit den Bildausschnitt einer Aufnahme.

- Von einem *Normalobjektiv* spricht man, wenn gilt:  
Brennweite  $\approx$  Diagonale des Aufnahmeformats (Film- bzw. Sensorfläche)  
Bezogen auf Kleinbildformat (24 x 36 mm; Bilddiagonale  $\approx$  43,2 mm) weist ein Normalobjektiv eine Brennweite von rund 43 mm auf. (Sog. Standardobjektive für das Kleinbildformat weisen daher häufig eine Brennweite von 50 mm auf.)

#### Bildwinkel

Abbildung 3.1.2.1 zeigt die Berechnung des Bildwinkels, wenn Brennweite und Sensorfläche gegeben sind, am Beispiel eines Normalobjektivs.

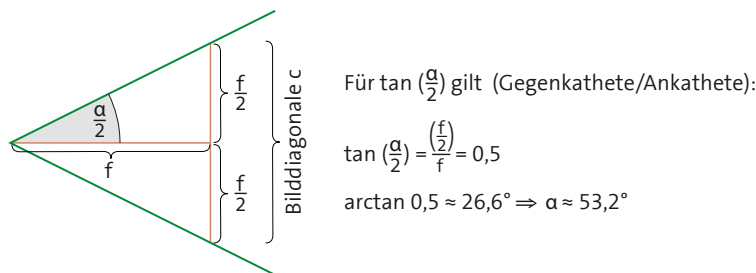


Abb. 3.1.2.1 | Berechnung des Bildwinkels  $\alpha$  am Beispiel eines Normalobjektivs.

- <Ø> Beispiel: Berechnung des Bildwinkels für ein Objektiv mit einer Brennweite  $f$  von 80 mm auf einer Kleinbildkamera (Sensorfläche 24 x 36 mm):

Für die Bilddiagonale  $c$  gilt (rechtwinkliges Dreieck):

$$(c)^2 = (24 \text{ mm})^2 + (36 \text{ mm})^2$$

Es errechnet sich eine Diagonale  $c$  von 43,2 mm.  $\frac{c}{2} = 21,6 \text{ mm}$

$$\text{Für } \tan\left(\frac{\alpha}{2}\right) \text{ ergibt sich somit: } \tan\left(\frac{\alpha}{2}\right) = \frac{21,6 \text{ mm}}{80 \text{ mm}} = 0,27$$

Ein Tangens von 0,27 entspricht einem Winkel von  $15,1^\circ$ , d. h. der Bildwinkel beträgt  $30,2^\circ$  (siehe auch Abbildung 3.1.2.4)

</Ø>

Abbildung 3.1.2.2 zeigt den Zusammenhang zwischen Brennweite und Bildwinkel bezogen auf das Kleinbildformat. Allgemein gilt: kurze Brennweite = großer („weiter“) Bildwinkel.

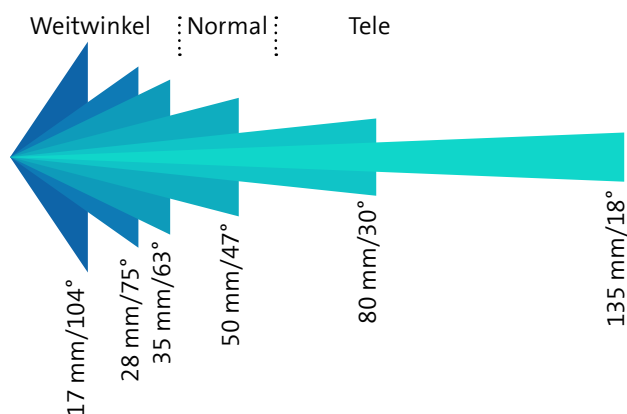


Abb. 3.1.2.2 | Brennweite und resultierender Bildwinkel bei Kleinbildformat (24 x 36 mm).



## Bildausschnitt, Perspektive

Bei zunehmender Brennweite und gleichbleibendem Bildabstand wird der Bildausschnitt kleiner. Der kleinere Bildausschnitt wird auf das gesamte Bildformat projiziert und dadurch im Vergleich zur Abbildung mit einem Objektiv kürzerer Brennweite vergrößert dargestellt (Abb. 3.1.2.3).

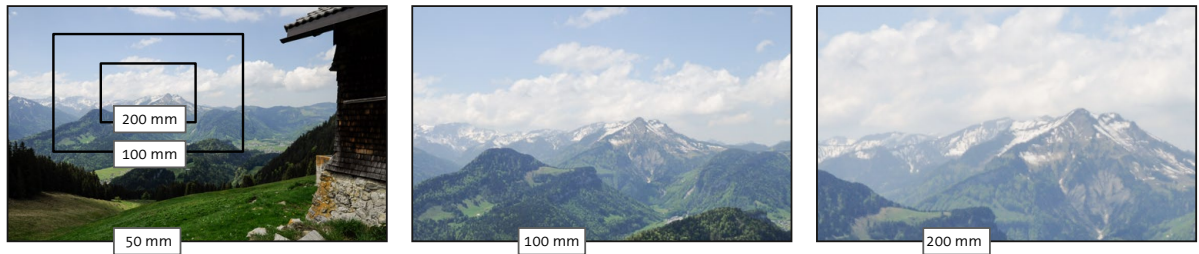


Abb. 3.1.2.3 | Wenn sich die Brennweite verdoppelt, wird ein Motiv doppelt so groß abgebildet.

Abbildung 3.1.2.4 zeigt, wie der Bildwinkel von der Brennweite und vom Sensorformat abhängt.

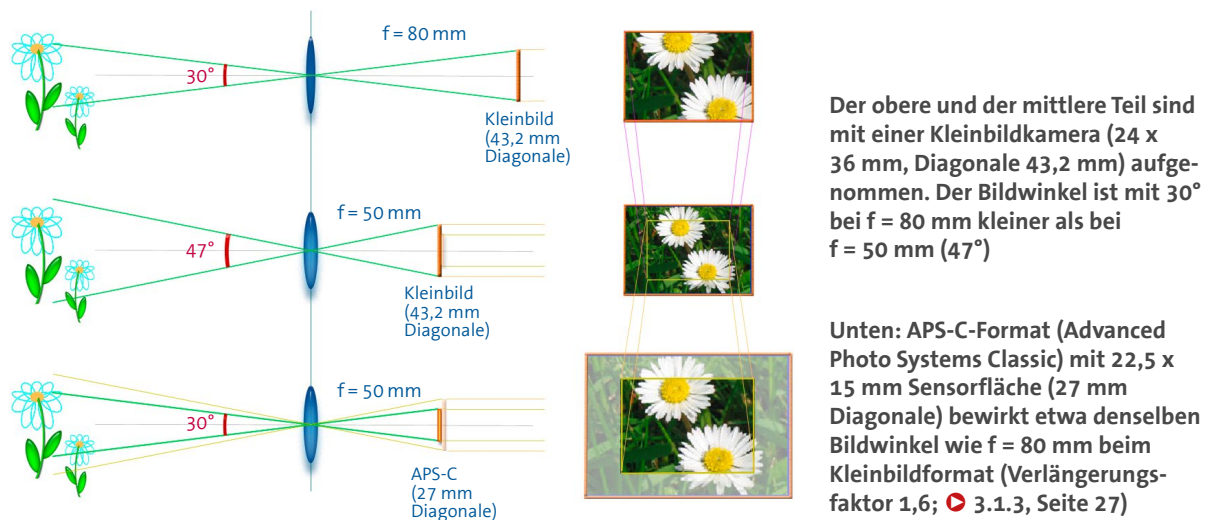


Abb. 3.1.2.4 | Brennweite und Sensorgröße bestimmen den Bildwinkel.  
Quelle: <http://de.wikipedia.org/wiki/Bildwinkel>

Abbildung 3.1.2.5 zeigt, wie sich die Brennweite auf die Bildperspektive auswirkt.

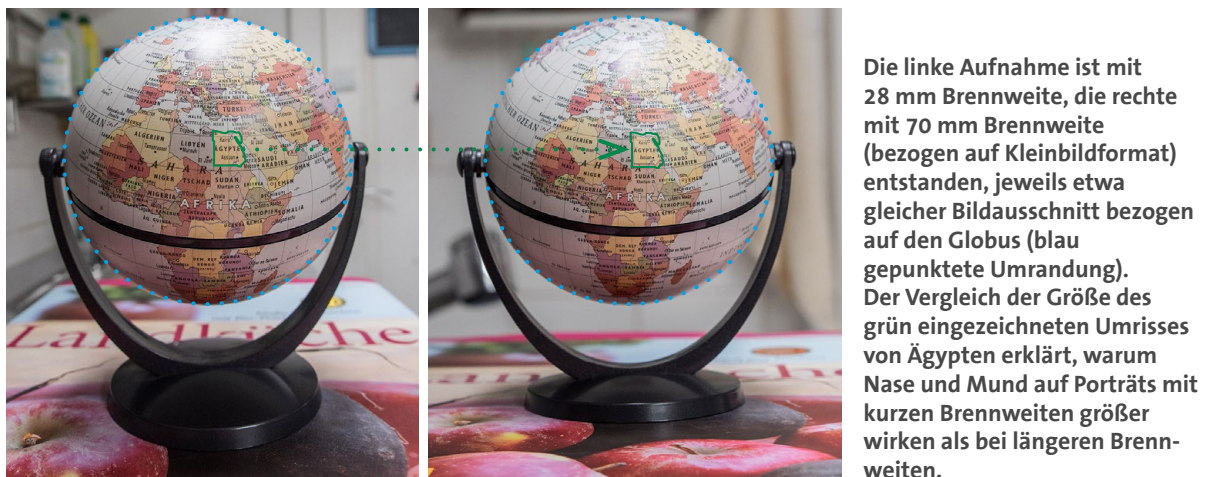


Abb. 3.1.2.5 | Einfluss der Brennweite auf die Bildperspektive.



Einteilung fotografischer Objektive nach Brennweite (immer bezogen auf Kleinbildformat)

■ **Normalobjektiv**

Brennweite entspricht in etwa der Diagonale des Aufnahmeformats.

■ **Teleobjektiv**

lange Brennweiten und kleine Bildwinkel; typische Brennweiten: 135 mm, 200 mm, 400 mm etc. (u. a. 80 bis 135 mm für Porträtfotografie).

■ **Weitwinkelobjektiv**

kurze Brennweiten und große Bildwinkel (Landschaften, Straßenfotografie).  
Die Brennweite von Smartphonekameras liegt in der Größenordnung von 30 mm.

■ **Fischaugenobjektiv**

extrem kleine Brennweite im Verhältnis zum Bildformat  $\Rightarrow$  extrem großer Bildwinkel, oft  $180^\circ$ , bis  $> 200^\circ$  ist möglich. Linien, die nicht durch die Bildmitte laufen, werden verzerrt.

■ **Zoomobjektive**

Veränderung der Brennweite möglich; Kategorisierung nach dem Verhältnis zwischen längster und kürzester Brennweite, z. B. 1:3 Zoomobjektiv (z. B. 14 bis 42 mm).

■ **Wichtig:**

▷ Kurze Brennweite = großer Bildwinkel = verkleinernde Abbildung\*

▷ Lange Brennweite = kleiner Bildwinkel = vergrößernde Abbildung\*

\* Bezogen auf ein Normalobjektiv

### 3.1.3 Sensorformate, Brennweitenfaktor (Crop Factor)

Angaben zur Brennweite sind im Allgemeinen auf das Kleinbildformat von 24 x 36 mm bezogen (Diagonale: 43,2 mm).

Die Sensoren der meisten Digitalkameras sind kleiner als 24 x 36 mm. Falls ein Objektiv verwendet wird, das auf einer Kleinbildkamera als Normalobjektiv (Bildwinkel etwa  $53^\circ$ , ▶ Seite 25) gilt, ist der Bildwinkel jetzt kleiner als  $53^\circ$ . Das Normalobjektiv verhält sich wie ein Teleobjektiv, d. h. die effektive Brennweite verlängert sich.

Um den Bildwinkel eines Normalobjektivs auf kleineren Sensoren unterzubringen, ist folglich eine kürzere Brennweite erforderlich. Zur Umrechnung auf die Verhältnisse beim Kleinbildformat dient der Brennweitenfaktor (synonym: Verlängerungsfaktor, Crop Factor).

■ **Berechnung des Brennweitenfaktors an einem Beispiel:**

Ein Sensor, der statt 36 mm (Kleinbild) nur 18 mm breit ist (Verhältnis der Bildbreiten 2:1), bedingt einen Brennweitenfaktor von 2. Ein 30-mm Objektiv (bezogen auf Kleinbild) verhält sich bei dem kleineren Sensor folglich wie ein 60-mm-Objektiv.

▷ Hinweis: Der Brennweitenfaktor entspricht dem Verhältnis der Sensorabmessungen in *einer* Dimension (z. B. Länge, Breite oder Diagonale), *nicht* dem Verhältnis der Flächen.

Abbildung 3.1.3.1 gibt einen Überblick über gängige Sensorformate und den zugehörigen Verlängerungsfaktor.

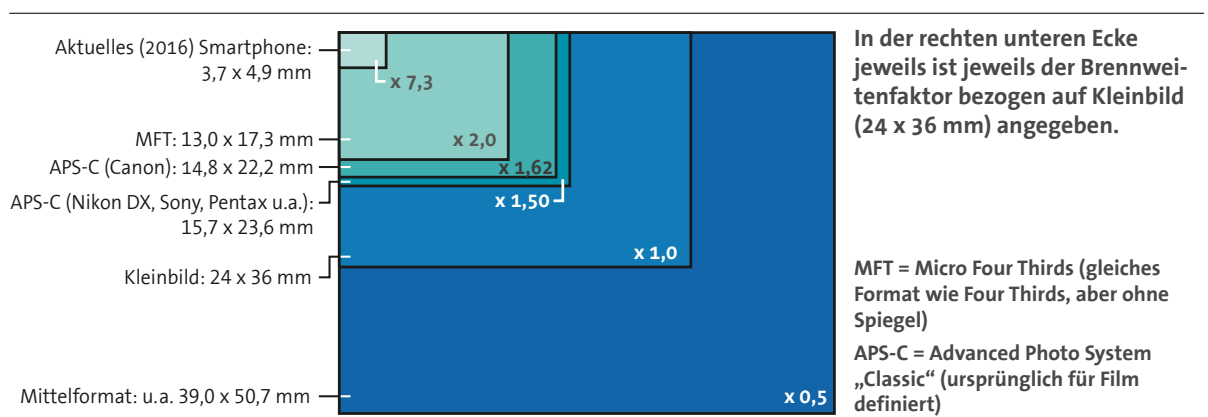


Abb. 3.1.3.1 | Sensorformate (in der Abbildung auf etwa 130 % vergrößert) und Brennweitenfaktor.

## 3.2 Blende (Aperture)



Abb. 3.2.1 | Unterschiedlich weit geöffnete Blende.

### 3.2.1 Blende und Blendenzahl k (Blendenstufe, F-Number, F-Stop)

Abbildung 3.2.1 oben zeigt die Funktion der Blende: Ist sie offen, kann viel Licht durch das Objektiv treten, ist sie klein, gelangt nur wenig Licht auf den Sensor/Film. Die Einstellung der Blende ist einer der drei Faktoren, die über die eintretende Lichtmenge (= Belichtung) entscheiden. Neben der Blende sind das die Verschlusszeit (Belichtungszeit, ► 3.5, Seite 33) und die Empfindlichkeit des Sensors (► 3.6, Seite 35).

#### Blendenzahl k

$k = \frac{f}{d}$  mit  $d$  = Durchmesser der Blendenöffnung sowie  $f$  = Brennweite, beides in mm ( $k$  ist dimensionslos)

- ✂  $k$  ist umgekehrt proportional zum Durchmesser der Blende.  
Das bedeutet: große Blendenzahl = kleine Blende = wenig Licht  
(häufige Quelle für Missverständnisse)

#### Berechnung der Blendenöffnungsfläche

Mit

$A$  = Fläche der annähernd kreisförmigen Blendenöffnung

$r$  = Radius der Blendenöffnung ( $\Rightarrow D = 2r$  = Durchmesser der Blendenöffnung)

$k$  = Blendenzahl

$f$  = Brennweite

gilt:

$$k = \frac{f}{d} \Rightarrow d = \frac{f}{k}$$

$$\Rightarrow A = \pi r^2 = \pi \left(\frac{d}{2}\right)^2 = \pi \frac{d^2}{4} = \frac{\pi f^2}{4k^2}$$

- <Ø> Die folgende Tabelle stellt Blende/Blendenzahl  $k$  und Blendenfläche  $A$  gegenüber. In der rechten Spalte ist die Blendenfläche  $A_n$  auf die Fläche  $A_1$  (Blende 1) bezogen.

| Blende | $k$                         | Blendenfläche $A_n$                                                          | Verhältnis $A_n$ zu $A_1$         |
|--------|-----------------------------|------------------------------------------------------------------------------|-----------------------------------|
| 1      | $k = (\sqrt[3]{2})^0 = 1$   | $A_1 = \frac{\pi}{4} \cdot \frac{f^2}{1^2} = \frac{\pi}{4} \cdot f^2$        | $A_1 = 1 \cdot A_1$               |
| 1,4    | $k = (\sqrt[3]{2})^1 = 1,4$ | $A_{1,4} = \frac{\pi}{4} \cdot \frac{f^2}{1,4^2} = \frac{\pi}{8} \cdot f^2$  | $A_{1,4} = \frac{1}{2} \cdot A_1$ |
| 2      | $k = (\sqrt[3]{2})^2 = 2$   | $A_2 = \frac{\pi}{4} \cdot \frac{f^2}{2^2} = \frac{\pi}{16} \cdot f^2$       | $A_2 = \frac{1}{4} \cdot A_1$     |
| 2,8    | $k = (\sqrt[3]{2})^3 = 2,8$ | $A_{2,8} = \frac{\pi}{4} \cdot \frac{f^2}{2,8^2} = \frac{\pi}{32} \cdot f^2$ | $A_{2,8} = \frac{1}{8} \cdot A_1$ |
| usw.   |                             |                                                                              |                                   |

<Ø>

## Blendenstufen

Die ganzen Blendenstufen sind eine Potenz von  $(\sqrt[3]{2})^n$  (n ist eine natürliche Zahl  $\geq 0$ ):

1 | 1,4 | 2 | 2,8 | 4 | 5,6 | 8 | 11 | 16 | 22 usw.

Moderne Kameras erlauben auch halbe (1,2 | 1,7 | 2,4 usw.) und Drittelblendenstufen.

- ⚡ Verdoppeln der Blendenzahl entspricht einem Sprung um 2 Blendenstufen ( $2 \cdot \sqrt[3]{2}$ ).

### 3.2.2 Blende und Belichtung

Bei Verdoppelung der Blendenzahl k (z. B. von 8 auf 16) reduziert sich die Belichtung auf ein Viertel. Erklärung: Die Fläche A der annähernd kreisförmigen Blendenöffnung beträgt  $r^2 \cdot \pi$ .

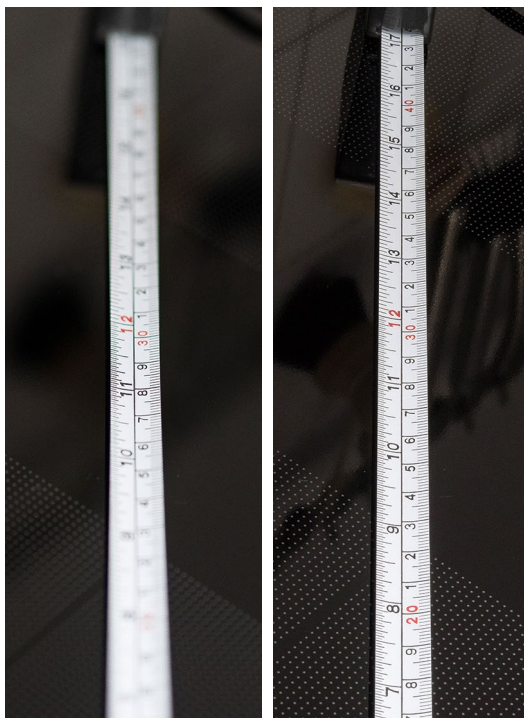
Daraus folgt: Wenn sich der Radius halbiert, verringert sich die Fläche der Blendenöffnung auf ein Viertel.

- ⚡ Verändern der Blende um *zwei Blendenstufen* (Blendenzahl k mal 2 bzw. geteilt durch 2) hat zur Folge, dass die Belichtung um den Faktor 4 steigt bzw. auf ein Viertel verringert wird.
- ⚡ Verändern der Blende um *eine Blendenstufe* (Blendenzahl k mal  $\sqrt[3]{2}$  bzw. geteilt durch  $\sqrt[3]{2}$ ) hat zur Folge, dass die Belichtung verdoppelt bzw. halbiert wird (Faktor 2).

Hinweis: Für die Angabe der Blendenzahlen wird  $\sqrt[3]{2}$  im Allgemeinen auf 1,4 gerundet.

### 3.3 Schärfentiefe (Depth of Field)

Wie Abbildung 3.1.1 auf Seite 24 zeigt, gibt es nur eine Gegenstandsweite g, bei der ein Gegenstand G bei gegebener Entfernungseinstellung der Linse scharf auf der Sensorebene abgebildet wird. Abbildung 3.3.1 zeigt aber, dass sich der vom Auge als *scharf wahrgenommene* Bereich unterschiedlich weit vor und hinter der Schärfebene ausdehnen kann. Dieser Bereich wird als Schärfentiefe (Depth of Field) bezeichnet und hängt, wie nachfolgend erklärt wird, im Wesentlichen von der Wahl der Blende ab.



Beispiel mit 90 mm Brennweite (bezogen auf Kleinbildformat)

Linke Aufnahme:

geringe Schärfentiefe, erreicht durch:  
Blende 1,8 (sehr große Öffnung) und  
1/15 s Verschlusszeit (ISO 400)

Rechte Aufnahme:

sehr große Schärfentiefe, erreicht durch:  
Blende 20 (sehr kleine Öffnung) und  
8 s Verschlusszeit (ISO 400)

Abb. 3.3.1 | Abhängigkeit der vom Auge wahrgenommenen Schärfentiefe von der Blende.

Im Beispiel der Abbildung 3.3.1 gilt für das Verhältnis der Blendenzahlen:

$$k_1 = 1,8; k_2 = 20; \Rightarrow \frac{k_2}{k_1} = 11,11$$

Mit  $f_1 = f_2$  (konstant 90 mm) und  $d = \frac{f}{k}$  (vgl. 3.2.1) gilt:

$$\frac{d_1}{d_2} = \frac{\frac{f}{k_1}}{\frac{f}{k_2}} = \frac{f \cdot k_2}{k_1 \cdot f} = \frac{k_2}{k_1} = \frac{11,11 \cdot k_1}{k_1} = 11,11$$

Daraus folgt für das Verhältnis der Blendenflächen:

$$\begin{aligned} \text{Blendenfläche } A_1 &= \pi \cdot \left(\frac{d_1}{2}\right)^2 \\ \text{Blendenfläche } A_2 &= \pi \cdot \left(\frac{d_2}{2}\right)^2 \end{aligned} \Rightarrow \frac{A_1}{A_2} = \frac{\pi \cdot \left(\frac{d_1}{2}\right)^2}{\pi \cdot \left(\frac{d_2}{2}\right)^2} = \frac{(d_1)^2}{(d_2)^2} = \frac{(11,11 \cdot d_2)^2}{(d_2)^2} = \frac{11,11^2 \cdot (d_2)^2}{(d_2)^2} = 11,11^2 \approx 123$$

Wenn im Beispiel in Abbildung 3.3.1 bei Blende  $k_1 = 1,8$  die rund 123-fache Lichtmenge im Vergleich zu  $k_2 = 20$  durchtritt, müssten bei gleicher Belichtung die Belichtungszeiten im umgekehrten Verhältnis zueinander stehen:  $8 \text{ s} : 1/15 \text{ s} \approx 120$ . (► 3.7 Lichtwert, Seite 36)

### 3.3.1 Erläuterung der Schärfentiefe anhand der Unschärfekreise

Wie bereits in 3.3 gesagt, gibt es nur eine Bildweite  $g$ , für die ein Gegenstand  $G$  scharf abgebildet wird. Die Tatsache, dass beim Betrachter der Eindruck von Schärfentiefe entsteht, lässt sich mithilfe des Konzepts der Unschärfekreise (Circles of confusion) erklären: Punkte, die außerhalb der Schärfeebene liegen, werden nicht als Punkt, sondern als Kreisfläche abgebildet. Der Durchmesser dieses (Unschärfe-)Kreises hängt, wie Abbildung 3.3.1.1 zeigt, sowohl vom Abstand des Punktes von der Schärfeebene als auch von der Öffnung der Blende ab. Wegen des begrenzten Auflösungsvermögens des Auges kann die Unschärfe erst ab einer bestimmten Größe der Unschärfekreise wahrgenommen werden.

Hinweis: Die Unschärfekreise erscheinen bei dieser 2-D-Projektion als vertikale Linien.

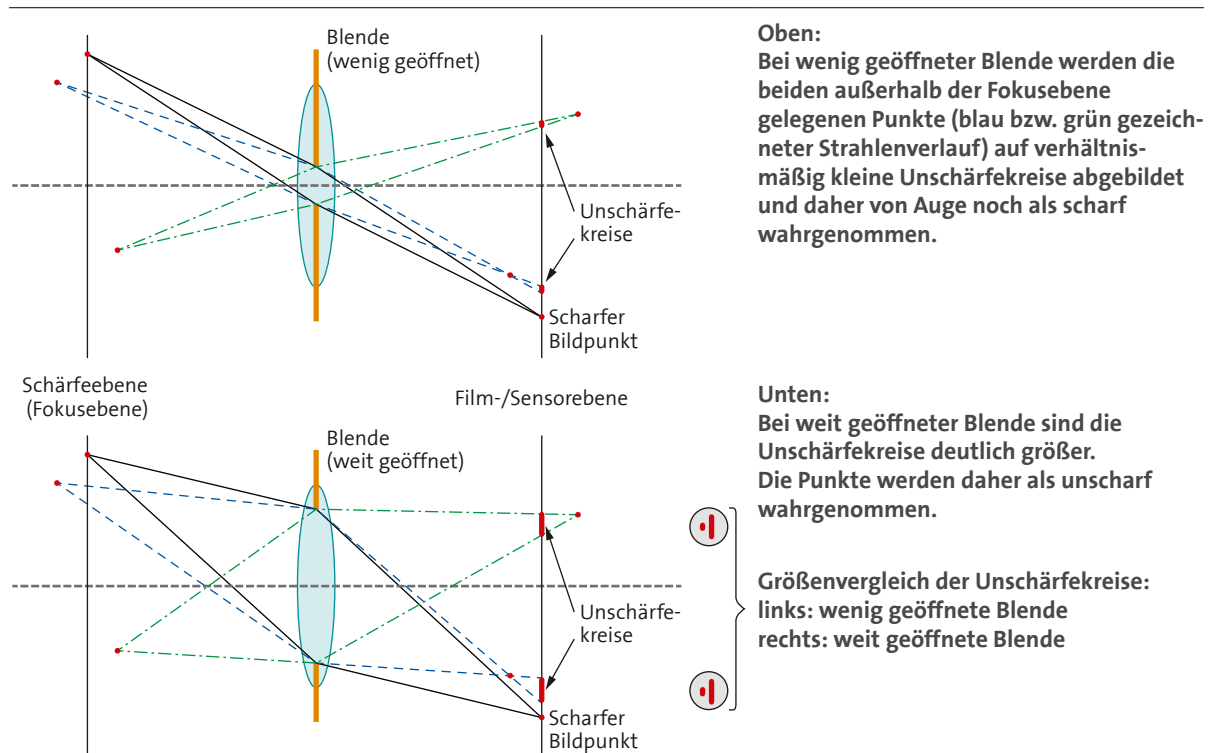


Abb. 3.3.1.1 | Die Blendenöffnung bestimmt weitgehend den Durchmesser der Unschärfekreise und damit die wahrgenommene Schärfentiefe.

### 3.3.2 Zwei verbreitete Irrtümer zur Schärfentiefe

#### Einfluss der Brennweite auf die Schärfentiefe

In vielen Lehrbüchern über Fotografie ist zu lesen, dass die Brennweite die Schärfentiefe beeinflusst: große Brennweite = geringe Schärfentiefe.

Dies steht in Übereinstimmung mit der praktischen Erfahrung, denn für eine Porträtfoto mit kleiner Schärfentiefe verwendet man eher ein (lichtstarkes) Objektiv mit mittelgroßer Brennweite (ca. 80 bis 100 mm bezogen auf Kleinbildformat) als ein Weitwinkelobjektiv.

Trotzdem hat die Brennweite nahezu keinen Einfluss auf die Schärfentiefe (berechnet anhand des Durchmessers der Unschärfekreise für verschiedene Brennweiten und identische Blende).

Der Hauptgrund dafür, dass die Schärfentiefe eines Teleobjektivs kleiner *erscheint* als die eines Weitwinkelobjektivs, liegt im *Vergrößerungsfaktor* durch Teleobjektive. Würde man ein definiertes Motiv mit einem Tele- und einem Weitwinkelobjektiv jeweils formatfüllend mit derselben Blendeneinstellung fotografieren, wäre die Schärfentiefe praktisch identisch! Dazu müsste man allerdings mit einem Weitwinkelobjektiv sehr nah an das Objekt herangehen und mit einem Teleobjektiv aus sehr großer Entfernung fotografieren.

Ein weiterer Faktor trägt dazu bei, dass die Schärfentiefe von Teleobjektiven bei den üblichen Abständen zum Motiv kleiner erscheint als die von Weitwinkelobjektiven: Große Brennweiten vergrößern wegen des engeren Bildwinkels den Hintergrund im Verhältnis zum Vordergrund. Ein unscharfer Hintergrund wird folglich als noch unschärfer empfunden, weil auch die Unschärfewirkung vergrößert wird.

#### Verteilung der Schärfentiefe

Eine häufig zitierte Faustregel besagt, dass etwa zwei Drittel der Schärfentiefe hinter der Fokusebene liegen, ein Drittel davor.

Wie folgende Tabelle zeigt, trifft das nur für sehr kleine Brennweiten (unter etwa 20 mm) zu, ab etwa 50 mm ist die Unschärfe vor und hinter der Fokusebene annähernd gleich verteilt.

| Brennweite (mm) | Verteilung der Schärfentiefe |                    |
|-----------------|------------------------------|--------------------|
|                 | hinter Fokusebene (%)        | vor Fokusebene (%) |
| 10              | 70,2                         | 29,8               |
| 20              | 60,1                         | 39,9               |
| 50              | 54,0                         | 46,0               |
| 100             | 52,0                         | 48,0               |
| 200             | 51,0                         | 49,0               |
| 400             | 50,5                         | 49,5               |

Quelle: <http://www.cambridgeincolour.com/tutorials/depth-of-field.htm>

Fotografiert man ein Motiv mit einem starken Weitwinkelobjektiv (z. B. einem Smartphone), reicht die Schärfentiefe deutlich weiter in den Hintergrund hinein als dies (beim selben Abbildungsmaßstab auf dem Sensor!) mit einem Teleobjektiv der Fall wäre. Dieser Effekt trägt dazu bei, dass die Schärfentiefe von Weitwinkelobjektiven größer erscheint als die von Teleobjektiven (siehe oben).



### 3.3.3 Schärfentiefe in der Porträtfotografie

Für Porträtfotos wird oft mit sehr geringer Schärfentiefe gearbeitet. Das Gesicht (oft sogar nur die Augen) liegt in der Schärfebene, Vorder- und Hintergrund liegen in der Unschärfe. Dafür sind lichtstarke Objektive (d. h. mit möglichst niedriger kleinster Blendenzahl) erforderlich. Abbildung 3.3.3.1 zeigt das Prinzip geringer Schärfentiefe anhand von drei unterschiedlichen Blendeneinstellungen (Blende 1,8, 5,6 und 18).



Abb. 3.3.3.1 | Einfluss der Blende auf die Schärfentiefe in der Porträtfotografie: Auf dem mittleren Bild der oberen Reihe mit weit geöffneter Blende verschwimmen Vorder- und Hintergrund in gewollter Unschärfe.

### 3.4 Lichtstärke eines Objektivs (relative Öffnung, Lens Speed)

Hinweis: Die Lichtstärke einer Linse wird im Englischen mit „Speed“ bezeichnet. Geschwindigkeit soll dabei zum Ausdruck bringen, dass eine „schnelle“ Linse kurze Belichtungszeiten ermöglicht. Analog dazu wird Filmempfindlichkeit mit „Film Speed“ übersetzt, d. h. ein „High-Speed-Film“ erlaubt wegen seiner hohen Empfindlichkeit kurze Belichtungszeiten.

Die fotografische Lichtstärke eines Objektivs ist definiert als Quotient aus größtem möglichem Blendendurchmesser  $d_{\max}$  und Brennweite  $f$ .

$$\text{Lichtstärke} = \frac{d_{\max}}{f}$$

$$\text{Aus } k = \frac{f}{d} \text{ folgt:}$$

$$d_{\max} = \frac{f}{k_{\min}}$$

Für die Lichtstärke ergibt sich somit:

$$\text{Lichtstärke} = \frac{d_{\max}}{f} = \frac{f}{k_{\min} \cdot f} = \frac{1}{k_{\min}}$$

Je niedriger die kleinst mögliche Blendenzahl eines Objektivs (= maximal geöffnete Blende) ist, desto lichtstärker ist es.

Lichtstärken werden in der Form 1: kleinste Blendenzahl angegeben (z. B. 1:2,8). Oft wird die Schreibweise verkürzt und es wird als Lichtstärke lediglich die kleinste Blendenzahl genannt. Im Beispiel würde anstelle von 1:2,8 verkürzt 2,8 genannt.

Beispiel:

Ein Objektiv mit einer Brennweite  $f = 50 \text{ mm}$  und der kleinsten Blendenzahl  $k = 2,0$  hat eine Lichtstärke von 1:2,0. Der größtmögliche Blendendurchmesser des Objektivs beträgt:

$$d_{\max} = \frac{f}{k_{\min}} = \frac{50 \text{ mm}}{2,0} = 25 \text{ mm}$$

Die Tatsache, dass die Lichtstärke aus Quotient von maximal möglichem Blendendurchmesser  $d_{\max}$  und Brennweite  $f$  definiert ist, erklärt, warum lichtstarke Teleobjektive großen technischen Aufwand erfordern (großer Durchmesser) und teuer sind. Außerdem erklärt dies, dass viele Zoom-Objektive am unteren Ende ihres Brennweitenbereichs lichtstärker sind als im Telebereich. Beispiel für ein solches Zoom-Objektiv: 18–140 mm 1:3,5–5,6. Das bedeutet: Bei 18 mm Brennweite hat dieses Objektiv die Lichtstärke 1:3,5, bei 140 mm nur noch die Lichtstärke 1:5,6 (entsprechend 62,5 % der Lichtstärke bei 18 mm).

## 3.5 Belichtungszeit (Verschlusszeit, Shutter Speed)

Die Belichtungszeit ist die Dauer, die ein Film oder Sensor der Wirkung des Lichts ausgesetzt ist. Sie wird üblicherweise in Bruchteilen von Sekunden angegeben.

Typische Werte in Sekunden: 1/1000; 1/125; 1/60 ... 1/2; 1; 2, 4 usw.

Belichtungszeit und Blende bestimmen die Lichtmenge (Belichtung), die auf dem Sensor/Film ankommt.

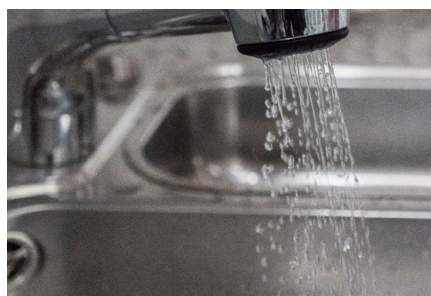
Je länger die gewählte Belichtungszeit ist, desto größer wird die Gefahr von Unschärfe, sei es durch „Verwackeln“ mit der Kamera oder weil sich das Objekt bewegt (Bewegungsunschärfe).

Als Faustregel gilt: Bei einer Belichtungszeit von maximal 1/Brennweite kann mit ruhiger Hand ohne Stativ fotografiert werden, ohne das Bild zu verwackeln. Beispiel: Mit einer Brennweite von 100 mm (bezogen auf Kleinbildformat), kann bis maximal 1/100 Sekunde belichtet werden.

Objektive bzw. Kameragehäuse mit Bildstabilisator verlängern die verwacklungsfreie Belichtungszeit um 2 und mehr Lichtwertstufen (→ 3.7 Lichtwert, Seite 36). Dabei gilt: 1 Lichtwertstufe Unterschied entspricht der Halbierung bzw. Verdoppelung der Belichtungszeit. Wichtig: Bei Stativaufnahmen sollte der Bildstabilisator abgeschaltet werden.

## Bewegungsunschärfe (Motion Blur)

Bei bewegten Motiven ist die Bewegungsunschärfe ein Stilmittel. Sie wird maßgeblich durch die Belichtungszeit (siehe oben) beeinflusst. Ein bekanntes Beispiel für Bewegungsunschärfe sind Fotos von fließendem Wasser. Je nach Belichtungszeit sind einzelne Tropfen erkennbar oder nur ein zusammenhängendes Band (Abb. 3.5.1).



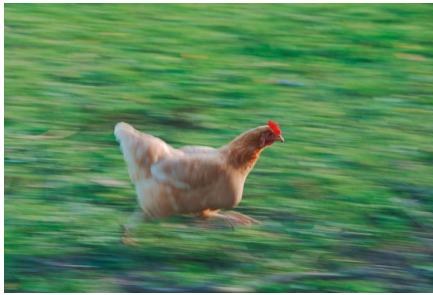
Wassertropfen:

Links: etwa 0,4 Sekunde,  
Blende 22, ISO 1600

Rechts: 1/500 Sekunde,  
Blende 2,8, ISO 6400

Abb. 3.5.1 | Bewegungsunschärfe durch Variieren der Belichtungszeit am Beispiel von fließendem Wasser.

Eine andere Art von Bewegungsunschärfe entsteht, wenn ein bewegtes Objekt mit ausreichend langer Verschlusszeit fotografiert und die Kamera im Winkeltempo des bewegten Objekts mitgezogen wird, siehe Abbildung 3.5.2.



Durch Mitziehen der Kamera beim Fotografieren eines bewegten Objekts wird die Bewegungsunschärfe auf den nicht bewegten Hintergrund übertragen.

Abb. 3.5.2 | Bewegungsunschärfe durch Mitziehen der Kamera.  
Quelle: Alvesgaspar / commons.wikimedia.org

## Wechselseitige Wirkung von Blende und Belichtungszeit auf Bewegungsunschärfe und Schärfentiefe

Eine bestimmte Belichtung lässt sich durch eine Vielzahl von Paaren aus Blende und Verschlusszeit erreichen (☛ 3.7, Lichtwert, Seite 36). Im folgenden Beispiel (Abb. 3.5.3) erreicht man mit Blende 2 und 1/1000 s Belichtungszeit dieselbe Belichtung wie mit Blende 8 und 1/60 s Belichtungszeit.

Was sich jedoch verändert, sind Bewegungsunschärfe und Schärfentiefe (☛ 3.3, Seite 29): Die Bewegungsunschärfe nimmt nach links zu kurzen Verschlusszeiten hin ab, die Schärfentiefe wächst nach rechts mit der Blendenzahl (= enger werdende Blende).

| Blenden-<br>zahl $k$     | 2      | 2,8   | 4     | 5,6   | 8    | 11   |
|--------------------------|--------|-------|-------|-------|------|------|
| Belichtungs-<br>zeit [s] | 1/1000 | 1/500 | 1/250 | 1/125 | 1/60 | 1/30 |

Abb. 3.5.3 | Einfluss von Blende und Belichtungszeit auf Bewegungsunschärfe und Schärfentiefe.

Die meisten Kameras bieten die Möglichkeit, entweder die Blende vorzugeben und die Belichtungszeit automatisch ermitteln zu lassen (oft als Modus A bezeichnet, wobei A für Aperture = Blende steht) oder umgekehrt die Verschlusszeit vorzugeben und die Blende automatisch ermitteln zu lassen (oft als Modus S bezeichnet, wobei S für Shutter Speed = Verschlusszeit steht).

Beispiel: Will man die Schärfentiefe im Griff haben, wird man die Blende gemäß obiger Beziehung (Abb. 3.5.3) fest einstellen, will man die Bewegungsunschärfe steuern, wird man die Verschlusszeit fest vorgeben (z. B. kurze Verschlusszeiten in der Sportfotografie).



Darüber hinaus weisen die meisten Kameras sog. Motivprogramme auf, die je nach Anforderung eine geeignete Gewichtung von Blende und Verschlusszeit vornehmen. Die folgende Tabelle zeigt die Prioritäten für Blendenstufe und Verschlusszeit bei typischen automatischen Kameramodi.

| Kameramodus                                                     | Blendenzahl k                             | Belichtungszeit [s]                                                          |
|-----------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------|
| Porträtmodus<br>(Person scharf, Hintergrund unscharf)           | Niedrig, da wenig Schärfentiefe gewünscht |                                                                              |
| Landschaft<br>(alle Details sollen scharf dargestellt werden)   | Hoch, um Schärfentiefe zu erhöhen         |                                                                              |
| Nacht<br>(Personen gegen Nachthimmel)                           |                                           | Vordergrund mit Blitz aufgehellte, lange Belichtungszeit für den Hintergrund |
| Sport<br>(schnelle Bewegungen einfrieren)                       |                                           | Kurz                                                                         |
| Bewegungen mit großer Unschärfe festhalten<br>(kein Einfrieren) |                                           | Lang                                                                         |

## 3.6 Fotografische Lichtempfindlichkeit (Film/Sensor Speed)

Bei der Analogfotografie auf Film bestimmen die chemischen und physikalischen Eigenschaften des Filmmaterials die Lichtempfindlichkeit, bei der Digitalfotografie wird die Sensorempfindlichkeit über die Signalverstärkung reguliert.

Traditionell wird die Lichtempfindlichkeit gemäß ASA (American Standards Association, heute ANSI [American National Standards Institute]) angegeben. Früher war auch die Angabe nach DIN gebräuchlich.

Die Skalierung erfolgt über die erforderliche Lichtmenge, die für eine definierte Bezugsbelichtung erforderlich ist. Die Bezugsbelichtung wird für unterschiedliche Filmmaterialien bzw. Sensoren (s/w oder bunt, negativ oder positiv, analog oder digital) spezifisch festgelegt.

### ASA/ANSI versus DIN

Die ASA-Skala ist *linear*, d. h. ein Film mit 200 ASA ist doppelt so empfindlich wie ein Film mit 100 ASA. Doppelt so empfindlich heißt, er benötigt entweder die halbe Belichtungszeit oder die Blendenöffnung kann um 1 Stufe verkleinert werden (z. B. von Blende 8 auf Blende 11).

Nach DIN erfolgt die Angabe mit der Einheit Grad (°). 15° DIN ist identisch mit 25 ASA. Die DIN-Skala ist jedoch *logarithmisch*. Ein Unterschied von 3° entspricht einer Verdoppelung der Empfindlichkeit. Ein Film mit 24° DIN ist demnach doppelt so empfindlich wie ein Film mit 21° DIN bzw. viermal so empfindlich wie ein Film mit 18° DIN.

### Angabe nach ISO

Standard ist heute die Angabe der Empfindlichkeit nach ISO (International Organization for Standardization). Dabei werden die Zahlenwerte von ASA/ANSI und DIN kombiniert.

Beispiel: Die Angabe ISO 100/21° steht für 100 ASA bzw. 21° DIN

Folgende Tabelle stellt die Eigenschaften und Anwendungsbereiche von Filmen verschiedener Empfindlichkeit gegenüber.

| ASA/ANSI | DIN | Eigenschaften                                                                                                                                                                                                      |
|----------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 50       | 18  | Sehr hohe Auflösung, niedrige Lichtempfindlichkeit<br>▷ Fotografie bei sehr großer Helligkeit<br>z.B. Naturfotografie in der Wüste, auf dem Meer, im Schnee<br>▷ Dias und Großformate (wegen sehr hoher Auflösung) |
| 100      | 21  | Hohe Auflösung, Standardempfindlichkeit<br>Bei mäßigen Lichtverhältnissen sind Stativ oder Blitz erforderlich.<br>Kurzzeitbelichtungen (z.B. Sportaufnahmen) erfordern sehr lichtstarke Objektive.                 |
| 200      | 24  | Mittlere Auflösung, höhere Lichtempfindlichkeit                                                                                                                                                                    |
| 400      | 27  | Fotografie mit langen Brennweiten und in Innenräumen                                                                                                                                                               |
| 800      | 30  | Hervorragend für Innenräume mit geringem Licht (available light),<br>aber: geringere Auflösung und Bildschärfe, höhere Körnigkeit                                                                                  |

Mit steigender Lichtempfindlichkeit sind Nachteile verbunden:

- ▶ geringere Auflösung des Films, höhere Körnigkeit (entspricht höherem Bildrauschen bei Digitalkameras)
- ▶ geringerer Kontrastumfang
- ▶ schlechtere Farbtreue

### 3.7 Lichtwert LW (Exposure Value, EV) und Belichtung

Die für eine korrekte Belichtung nötige Lichtmenge pro Fläche (das Produkt aus Beleuchtungsstärke und Belichtungszeit, ► Seite 20) hängt ab von:

- ▶ Blende
- ▶ Belichtungszeit
- ▶ Lichtempfindlichkeit des Films/Sensors

#### Definition des Lichtwerts

$$2^{LW} = \frac{k^2}{t}$$

mit  $k$  = Blendenzahl;  $t$  = Verschlusszeit

⇒  $LW = 2 \cdot \log_2(k) - \log_2(t)$

Für die Berechnung des Lichtwerts ist der Logarithmus zur Basis 2 maßgeblich.

Ein Lichtwert von 0 wird z. B. mit der Blendenzahl  $k = 1$  und der Belichtungszeit 1 s erreicht:

$$2 \cdot \log_2(1) - \log_2(1) = 0$$

Halbiert man die Belichtungszeit auf 0,5 s und erhöht die Blendenzahl auf 1,4 (1 Blendenstufe, entsprechend der Multiplikation mit  $\sqrt[3]{2} = 1,4$ ), ergibt sich derselbe Lichtwert:

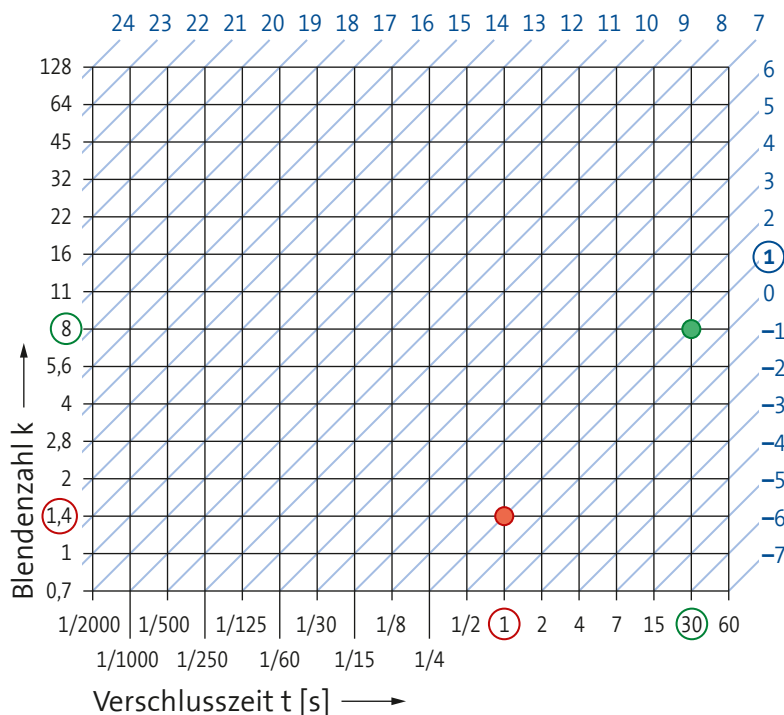
$$2 \cdot \log_2(1,4) - \log_2(0,5) = 2 \cdot 0,5 - (-1) = 0$$

Bei vorgegebener Lichtempfindlichkeit von Film/Sensor quantifiziert der Lichtwert ein Kombination von Blende und Verschlusszeit, die zur korrekten Belichtung führt. Dieser Lichtwert lässt sich durch zahlreiche Kombinationen (sog. Schar) aus Blende und Verschlusszeit erreichen (► Lichtwerttabelle, Seite 37).

- Halbieren/Verdoppeln der Belichtungszeit und Verringern/Erhöhen der Blendenzahl  $k$  um eine Stufe heben sich auf, das heißt es errechnet sich derselbe Lichtwert. Lichtwerte können negativ sein, z. B.: Blendenzahl 1, Belichtungszeit 4 s:  
 $2 \cdot \log_2(1) - \log_2(4) = 0 - 2 = -2$

## Lichtwerttabellen

Abbildung 3.7.1 zeigt die Kombinationen aus Belichtungszeit und Blendenzahl, die jeweils identische Lichtwerte ergeben. Jeder Diagonalen entspricht ein Lichtwert, d. h. alle Punkte auf einer Diagonalen haben denselben Lichtwert.



Ein bestimmter Lichtwert gilt für alle Kombinationen von k und t, die auf der zugehörigen 45°-Diagonalen liegen.

Beispiel:

Der rote und der grüne Punkt mit den Koordinaten

**t = 1 s** und **k = 1,4**

bzw.

**t = 32 s (30 s)** und **k = 8**

haben den **Lichtwert 1**

Abb. 3.7.1 | Lichtwerttabelle.

Jedem Lichtwert entspricht eine blau gezeichnete 45°-Diagonale. Ein Lichtwert lässt sich theoretisch mit beliebig vielen Kombinationen aus Blendenzahl und Verschlusszeit realisieren.

⚡ Der Lichtwert ist *nicht* die Lichtmenge [ $\text{lm} \cdot \text{s}$ ], die vom Motiv ausgesendet wird, oder die Belichtung (Lichtmenge pro Empfängerfläche, [ $\text{lm} \cdot \text{s}/\text{m}^2 = \text{lx} \cdot \text{s}$ ]), die auf dem Film/Sensor ankommt. Er charakterisiert nur eine bestimmte Kombination aus Blende und Verschlusszeit, die bei gegebener Motivhelligkeit und Film-/Sensorempfindlichkeit eine korrekte Belichtung ergibt.

### <∅> Hintergrundinformation

Ein Lichtwert von 0 steht für die Lichtmenge, die erforderlich ist, um einen S/W-Negativfilm mit einer Empfindlichkeit von ISO 100/21° bei einem Blendenwert von 1 und einer Verschlusszeit von 1 s zu 18 % (neutralgrau) zu belichten.

## Belichtung, Belichtungsmessung

Belichtung ist definiert als der Lichtstrom pro Fläche (= Beleuchtungsstärke [ $\text{lm}/\text{m}^2 = \text{lx}$ ]), der über eine bestimmte Zeit (= zeitliches Integral) auf einem Film bzw. Sensor auftrifft (► Seite 20). Die Einheit ist  $\text{lx} \cdot \text{s}$ .

Bei der Belichtungsmessung (intern durch die Kameralinse [through the lens, TTL] oder mit einem externen Belichtungsmesser) wird meistens die vom Objekt ausgehende Leuchtdichte (► Seite 19) bestimmt (*Objektmessung*). Bei der *Lichtmessung* wird das auf das Objekt treffende Licht gemessen. (Auf Details wie Integral- oder Spotmessung wird hier nicht eingegangen.)

Unter Berücksichtigung der Film-/Sensorempfindlichkeit gibt der Belichtungsmesser einen Lichtwert an, der bei den gegebenen Lichtverhältnissen eine optimale Belichtung erzielt. Wie zuvor beschrieben (► Lichtwerttabellen, Seite 37), kann ein bestimmter Lichtwert theoretisch durch beliebig viele Kombinationen von Blende und Verschlusszeit realisiert werden.

</Ø>

## Belichtungskorrektur

In der Praxis gibt es zahlreiche Situationen, in denen die automatisch ermittelte Belichtung manuell korrigiert werden muss.

Ein Beispiel: Ist der Motivkontrast größer als der Kontrastumfang des Films/des Sensors (z. B. bei starkem Gegenlicht), brechen die hellsten Stellen des Motivs aus (weiß ohne Detailzeichnung), während die dunkelsten Stellen „absaufen“ (schwarz ohne Detailzeichnung). Während dieses Clipping bei Filmmaterial sanft verläuft, ist Clipping in der Digitaltechnik hart, d. h. ohne Übergangsbereich (► Digitalisieren von Bildern, ab Seite 43). Durch manuelle Korrektur kann man entscheiden, ob man lieber die hellen oder die dunklen Bereiche mit Zeichnung darstellen will. Ersteres gelingt durch Verringern der Belichtung, Letzteres durch Erhöhen. In beiden Fällen wird dadurch das Clipping am jeweils anderen Bereichsende verstärkt.

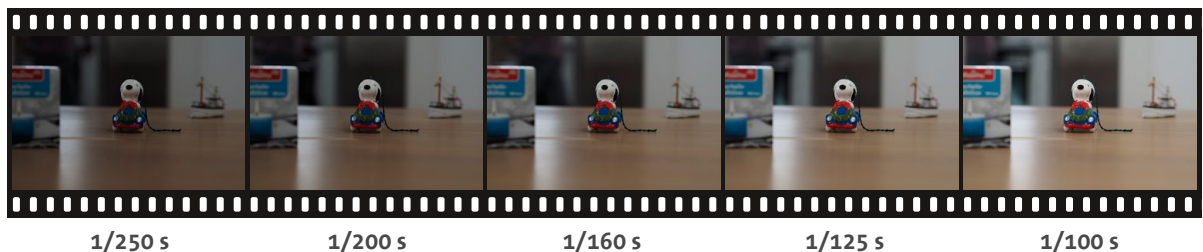
Die meisten Kameras erlauben eine manuelle Belichtungskorrektur des gemessenen Werts, häufig mit einer Schrittweite von  $\frac{1}{3}$  Lichtwert (EV).

## Belichtungsreihen (Exposure Bracketing)

Der Lichtwert wird auch zur Steuerung von Belichtungsreihen verwendet.

Abbildung 3.7.2 zeigt eine Beispiel mit 5 Aufnahmen, die sich um jeweils  $\frac{1}{3}$  EV unterscheiden

Blendenzahl  $k = 3,2$ ; ISO = 200; Belichtungszeiten:  $1/250$ ,  $1/200$ ,  $1/160$ ,  $1/125$ ,  $1/100$  s.



Der Unterschied zwischen dem 1. und 4. Bild beträgt + 1 EV. Ein Unterschied von + 1 EV entspricht bei konstanter Blende einer Verdoppelung der Belichtungszeit (von  $1/250$  s auf  $1/125$  s).

Abb. 3.7.2 | Belichtungsreihe mit einer Schrittweite von  $\frac{1}{3}$  EV.

## <Ø> Motivkontrast, Objektkontrast, Beleuchtungskontrast

- *Motivkontrast* ist der aktuell gegebene Kontrast, der für die Belichtungssteuerung maßgeblich ist. Er wird bestimmt vom *Objektkontrast* und vom *Beleuchtungskontrast*.
- *Objektkontrast* ist der Kontrast, den ein Objekt „von sich aus“ aufweist, das heißt wenn die Beleuchtung an allen Stellen gleich ist. Der ► *Beleuchtungskontrast* beträgt dann null.
  - ▷ Ein Beleuchtungskontrast kann z. B. entstehen, wenn Lampen in der Reprofotografie das Objekt nicht gleichmäßig ausleuchten. Bei einem ebenen Objekt entsteht dadurch ein Gradient (Helligkeitsverlauf, ► 5.8, Seite 65)).
  - ▷ Ein starker Beleuchtungskontrast tritt auch bei Blitzlichtaufnahmen auf: Nahe Objekte/Personen sind stark beleuchtet, weiter entfernte nur schwach.

</Ø>

## 3.8 Weißabgleich (White Balance)

Die Farbtemperatur der Beleuchtungslichtart bestimmt die Motivfarbe, wie Abbildung 3.8.1 zeigt.

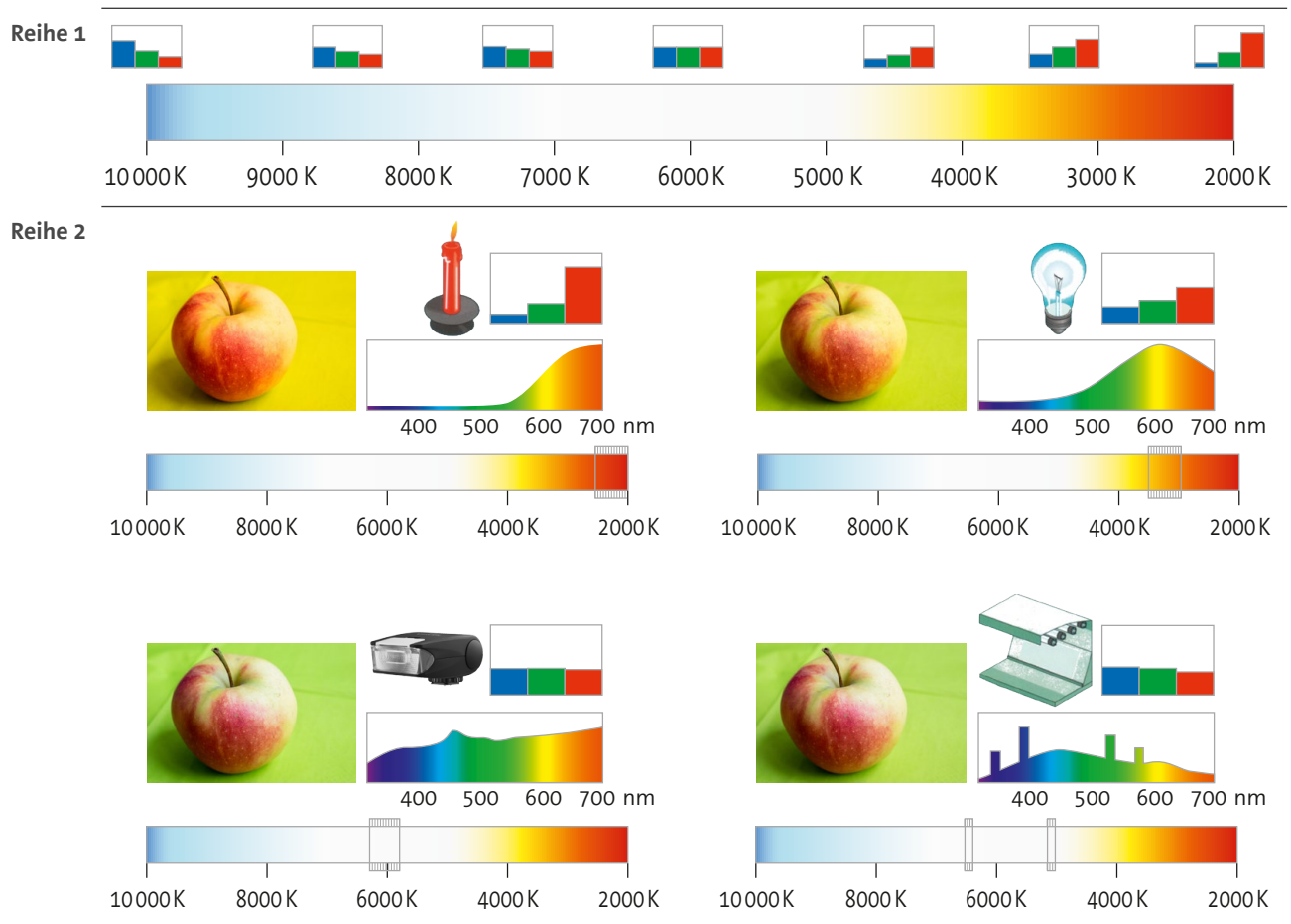


Abb. 3.8.1 | Einfluss der Farbtemperatur (◀ Seite 40) der Beleuchtung auf die Motivfarbe.

Oben (1): Zwischen etwa 4500 und 6500 K wird Licht annähernd als weiß empfunden.

Bei 4500 K ist bereits ein deutlicher Gelbanteil („warmes Weiß“) wahrnehmbar, oberhalb von 6500 K steigt der Blauanteil („kaltes Weiß“).

Unten (2): Jeweils dasselbe Objekt bei verschiedenen Farbtemperaturen: Kerzenlicht (links oben), Glühlampenlicht (rechts oben), Normlicht D<sub>50</sub> mit 5000 K (unten rechts) und Blitzlicht (ca. 5500 K) (unten links).

Um Farben korrekt wiederzugeben, muss die Farbtemperatur des Beleuchtungslichts berücksichtigt werden. Dies wird als *Weißabgleich* bezeichnet. Hierzu gibt es verschiedene Möglichkeiten:

- ▶ Man kennt die Farbtemperatur der Beleuchtung und stellt diesen Wert an der Kamera ein, z. B. Tageslicht, Schatten, Wolken, Leuchtstoffröhre, Glühlampe, Kerzenlicht. In der Kamera ist hinterlegt, wie die Farben für die einzelnen Beleuchtungsarten verändert werden müssen, damit sie zutreffend ausgegeben werden.
- ▶ Automatischer Weißabgleich: Prinzip dabei ist, dass die Kamera die hellsten flächigen Stellen im Motiv als Weiß bzw. Neutralgrau interpretiert. Enthält das Motiv tatsächlich reines Weiß bzw. neutrales Grau, sind die Ergebnisse im Allgemeinen gut. Ist die hellste Stelle aber z. B. Hellblau, entsteht durch die Automatik ein Gelbstich, denn die Farben werden so verändert, dass das Hellblau zu Weiß bzw. Neutralgrau wird.
- ▷ Hinweis: Nach demselben Prinzip funktioniert in Photoshop die Tonwertkorrektur mit der Option „dunkle und helle Farben suchen“. Wendet man diese Korrektur an, weist das Bild danach an der hellsten Stelle neutrales Weiß und an der dunkelsten Stelle neutrales Schwarz auf (Standardvorgaben für das Lichter- bzw. Tiefenfeld; ▶ Kapitel 5, Grundlagen der Bildbearbeitung, ab Seite 53)

- Manueller Abgleich mithilfe eines weißen Papiers oder – besser – einer Graukarte:  
Die Kamera kann anhand der Abweichung zwischen dem zu erwartenden Neutralgrau (oder Weiß) der Vorlage und den von der Kamera gemessenen Werten eine Farbkorrektur vornehmen. Vorsicht ist geboten, wenn das Papier optische Aufheller enthält, weil diese bei Beleuchtung mit UV-Licht für die Kamera bläulich erscheinen. Die Kamera rechnet diesen Blauanteil heraus, sodass ein Gelbstich resultiert.

Wie für die Ermittlung der optimalen Belichtung ermöglichen viele Kameras auch für den Weißabgleich das Erstellen von Reihenaufnahmen (Bracketing) mit verschiedenen Einstellungen.

- ⌘ Bei Mischlicht (z. B. Tageslicht plus Glühlampenlicht) versagt der Weißabgleich bzw. nur eine Beleuchtungsart kann berücksichtigt werden. Ausweg: Motivanteile in Bildbearbeitungsprogramm mit weicher Auswahlkante maskieren und für jedes Gebiet die Farbtemperatur korrigieren (kann nur näherungsweise gelingen).

## Farbtemperatur

Die Farbtemperatur (in Kelvin, K) charakterisiert mehr oder weniger weißes Licht. Sie entspricht derjenigen Temperatur, auf die ein *idealer schwarzer Körper* erhitzt werden muss, damit er Licht dieser Farbe abstrahlt.

- Die Angabe der Farbtemperatur sagt nur etwas über die *visuelle* Übereinstimmung einer Lichtquelle mit dem schwarzen Körper der zugehörigen Temperatur aus. Sie sagt nichts über die *spektrale Zusammensetzung* des Lichts eines Strahlers aus. Das heißt: Strahler mit unterschiedlichem Spektrum können die gleiche Farbtemperatur haben.
- Bei herkömmlichen Glühlampen entspricht die Angabe der Farbtemperatur in etwa der Temperatur, die der Glühdraht hat. Bei Nichttemperaturstrahlern (z. B. LEDs) spricht man (siehe voriger Punkt) von der *korrelierten* Farbtemperatur (CCT, correlated color temperature).

Tageslicht hat eine Farbtemperatur zwischen etwa 4500 K und 6500 K. Die weißen Normlichtarten D50 und D65 entsprechen einer Farbtemperatur von 5000 K bzw. 6500 K.

Abbildung 3.8.2 zeigt das Spektrum eines idealen schwarzen Körpers bei 5500 K.

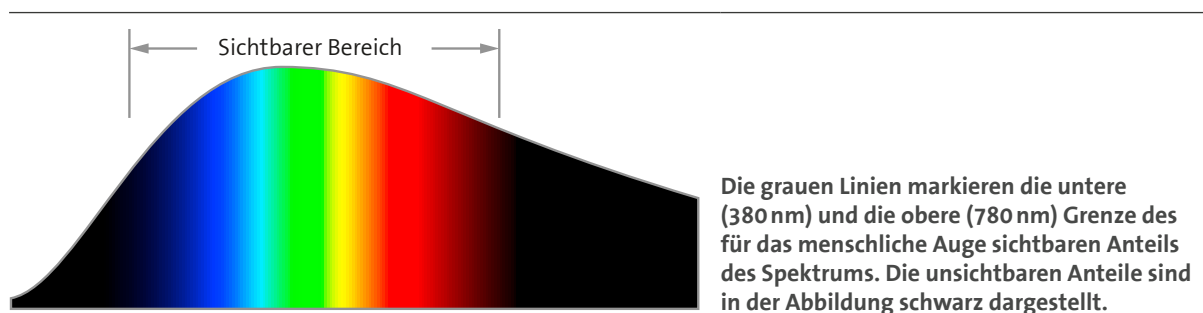


Abb. 3.8.2 | Spektrum der Strahlung eines idealen schwarzen Körpers bei 5500 K.

- ⌘ Hohe Farbtemperatur bedeutet *kalt*es Licht (und umgekehrt). „Kühler“ bewölkter Himmel hat eine Farbtemperatur von etwa 7000 K, während „warmes“ Kerzenlicht eine Farbtemperatur von rund 2000 K hat.

## <Ø> Idealer schwarzer Körper/Strahler

Der ideale schwarze Strahler ist ein Modell für die maximal mögliche Umwandlung von Wärme in elektromagnetische Strahlung. Die Strahlungsenergie eines erhitzten idealen schwarzen Strahlers hängt ausschließlich von seiner Temperatur [K] ab.

Abbildung 3.8.3 zeigt den Zusammenhang zwischen der spektralen Verteilung der abgegebenen Strahlungsenergie und der Temperatur. Mit steigender Temperatur verschiebt sich die Farbe des vom schwarzen Körper abgestrahlten Lichts (= Farbtemperatur) vom Infrarotbereich (IR) über Rot Richtung Blau und weiter in den Ultraviolettbereich (UV).

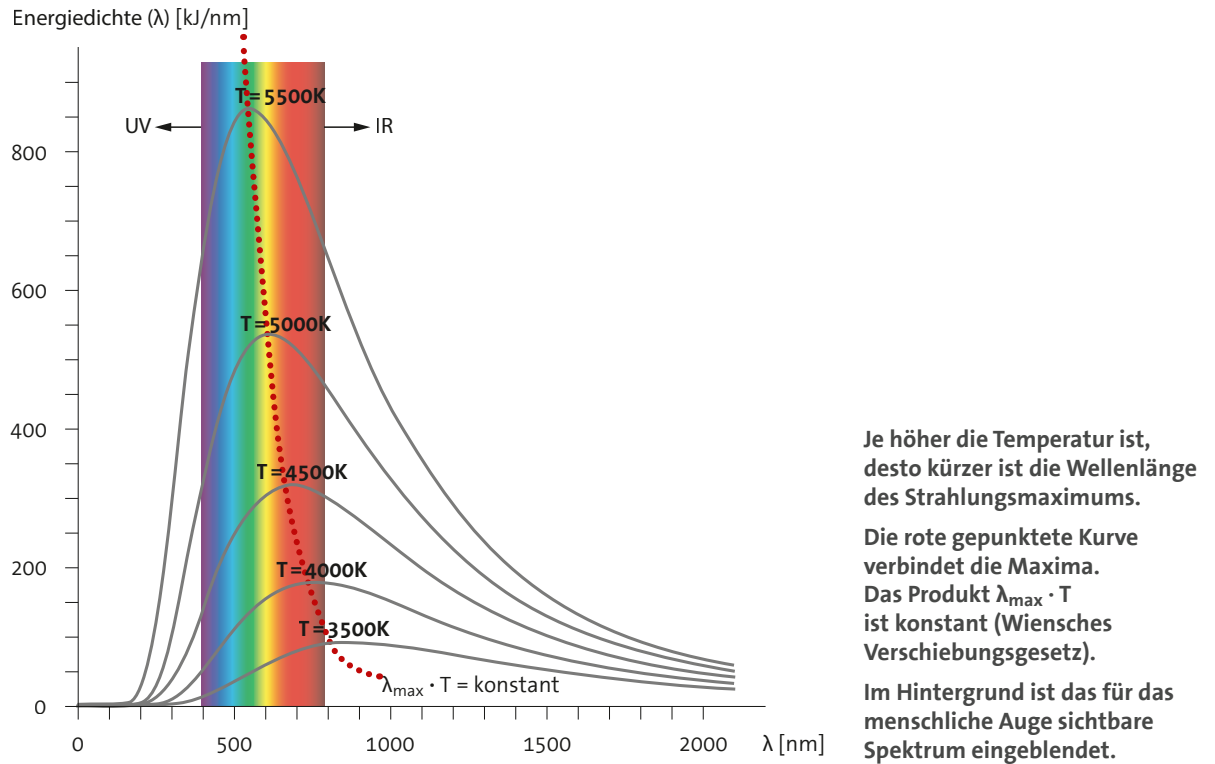


Abb. 3.8.3 | Abhängigkeit des Strahlungsspektrums des idealen schwarzen Körpers von der Temperatur [K].

</Ø>



## <!> Zusammenfassung „Lichttechnik in der Fotografie“

- Für Sammellinsen lautet die Linsengleichung (Abbildungsgleichung):

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g} \quad (\text{nicht Gegenstand der Prüfung GDVT})$$

- Die Brennweite eines Objektivs bestimmt den Bildausschnitt und den Bildwinkel:  
kurze Brennweite = großer Bildwinkel (Weitwinkelobjektive),  
große Brennweite = kleiner Bildwinkel (Teleobjektive).
- Brennweitenangaben für Objektive werden im Allgemeinen auf das Kleinbildformat (24 x 36 mm) bezogen. Je kleiner die Sensorfläche ist, desto kleiner ist der Bildwinkel, d. h. desto größer ist die effektive Brennweite eines gegebenen Objektivs. Dies wird durch den Brennweitenfaktor (Verlängerungsfaktor) ausgedrückt. Er entspricht (bei identischem Seitenverhältnis) dem Quotienten der Bilddiagonalen der zu vergleichenden Objektive.
- Die Blende ist der Quotient aus der Brennweite  $f$  und dem Durchmesser  $d$  der Blendenöffnungsfläche. Sie wird als dimensionslose Blendenzahl  $k$  angegeben.
$$k = \frac{f}{d}$$
  - ▷ Je *größer*  $k$ , desto *kleiner* ist die *Öffnungsfläche* und desto weniger Licht trifft auf den Film/Sensor.  
Große Blendenzahl = kleine Blende (wenig Licht),  
kleine Blendenzahl = große Blende (viel Licht).
- Die ganzen Blendenstufen sind Potenzen von  $(\sqrt[3]{2})^n$  ( $n$  ist eine natürliche Zahl  $\geq 0$ ).  
Verkleinern der Blende um eine Stufe, z. B. von 5,6 auf 8 ( $5,6 \cdot \sqrt[3]{2} \approx 8$ ) halbiert die Lichtmenge und verdoppelt daher die nötige Belichtungszeit.
- Die Schärfentiefe ist der Bereich vor und hinter der Fokusebene, der vom Auge noch als scharf wahrgenommen wird. Er hängt bei gegebener Brennweite von der Größe der Blendenöffnung ab, wobei gilt: kleine Blende (= große Blendenzahl) ergibt große Schärfentiefe, große Blende (= kleine Blendenzahl) ergibt kleine Schärfentiefe. Der Einfluss der Brennweite auf die Schärfentiefe kann vernachlässigt werden, obwohl die Schärfentiefe von Weitwinkelobjektiven in der Praxis als größer empfunden wird als die von Teleobjektiven.
- Die Verschlusszeit (Belichtungszeit) ist die Dauer, während der Licht auf den Film/Sensor gelangt.
- Der Lichtwert verknüpft Blendenzahl und Belichtungszeit:
$$LW = 2 \cdot \log_2(k) - \log_2(t)$$

Er entspricht *nicht* der Lichtmenge bzw. der Belichtung.  
Gleiche Lichtwerte lassen sich durch unterschiedliche Kombinationen von Blendenzahl und Belichtungszeit erreichen (Lichtwerttabellen).
- Die Farbtemperatur  $K$  der Beleuchtungslichtart bestimmt die Farbe eines Gegenstands. Die Farbtemperatur leitet sich vom Licht ab, das der ideale schwarze Strahler abhängig von seiner Temperatur abgibt. Für den Weißabgleich einer Kamera muss dieser Wert berücksichtigt werden.
  - ▷ Je *höher* die Farbtemperatur, desto *höher* ist der Blauanteil  
( $\Rightarrow$  Licht wird als „*kalt*“ empfunden).

## 4 Digitalisieren von Bildern

### Übersicht

#### Prinzip der Bilddigitalisierung

Aus einer analogen Vorlage (z. B. einer Freihandskizze, einem Fotoabzug, einem Dia) wird mithilfe eines Scanners oder einer Digitalkamera eine Bilddatei erstellt. Dazu wird mithilfe einer Optik ein 2-D-Abbild auf dem Sensor des Scanners oder der Kamera erzeugt. Der Sensor wandelt das eintreffende Lichtsignal in ein analoges elektrisches Signal um, das vom Analog-Digital-Wandler (AD-Wandler) in ein digitales Signal umgewandelt wird.

#### Hinweis:

Als Sensoren kommen CCD- und CMOS-Elemente zum Einsatz.

CCD steht für „Charge Coupled Device,“

CMOS für „Complementary Metal Oxide Semiconductor“.

Auf Vor- und Nachteile der Sensortypen, z. B. Blooming-Verhalten, wird hier nicht eingegangen.

Beim Digitalisieren einer analogen Vorlage geht generell Information unwiederbringlich verloren:

- ▶ Der Sensor weist eine begrenzte räumliche Auflösung auf. Die kleinste Bildeinheit (Pixel) fasst die Information aller darauf projizierten analogen Informationen in einer einzigen Information zusammen. Die Pixelgröße wird auch als Samplegröße bezeichnet.
- ▶ Die für jedes Pixel (= Sample) gespeicherte Information hat ebenfalls eine begrenzte Auflösung, d. h. sie kann nur bestimmte Werte (= Quantisierungsstufen) annehmen.

Abbildung 4.0.1 simuliert den beschriebenen Informationsverlust.

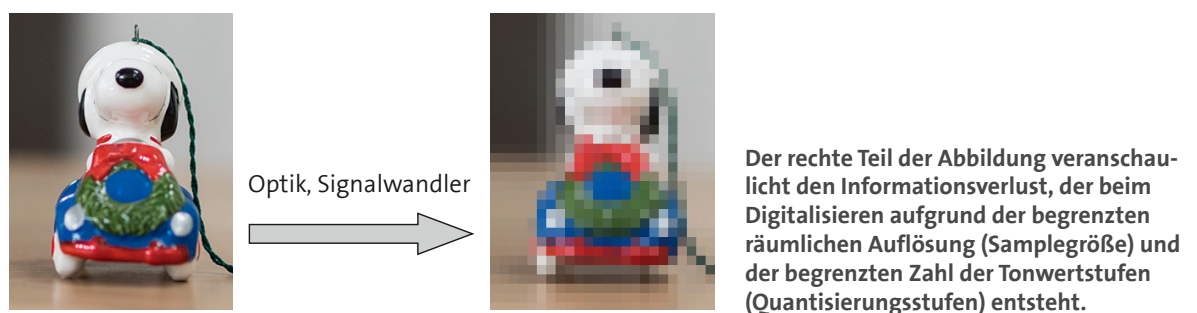


Abb. 4.0.1 | Aus der analogen Vorlage entstehen mithilfe einer Optik und eines Signalwandlers digitale Bilddaten.

### 4.1 Grundlagen der Analog-Digital(AD-)Wandlung

- Diskretisierung und Quantisierung
- Quantisierungs (Abtast-)theorem
- Technische Grenzen der Digitalisierung, Störquellen
- Bedeutung der Pixeltiefe (Bittiefe) beim Kodieren

### 4.2 Technische Größen und Begriffe

- Bildauflösung
- Tonwert im digitalen Bild
- Tonwert im gedruckten Bild, Lichtfang
- Optische Dichte, Dynamik

### 4.3 Scannertypen

## 4.1 Grundlagen der Analog-Digital(AD-)Wandlung

*Hinweis:*

Einige der nachfolgend beschriebenen Schritte beim Digitalisieren gelten sowohl für Bild- als auch für Audiosignale. Darauf wird im Text hingewiesen.

**Prüfungsrelevant** sind nur die Ausführungen zur Digitalisierung von Bildern.

### 4.1.1 Diskretisierung und Quantisierung

Jede analoge Quelle – ein Audiosignal ebenso wie eine Fotografie oder jede andere reale Bildvorlage – sind *kontinuierlich* hinsichtlich ihrer

- ▶ zeitlichen bzw. räumlichen Dimension (zeitkontinuierlich bzw. flächen- oder raumkontinuierlich) sowie
- ▶ hinsichtlich der Werte (wertekontinuierlich).  
Wertekontinuierlich bedeutet, dass z.B. der Schalldruck, den eine Tonquelle zu einem bestimmten Zeitpunkt erzeugt, oder die Helligkeit eines Bildmotivs an einer bestimmten Stelle jeden beliebigen Wert annehmen können.

⌘ Diskretisierung ist die Unterteilung kontinuierlicher analoger Ausgangsgegebenheiten in diskrete (und damit in ihrer Zahl begrenzte) Schritte.

■ Für ein Audiosignal ist das die Festlegung,

- ▷ wie oft der Spannungswert z. B. am Ausgang eines Mikrofons pro Zeiteinheit analysiert wird (= *Sampledauer* bzw. *Samplingfrequenz*; bei der Compact Disk beträgt die Samplingfrequenz 44,1 kHz, bei modernen Studioaufnahmen 192 kHz);
- ▷ wie viele Schritte (= *Quantifizierungsstufen*) für die AD-Wandlung des pro Sample gemessenen Spannungswerts zur Verfügung stehen (Bittiefe); bei der Audio-CD sind das 16 bit pro Audiokanal, d. h. jedem Sample kann beim *Quantisieren* einer von  $2^{16} = 65\,536$  Werten zugewiesen werden (Studiomaster: 24 bit, d. h.  $2^{24} = 16\,777\,216$  Quantisierungsstufen).

*Hinweis:*

Quantisieren bezeichnet den Vorgang, bei dem ein analoges Messsignal einer der verfügbaren Quantisierungsstufen zugeordnet wird.

■ Für ein Bild bedeutet Diskretisierung:

- ▷ Festlegen der Größe der Flächenelemente (Pixel), in die das Bild zerlegt wird. Daraus ergibt sich die Auflösung, z. B. 304 Pixel pro Inch (ppi). Pixel sind in der Druckvorstufe im Allgemeinen quadratisch, können aber auch ein anderes Seitenverhältnis haben.
- ▷ Festlegen, wie viele Quantisierungsstufen bei der AD-Wandlung der an der Sensorzelle gemessenen Spannung verfügbar sind. Bilddaten werden häufig mit 8 bit pro Farbkanal (Rot, Grün und Blau) ausgegeben. Intern (d. h. in der Kamera bzw. im Scanner) wird mit 10, 12, 14 oder 16 bit pro Farbkanal quantisiert ( $3 \times 16 \text{ bit} = 48 \text{ bit}$  für die drei Farbkanäle zusammen; ➤ optische Dichte, Dynamik, Seite 50).

⌘ Samplegröße (zeitliche bzw. räumliche Auflösung) und Bittiefe (Zahl der Quantisierungsstufen) bestimmen wesentlich die Genauigkeit und damit die Qualität einer Digitalisierung.

Die Abbildungen 4.1.1.1 und 4.1.1.2 zeigen Diskretisierung und Quantisierung eines Audio-signals bzw. eines Bildes.

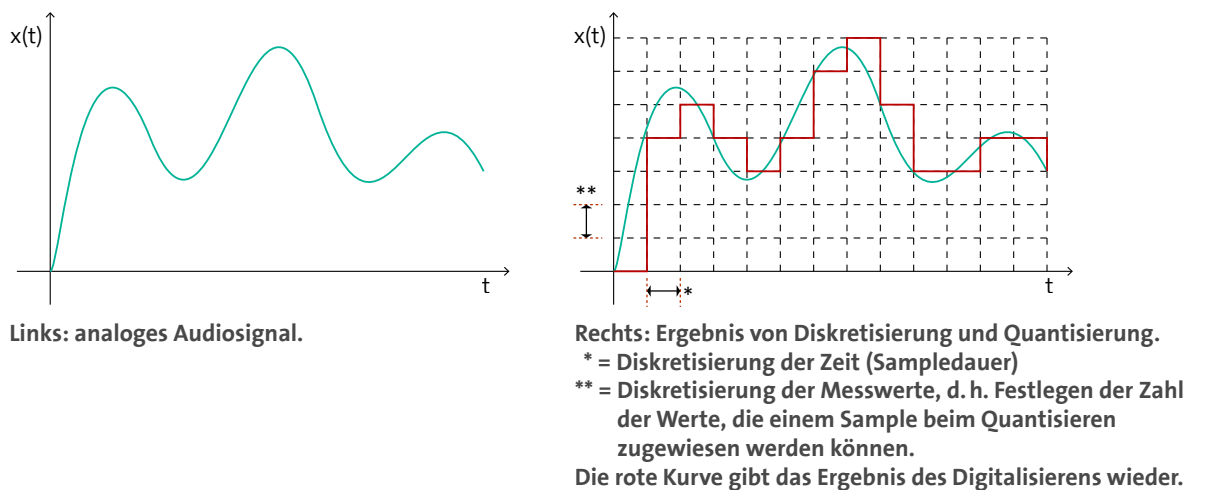


Abb. 4.1.1.1 | Diskretisierung und Quantisierung eines Audiosignals.

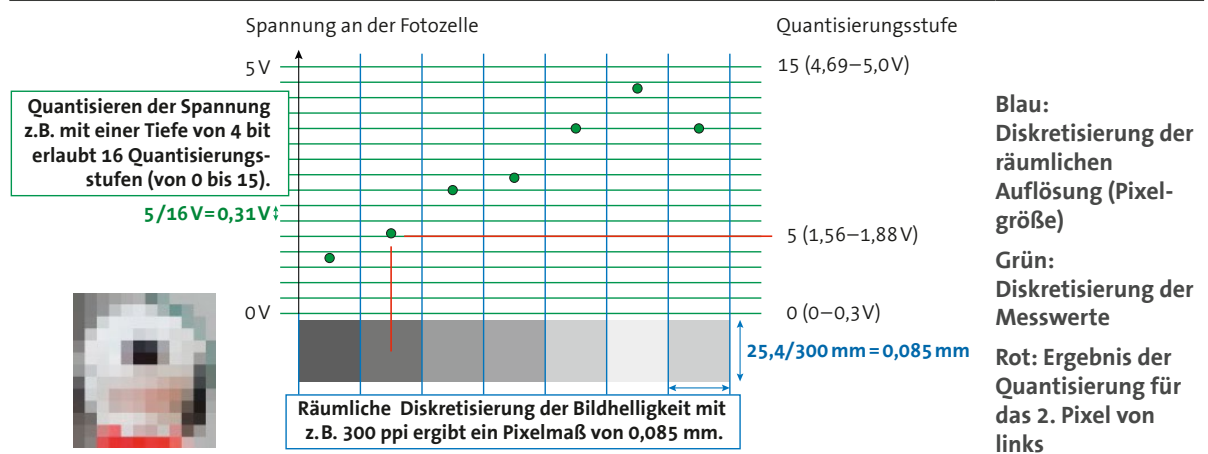


Abb. 4.1.1.2 | Diskretisierung und Quantisierung eines Bildes.

## 4.1.2 Quantisierungs (Abtast-)theorem

Das Quantisierungstheorem beschreibt den Zusammenhang zwischen der Samplegröße und der Rekonstruierbarkeit des analogen Signals bei der Digital-Analog-Wandlung (DA-Wandlung), z. B. bei der Wiedergabe einer Audio-CD:

- Wird ein kontinuierliches Signal, das die obere Grenzfrequenz  $f_{\max}$  hat, mit einer Abtastrate (Samplingfrequenz) von mindestens  $2 \cdot f_{\max}$  erfasst, kann das analoge Ursprungssignal eindeutig aus dem digitalen Signal rekonstruiert werden.

Im Beispiel der Audio-CD heißt das: Bei der Abtastrate von 44,1 kHz können Frequenzen bis maximal 22,05 kHz aufgezeichnet und wiedergegeben werden.

- Übertragen auf die Digitalisierung eines Bildes bedeutet das, dass die Rasterfrequenz, mit der ein Bild gedruckt werden soll, die nötige Scanauflösung bestimmt.

Beispiel:

Wird mit einem 60er-Raster gedruckt (60 Linien pro cm = 152 Linien pro inch), muss die Scanauflösung mindestens doppelt so hoch sein, d.h. 304 Pixel pro inch.

Hinweis:

In manchen Scanprogrammen wird der Faktor 2 aus dem Abtasttheorem folgendermaßen umgesetzt: Man wählt die Zielgröße des Bildes, die Rasterfrequenz des Drucks und den „Qualitätsfaktor“. Letzterer sollte für einen hochwertigen Druck 2 betragen.

### 4.1.3 Technische Grenzen der Digitalisierung, Störquellen

#### Aliasing, Jitter, Blooming

Beim Digitalisieren können verschiedene technisch bedingte Probleme auftreten.

■ Bei Audiosignalen gehören dazu unter anderem *Aliasing* und *Jitter*.

- ▷ Aliasing entsteht, wenn das zu quantisierende Signal Frequenzen enthält, die höher als die halbe Samplingfrequenz sind (CD: über 22,05 kHz). Da diese Signalanteile bei der Rückwandlung (d.h. von Digital nach Analog) nicht eindeutig zugeordnet werden können (siehe Abtasttheorem), entstehen Stellvertreterfrequenzen (Alias = Stellvertreter) die extrem störend wirken. Daher müssen vor der AD-Wandlung aus einem Audiosignal alle Frequenzen, die höher als die halbe Samplingfrequenz sind, zuverlässig gefiltert (Antialiasingfilter) werden.
- ▷ Jitter entsteht, wenn die Dauer einzelner Samples auch nur minimal vom vorgegebenen Takt („clock“) abweicht. Da diese beim Quantisieren aufgetretene Abweichung der Samedauer bei der Rückwandlung nicht bekannt ist, tauchen im analogen Ausgangssignal Frequenzen auf (sog. Modulationsseitenbänder), die im Original nicht vorhanden sind und ebenfalls erheblich stören können.

✂ Fehler durch Aliasing und Jitter lassen sich durch Nachbearbeitung des digitalen Signals nicht eliminieren, weil diese Fehler nicht als solche erkennbar sind. Das bedeutet, dass zum Beispiel Antialiasingfilter vor der AD-Wandlung zuverlässig Frequenzen oberhalb der zweifachen Samplingfrequenz eliminieren müssen.

■ Beim Digitalisieren von Bildern kann es ebenfalls zu Störungen durch Aliasing kommen. Eine weitere Störungsquelle ist das *Blooming*.

- ▷ Auch beim Digitalisieren von Bildmaterial kann durch Unterabtastung (d.h. zu niedrige Auflösung beim Scannen) stark störendes Aliasing entstehen. Ein bekanntes Beispiel ist eine Form von Moiré, die beim Scannen bereits gerasterter Vorlagen auftritt. Ein solches Moiré entsteht, wenn die Scanauflösung nicht mindestens das Doppelte der Rasterfrequenz der gedruckten Vorlage beträgt (Abbildung 4.1.3.1). Es lässt sich z. B. durch Weichzeichnen nicht entfernen.



Eine gerasterte Vorlage (Zeitschrift) wurde links mit 100 ppi gescannt, rechts mit 400 ppi.

Beide Scans sind nachträglich leicht weichgezeichnet (Radius an die Auflösung angepasst) worden. Trotzdem lässt sich das Moiré in der linken Aufnahme *nicht mehr* entfernen. Es fällt vor allem im Bereich des blauen Himmels auf, der sonst keine Zeichnung enthält.

Abb. 4.1.3.1 | Moiré durch Aliasing bei zu niedriger Scanauflösung für eine gerasterte Vorlage.



#### *Hinweis:*

Dieses bereits im Scan vorhandene Moiré darf nicht mit dem Moiré verwechselt werden, das beim erneuten Rastern einer gescannten gerasterten Vorlage (gilt jeweils für amplitudenmodulierte Raster, ► Rastertechnik, Seite 118) entsteht. Dieses beim erneuten Rastern entstehende Aliasing kann z.B. durch vorherige (!) Weichzeichnung verhindert werden, denn Weichzeichnung entspricht dem Ausfiltern hoher Frequenzen, die an scharfen Übergängen (Kanten der Rasterpunkte) vorliegen.

- Blooming entsteht, wenn nicht belichtete CDD-Zellen in der Umgebung stark belichteter Zellen mit angeregt werden. Ursache: Lichtsensoren sind häufig aus einer Fläche von CCD-Zellen aufgebaut. Eine einzelne CCD-Zelle kann pro Zeiteinheit nur eine begrenzte Lichtmenge verarbeiten. Fällt mehr Licht ein, dann wird der Überschuss an die Nachbarzellen verteilt. Die entsprechende CCD-Zelle und ihre Nachbarn erscheinen als überbelichtet. CMOS-Sensoren weisen ein deutlich geringeres Blooming auf.

### 4.1.4 Bedeutung der Pixeltiefe (Bittiefe) beim Kodieren

Neben der Auflösung (Samplegröße) entscheidet die Bittiefe über die Qualität eines Scans bzw. einer Aufnahme mit einer Digitalkamera. Die Bittiefe bestimmt, wie viele verschiedene diskrete Werte kodiert werden können.

Abbildung 4.1.4.1 zeigt ein Graustufenbild mit 8, 7, 4, 2 und 1 bit Datentiefe.



Abb. 4.1.4.1 | Graustufenbild mit 8, 7, 4, 2 und 1 bit Farbtiefe (entsprechend 256, 128, 16, 4 und 2 Graustufen).

Mehr als 256 Graustufen kann das Auge unter konstanten Bedingungen nicht unterscheiden, meistens sogar deutlich weniger. In Abbildung 4.2.4.1 ist zwischen 8 und 7 bit kaum ein Unterschied zu erkennen, zwischen 7 und 4 bit (mittleres Bild) erst bei genauerem Hinsehen.

#### *Hinweis:*

Auch wenn das Auge oft weniger als 256 Graustufen (8 bit) unterscheiden kann, ist es sinnvoll, Bilddaten in möglichst hoher Bittiefe zu halten. Stärkere Tonwertkorrekturen führen bei 8 bit und weniger zwangsläufig zu Tonwertabrissen, die deutlich sichtbar sein können. (► Bildbearbeitung, ab Seite 53)

Bei Farbbildern im RGB-Modus (je ein Kanal für Rot, Grün und Blau) stehen im 8-bit-Modus  $8 \text{ bit} + 8 \text{ bit} + 8 \text{ bit} = 24 \text{ bit}$  (16,7 Millionen Farben) zur Verfügung.

Gute Kameras und Scanner arbeiten intern mit mehr als 8 bit pro Farbkanal (10, 12, 14 oder 16). JPEG-Bilder werden immer mit 8 bit pro Kanal gespeichert, in Raw-Formaten ist die ursprüngliche Pixeltiefe (mehr als 8 bit) erhalten. TIFF-Bilder erlauben 16 bit/Farbkanal im CMYK-Modus und 32 bit/Farbkanal (HDR-Aufnahmen) im RGB-Modus.

(Zum Zusammenhang zwischen hardwareseitiger Pixeltiefe und kodierbarem Dichteumfang

► Seite 50.)

## 4.2 Technische Größen und Begriffe

### 4.2.1 Bildauflösung

Die Bildauflösung entspricht der Samplegröße. Sie wird in Pixel pro Inch (ppi) angegeben. Da Pixel eines *Scanners* bzw. einer *Kamera* in der Regel quadratisch sind, ist die Auflösung horizontal und vertikal gleich.

- Im *Webdesign* wird häufig nicht mit der Auflösung gearbeitet, sondern mit Pixelzahlen (z. B. 480 x 200 Pixel). Die Zahl der Pixel sagt nichts über die Auflösung, denn dazu muss die Pixelzahl zur Breite bzw. Höhe in Beziehung gesetzt werden. Im Webdesign ist das Rechnen mit absoluten Pixelzahlen aber sinnvoll, weil als Bezugsgröße die Bildschirmauflösung gilt.
- *Pixelpitch* (Monitor): Als Bildschirmauflösung gelten im Webbereich 72 ppi. Ein Bild, das 72 Pixel breit ist, müsste demnach auf einem Monitor mit einer Breite von 1 Inch (25,4 mm) dargestellt werden. Wie groß ein 72 Pixel breites Bild auf einem Monitor abgebildet wird, hängt aber vom Pixelabstand des Monitors ab (*Pixelpitch*, auch als *Dotpitch* bezeichnet, Dots per Inch [dpi]). Beträgt der Pixelpitch  $25,4/72 \text{ mm} = 0,35 \text{ mm}$ , entsprechen 72 Pixel tatsächlich 2,54 cm auf dem Monitor. Der Pixelpitch liegt heute je nach Modell aber zum Teil bei 0,27 oder 0,24 mm (bei 4K-Geräten noch niedriger), was einer Bildschirmauflösung von 94 bzw. 106 dpi entspricht. 72 Pixel ergeben dann nur 19,4 bzw. 17,3 mm auf dem Monitor.

### 4.2.2 Tonwert im digitalen Bild

Der Tonwert eines Pixels entspricht der Quantisierungsstufe und damit der Lichtmenge, die auf dem Sensor des zugehörigen Pixels angekommen ist.

- ▶ Die maximale Lichtmenge entspricht dem größten Tonwert.
- ▶ Die minimale Lichtmenge entspricht dem kleinsten Tonwert.

Wie viele verschiedene Tonwerte ein Farbkanal haben kann, hängt von der Pixeltiefe ab. Bei 8 bit sind es 256 Abstufungen. Ein Tonwert von 255 ist der hellste mögliche, ein Wert von 0 der dunkelste mögliche (Abbildung 4.2.2.1).

|          | Lichter | Vierteltöne | Mitteltöne | Dreivierteltöne | Tiefen |        |
|----------|---------|-------------|------------|-----------------|--------|--------|
| TW = 255 |         |             |            |                 |        | TW = 0 |

Abb. 4.2.2. 1 | Tonwertstufen in einem digitalen Bild, Farbkanal grün, 8 bit (die Farbkanaäle Rot und Blau sind auf 0 gesetzt).

### 4.2.3 Tonwert im gedruckten Bild

Beim gedruckten Bild ist der Tonwert ein Maß dafür, wie gedeckt (mit Druckfarbe) ein Bildbereich einem Normalbetrachter erscheint.

- ▶ Unbedruckte Flächen (Papierweiß) haben einen Tonwert von 0 %.
- ▶ Maximal gedeckte Flächen (Vollton) haben einen Tonwert von 100 %.

*Hinweis:*

Dies ist umgekehrt zum Tonwert in digitalen Bildern. Während die hellste Fläche dort den höchsten Tonwert besitzt, hat die hellste Fläche im Druck den niedrigsten Tonwert (Abbildung 4.2.3.1).

|        | Lichter | Vierteltöne | Mitteltöne | Dreivierteltöne | Tiefen |          |
|--------|---------|-------------|------------|-----------------|--------|----------|
| TW = 0 |         |             |            |                 |        | TW = 255 |

Abb. 4.2.3.1 | Tonwertstufen in einem gedruckten Bild, Farbkanal Cyan, 8 bit.



Im gedruckten Bild bestehen Flächen in der Regel aus Anteilen von Papierweiß und voll deckender Farbe (Rasterpunkte). Beim üblichen Betrachtungsabstand können die Rasterpunkte vom Auge nicht aufgelöst werden, sodass sich je nach Größe der Rasterpunkte ein Mischwert als Tonwert ergibt.

## Lichtfang (Light Gathering), Murray-Davis-Formel

- Der Tonwert einer bedruckten Fläche ist *nicht* ein geometrisches Maß für die Flächendeckung, sondern berücksichtigt die im Auge des Betrachters (und für ein Densitometer) *optisch wirksame* Flächendeckung.

Die optisch wirksame, mit einem Densitometer gemessene Flächendeckung ist je nach Tonwertbereich mehr oder minder deutlich größer als die geometrische. Einer der Gründe für diesen Tonwertzuwachs (*Dot Gain*) ist der sog. *Lichtfang*, Abbildung 4.2.3.2). Im Prinzip bedeutet Lichtfang, dass Licht, das in unmittelbarer Nähe zu einem Rasterpunkt auf das Papier trifft, teilweise in das Papier eindringt, wegen Streuung unter anderem Winkel zurückgeworfen und dabei vom Rasterpunkt „gefangen“ wird. Lichtfang tritt folglich nicht bei transparenten Druckstoffen oder Druckstoffen auf, in die das Licht nicht eindringen kann (z.B. Blech).

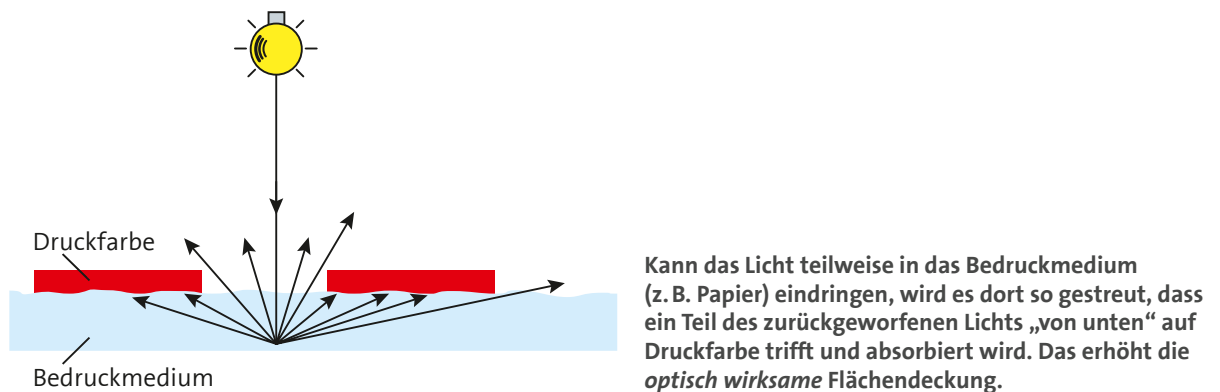


Abb. 4.2.3.2 | Lichtfang auf einem Druckmedium, in das das Licht teilweise eindringen kann, z.B. Papier.  
Quelle: mod. nach Kipphan, Handbuch der Printmedientechnik

- <Ø> Der Tonwert wird immer auf die Volltonfläche bezogen. Er wird nach der *Murray-Davis-Formel* aus der optischen Dichte (► 4.3.4) ermittelt. Eine gängige Darstellung beschreibt den *Tonwert A* folgendermaßen:

$$A [\%] = \frac{1 - 10^{-D_r}}{1 - 10^{-D_v}} \cdot 100$$

mit  $D_r$  als Dichte der zu beurteilenden Fläche und  $D_v$  als Dichte einer Volltonfläche.

Da der Lichtfang an den Rändern der Rasterpunkte auftritt, ist er bei feinem Raster stärker als bei grobem Raster, weil die Summe der Umfänge vieler kleiner Rasterpunkte bei gleicher Flächendeckung größer ist als bei großen Rasterpunkten (► Rastertechnik, Seite 118). (Herleitung: Die Fläche eines Kreises wächst mit den Quadrat des Radius, der Umfang linear. Ein Kreis mit dem Radius 1 mm hat eine Fläche von  $1 \text{ mm}^2 \cdot \pi = 3,14 \text{ mm}^2$  und einen Umfang von  $2 \cdot 1 \text{ mm} \cdot \pi = 6,28 \text{ mm}$ . Zwei Kreise mit der jeweils halben Fläche von  $1,57 \text{ mm}^2$  haben einen Radius von  $(\sqrt[3]{0,5}) \text{ mm} = 0,71 \text{ mm}$  und damit einen Umfang von jeweils  $2 \cdot 0,71 \text{ mm} \cdot \pi = 4,44 \text{ mm}$ , zusammen also  $8,88 \text{ mm}$ .)

</Ø>

- ⌘ Punktzuwachs beschreibt generell den Umstand, wenn bei einem Bearbeitungsschritt ein bestimmter Tonwert höher ausfällt als erwartet. Neben dem oben beschriebenen Lichtfang tritt Tonwertzuwachs u. a. beim Drucken auf (z. B. je nach Eindringverhalten der Druckfarben in das bedruckte Medium).

## 4.2.4 Optische Dichte, Dynamik

Trifft Licht auf eine Fläche, ist die Summe aus durchgelassenem (Transmission  $\tau$ ), reflektiertem bzw. remittiertem (Reflexion bzw. Remission\*  $\rho$ ) und absorbiertem (Absorption  $\alpha$ ) Anteil gleich 100 % bzw. gleich 1:

$$\tau + \rho + \alpha = 1$$

- Wird Licht zu 100 % durchgelassen, beträgt die Transmission 1 (Reflexion/Remission + Absorption = 0), wird es zu 100 % reflektiert/remittiert, beträgt die Reflexion/Remission 1 (Transmission + Absorption = 0). Im Allgemeinen ist keiner der drei Werte gleich null.

- Der Kehrwert der *Transmission* wird als *Opazität*  $O$  bezeichnet.  $O = \frac{1}{\tau}$   
Die Opazität (Maß für die Lichtundurchlässigkeit von Papier) liegt zwischen 1 und  $\infty$ .

- Optische *Dichte*  $D$  ist der Logarithmus der Opazität zur Basis 10:

$$D = \log_{10}(O).$$

$D$  liegt zwischen 0 und  $\infty$ .

$D = 0$  bedeutet volle Durchlässigkeit, ab etwa  $D = 5$  kann man von Undurchsichtigkeit sprechen, denn nur der  $1/10^5$  des einfallenden Lichts wird durchgelassen.

- ▷ Für bedrucktes Papier wird die Dichte nicht auf die Opazität (Kehrwert der Transmission) bezogen, sondern auf den Kehrwert des *Remissionsgrades*  $\rho$ .

Wird sämtliches Licht remittiert ( $\rho = 1$ , unbedruckte Stelle), entspricht das einer Dichte

$$D = \log_{10} \frac{1}{1} = 0.$$

Wird nur 1 % des Lichts remittiert ( $\rho = 0,01$ , bezogen auf eine unbedruckte Stelle), entspricht das einer Dichte

$$D = \log_{10} \frac{1}{0,01} = \log_{10} 100 = 2.$$

- Zusammenhang zwischen Dichteunterschied und Lichtwert

- ▷ (1) Die Verdopplung der Lichtmenge entspricht einem Kontrastverhältnis von 2 und damit einem Dichteunterschied  $\Delta D = \log_{10} 2 = 0,3010$ .
- ▷ (2) Doppelte Lichtmenge entspricht einer Verringerung der Blendenzahl um eine Stufe (z. B. von 8 auf 5,6).

Daraus folgt:

Ein Dichteunterschied  $\Delta D = 1$  entspricht ungefähr 3 Blendenstufen ( $3 \cdot \log_{10} 2 \approx 0,9$ ).

- Ein Maß für die Qualität von CCD- bzw. CMOS-Zellen ist der *Dichteumfang* (auch: Kontrastumfang, *Dynamik*), innerhalb dessen Tonwertunterschiede differenziert werden können.

- ▷ Aufsichtsvorlagen haben einen Dichteumfang von bis zu 1,8 (Offsetdruck) bzw. bis über 2 (Fotopapiere), der auch von einfachen Scannern bewältigt wird. Hochwertige Dias hingegen haben einen Dichteumfang von etwa 3 (etwa 10 Blendenstufen), Negativfilme von etwa 3,7 und Spezialfilme bis zu 4. Hochwertige Scanner bewältigen ein  $\Delta D$  von 4,2 und mehr. Das entspricht einer Dynamik von 14 Blendenstufen bei einer Kamera.

- ▷ Um Dias ohne Verlust zu digitalisieren, zählt nicht nur der Dichteumfang, d.h. der Abstand zwischen der höchsten und der niedrigsten Dichte, sondern auch die absolute Lage der minimalen Dichte. Der Grund: Diafilme haben ein Dichteminimum von etwa 0,2. Daher sollte ein Scanner schon ab diesem Wert Dichteunterschiede erfassen können.

- Um die Dynamik hochwertiger Sensorzellen nutzen zu können, muss der nachgeschaltete Analog-Digital(AD)-Wandler mehr als 256 Dichtestufen ( $> 8$  bit) quantisieren können.

*Erklärung:*

Das niedrigste Spannungssignal, das sich vom Rauschen der Sensorzelle unterscheiden lässt, entspricht  $D_{\max}$  (dunkelste Stelle,  $U_{\min}$  an der Sensorzelle). Besteht ein linearer Zusammenhang zwischen Lichtmenge und Spannung an der Sensorzelle, gilt, dass der maximale quantisierbare Spannungswert  $U_{\max}$  (hellste Stelle, entsprechend  $D_{\min}$ ) bei einem 8-bit-Wandler dem 256-Fachen des Spannungswerts  $U_{\min}$  (bei  $D_{\max}$ ) entspricht.

\* Papier weist keine *gerichtete* Reflexion auf, sondern eine *gestreute* (diffuse) Reflexion = Remission.

Daraus folgt für den Dichteumfang  $\Delta D$  eines 8-bit-Wandlers:

$$\Delta D = |\log_{10}(U_{\min}/U_{\max})| = \log_{10}(256/1) \approx 2,4$$

⇒ Ein 8-bit-Wandler kann den Dichteumfang hochwertiger Sensorzellen nicht verarbeiten.

*Rechenbeispiel:*

Welche Bittiefe muss ein AD-Wandler mindestens haben, wenn die Sensorzellen einen Dichteumfang von 4,2 bewältigen?

Lösung: Um den Dichteumfang des Scanners zu ermitteln, müssen zwei Angaben zu den lichtempfindlichen Zellen des Scanners bekannt sein:

(1) Wie hoch ist das Grundrauschen der Zellen (minimales Signal)?

(2) Wie groß ist der Potentialtopf der Zellen (maximales Signal)?

Für das Beispiel gelte: Grundrauschen 8 Elektronen, Potentialtopfgröße 120.000 Elektronen.

8 Elektronen entspricht der maximalen erfassbaren Dichte  $D_{\max}$  (dunkelste Stelle), 120.000 Elektronen der minimalen erfassbaren Dichte  $D_{\min}$  (hellste Stelle). Das Verhältnis von maximaler zu minimaler Dichte entspricht daher  $120.000/8 = 15.000$ .

Folglich gilt:  $\Delta D = \log_{10} 15.000 \approx 4,2$ .

Das bedeutet, dass für das Quantisieren und anschließende Kodieren mindestens 15.000 Stufen zu Verfügung stehen müssen (► Abb. 4.1.1.2, Seite 45).

Für die nötige Bittiefe des Wandlers ergibt sich:

10 bit erlauben 1024, 12 bit 4096, 13 bit 8192 und 14 bit 16.384 Abstufungen.

Der Wandler muss folglich mit einer Datentiefe von mindestens 14 bit arbeiten.

*Hinweise:*

Es handelt sich hier um die *hardwareseitige* Bittiefe des AD-Wandlers.

Die Tonwertstufen eines Signals mit mehr als 8 bit (z.B. 16 bit) können nach der AD-Wandlung durch *Mapping* auf ein 8-bit-Signal verteilt werden. Das entstehende 8-bit-Signal deckt zwar den ganzen von der Hardware erfassten Dichteumfang ab, aber bei der Reduktion auf 8 bit gehen zwangsläufig sehr viele Werte zwischen minimaler und maximaler Dichte verloren, denn: 16 bit lassen 65.536 Werte zu, 8 bit nur 256.

In den Datenblättern mancher Scanner wird als Dichteumfang der vom AD-Wandler differenzierbare Dichtebereich angegeben. Der optisch differenzierbare Dichtebereich kann (bei geringwertigen Sensorzellen) deutlich kleiner sein.

## 4.3 Scannertypen

Abbildung 4.3.1 zeigt die beiden wesentlichen Scannertypen.



Abb. 4.3.1 | Trommel- und Flachbettscanner.  
Quelle: Kipphan, Handbuch der Printmedientechnik

## Trommelscanner

Dieser Scannertyp hat über Jahrzehnte den High-End-Bereich dominiert. Hinsichtlich erzielbarer Auflösung und erfassbarem Dichteumfang ist er noch immer unübertroffen.

Prinzip ist, dass die Vorlage auf eine Trommel aufgezogen wird. Das durchgetretene (Dias) bzw. reflektierte/remittierte (Aufsichtsvorlagen) Licht wird von einem einzigen Photomultiplier erfasst. Die Trommel rotiert daher, damit der gesamte Umfang der Trommel eingelesen werden kann. Nach jeder Umdrehung macht der Photomultiplier einen kleinen Vorschub und erfasst auf diese Weise Umdrehung für Umdrehung die gesamte Vorlage.

## Flachbettscanner und Diascanner mit CCD-Zellen

Heute bewältigen auch hochwertige Scanner mit CCD-Zellen Dichteumfänge von 4,2 und höher. Da sich CCD-Zellen immer dichter packen lassen, werden Auflösungen im Bereich von 4000 ppi und höher erreicht. Alternativ lässt sich die Auflösung erhöhen, indem ein kleiner Teil des Vorlagenbereichs mithilfe einer Optik auf die CCD-Zeile vergrößert wird und die gesamte Vorlage aus einzelnen Teilflächen zusammengesetzt wird („Stitching“).

Die rasche Verbreitung hochwertiger Digitalkameras hat dazu geführt, dass Scannen heute in der Druckvorstufe immer seltener erforderlich ist.

## <!> Zusammenfassung „Digitalisieren von Bildern“

- Diskretisierung: Unterteilung kontinuierlicher analoger Ausgangsgegebenheiten in diskrete (und damit in ihrer Zahl begrenzte) Schritte:  
Festlegen der Größe der Flächenelemente (Pixel), in die das Bild zerlegt wird und  
Festlegen, wie viele Quantisierungsstufen bei der AD-Wandlung verfügbar sind.
- Quantisierungstheorem (= Abtasttheorem): Wird ein kontinuierliches Signal, das die obere Grenzfrequenz  $f_{\max}$  hat, mit einer Abtastrate von mindestens  $2 \cdot f_{\max}$  erfasst, kann das analoge Ursprungssignal eindeutig aus dem digitalen Signal rekonstruiert werden.  
⇒ Scanauflösung (Pixel pro Inch) soll das Doppelte der Rasterfrequenz betragen, mit der ein Bild gedruckt wird. Beim Scannen gerasterter Vorlagen *muss* die Auflösung das Doppelte der Rasterfrequenz betragen, mit der gedruckt wurde (andernfalls tritt *Aliasing* in Form eines nicht filterbaren Moirés auf).
- Im 8-bit-Modus stehen pro Farbkanal 256 Quantisierungsstufen (0–255) zur Verfügung.
- Tonwert digitales Bild: Der Tonwert eines Pixels entspricht der Quantisierungsstufe.  
Maximale Lichtmenge = größter Tonwert, minimale Lichtmenge = kleinster Tonwert.
- Tonwert gedrucktes Bild: Maß für die Deckung durch Druckfarbe.  
Unbedruckte Fläche: Tonwert = 0 %; maximal gedeckte Fläche: Tonwert = 100 %.  
Der Tonwert einer bedruckten Fläche ist *nicht* ein geometrisches Maß für die Flächendeckung, sondern entspricht der *optisch wirksamen* Flächendeckung (abhängig vom Lichtfang).
- Tonwertstufen werden traditionell in Lichter, Vierteltöne, Mitteltöne, Dreivierteltöne und Tiefen eingeteilt.
- Transmission  $\tau$  + Reflexion bzw. Remission  $\rho$  + Absorption  $\alpha$  = 100 % bzw. 1:  
 $\tau + \rho + \alpha = 1$
- *Opazität*  $O$  ist der Kehrwert der Transmission:  $O = \frac{1}{\tau}$
- Optische *Dichte*  $D$  ist der Logarithmus der Opazität zur Basis 10:  $D = \log_{10}(O)$ .
- Ein Maß für die Qualität von CCD- bzw. CMOS-Zellen ist der *Dichteumfang* (*Dynamik*).
- $\log_{10}(256/1) \approx 2,4$ . Ist die Dynamik größer als 2,4, muss der AD-Wandler mit höherer Datentiefe arbeiten.

## Übersicht

Von den Möglichkeiten, die Photoshop und andere Bildbearbeitungsprogramme wie GIMP (GNU Image Manipulation Program) bieten, sind für die Druckvorstufe nur wenige relevant:

- ▶ Globale Belichtungskorrekturen, z. B.:
  - ▷ Tonwertkorrektur (Tonwertspektrum, Histogramm, Gamma, ► Seite 55)
  - ▷ Gradationsänderung (► Seite 60)
  - ▷ Farbstichkorrekturen (► Seite 64)
- ▶ Globale geometrische Korrekturen z. B.:
  - ▷ Rotation (z. B. nicht optimal ausgerichtete Scanvorlage);  
*Beachte:*  
 Jede Rotation, die nicht in 90-Grad-Schritten erfolgt, geht auf Kosten der Bildschärfe.
  - ▷ Korrektur von perspektivischen Verzerrungen (z. B. stürzende Linien)
  - ▷ Skalierung auf Zielauflösung für Druckausgabe;  
*Beachte:*  
 Das Herunterrechnen (Downsampling) eines Bildes sollte *nach* Tonwertkorrekturen (► Seite 58) und *vor* dem Schärfen/Weichzeichnen erfolgen (► Seite 66).
- ▶ Globale Schärfekorrekturen
  - ▷ Weichzeichnen (► Abb. 5.9.1, Seite 66), z. B. zum Verhindern von Moiré beim erneuten Rastern bereits gerasterter Vorlagen.  
 (Alternative zum Verhindern von Moiré: Hochpassfilter, Demonstration in der Übung)
  - ▷ Unschärf maskieren („scharfzeichnen“; ► Abb. 5.9.2, Seite 66).

### Hinweis:

Die nachfolgenden Beschreibungen von Bildkorrekturen (z. B. Menübezeichnungen) beziehen sich auf Photoshop. Die Taste „Steuerung“ (Command) unter Windows entspricht für die hier gezeigten Tastaturkürzel der „Apfeltaste“ (⌘) unter Mac OS.

## 5.1 Bilddatenformate für die Druckvorstufe

Die am häufigsten in der Druckvorstufe verwendeten Formate für Pixelbilder sind:

TIFF (Tagged Image File Format, Dateisuffix .tif) und JPEG (Joint Photographic Experts Group, Dateisuffix .jpg). Auch das Photoshopformat (Dateisuffix .psd) ist verbreitet. Diese drei Bildformate können sowohl RGB-Daten als auch farbseparierte Bilddaten verarbeiten (CMYK).

Das JPEG-Format weist einige wesentliche Einschränkungen auf: Die Kompression ist generell verlustbehaftet, pro Farbkanal sind maximal 8 bit möglich und es können keine Alphakanäle eingebunden werden. Das Format JPEG 2000 erlaubt dagegen auch 16 bit pro Farbkanal sowie Alphakanäle. Zudem ist die Kompression und damit die erzielbare Qualität viel besser steuerbar als bei JPEG. Die Verbreitung in der Druckvorstufe ist aber bislang gering.

TIFF erlaubt 16 bit/Farbkanal (RGB: 32 bit) sowie Alphakanäle. Für die verlustfreie Kompression stehen u. a. LZW (Lempel-Ziv-Welch) und ZIP (Zipper = Reißverschluss) zur Verfügung. LZW erfordert eine Lizenz und kann daher nicht von allen Programmen gelesen werden (aber von allen CS-Programmen).

Hinweis: JPEG ab Qualitätsstufe 10 (von 12) kann auch für höherwertigen Druck verwendet werden. Für feine Raster (= hohe Rasterfrequenz) bzw. FM-Raster (► Rastertechnik, ab Seite 118) und sehr hochwertigen Druck werden Formate empfohlen, die eine verlustfreie Kompression erlauben.

## <Ø> Diskussionsanmerkung zu medienneutraler Datenhaltung

Medienneutrale Datenhaltung hat das Ziel, Bilddaten so zu speichern, dass sie für verschiedene Druckprozesse und auch für das Web verwendet werden können.

### ■ RGB oder CMYK?

Lange war es üblich, Bilddaten als CMYK-Daten zu speichern. Auch wenn Bilddaten nur für den Druck verwendet werden, spricht einiges dafür, Bilder als RGB-Daten zu speichern:

- ▷ Alle CMYK-Farbräume sind kleiner als z. B. der ECI-RGB-Farbraum. Farben, die außerhalb eines CMYK-Farbraums liegen, sind nach der Umwandlung abgeschnitten bzw. komprimiert (je nach Renderingintent) und verloren.
- ▷ Wo die Farbseparation im Workflow stattfindet, hat keinen Einfluss auf die Druckqualität, wenn die gleichen Farbprofile und dieselben Renderingintents angewendet werden. Für die Bildbearbeitung kann eine CMYK-Vorschau aktiviert werden.
- ▷ RGB-Daten können gleichermaßen für den Druck wie für das Web genutzt werden. Außerdem ist es möglich, die Daten für verschiedene Druckprozesse zu verwenden (Offset, Tiefdruck, unterschiedliche Rasterverfahren, unterschiedliche Papiersorten).

### ■ Welches Farbprofil?

- ▷ Die European Color Initiative (ECI, [www.eci.org](http://www.eci.org)) empfiehlt derzeit, für die Archivierung von Bilddaten das Farbprofil ECI-RGB v2 zu verwenden. Es kann aber auch mit anderen großen Farbräumen gearbeitet werden. Der Farbraum sRGB ist nicht geeignet, weil er nicht alle druckbaren Farben umfasst.

### ■ Grenzen der medienneutralen Datenhaltung

- ▷ Für den Druck werden Bilddaten meistens geschärft. Die optimale Schärfung hängt aber vom Druckprozess und der Zielgröße ab (► Globale Schärfekorrekturen, Seite 66). Umgehen lässt sich dieses Dilemma, indem ein Bild vor dem Schärfen in ein Smartobjekt umgewandelt wird. Der Schärfefilter ist dann wie eine Ebene ein- und ausblendbar (d. h. er ist zerstörungsfrei bzw. reversibel).

## 5.2 Mathematische Bildbearbeitungsoperationen

### Punktoperationen

Einzelne Pixel werden unabhängig von Nachbarpixeln verändert.

- Beispiel 1: schwellenwertbasierte Umwandlung vom Graustufenmodus (8 bit, Bytemap) in den S/W-Modus (1 bit, Bitmap)

$$TW_{\text{neu}}(x,y) = \begin{cases} 1, & \text{falls } TW(x,y) > \text{Schwellenwert} \\ 0, & \text{falls } TW(x,y) \leq \text{Schwellenwert} \end{cases}$$

Abbildung 5.2.1 zeigt die Umwandlung eines 8-bit-Bildes in ein Bitmap (1 bit).



Links: 8 bit (256 Graustufen)

Rechts: 1 bit;  
als Schwellenwert für die Tontrennung wurde 128 gewählt.

Abb. 5.2.1 | Schwellenwertbasierte Umwandlung eines 8-bit-Bildes (Bytemap) in ein 1-bit-Bild (Bitmap).

- Beispiel 2: Tonwertspreizung mit folgender linearer Transformation:

$$TW_{\text{neu}}(x,y) = \{ (TW(x,y) - TW_{\text{min}}) / (TW_{\text{max}} - TW_{\text{min}}) \} \cdot TW_{\text{neu,max}}$$



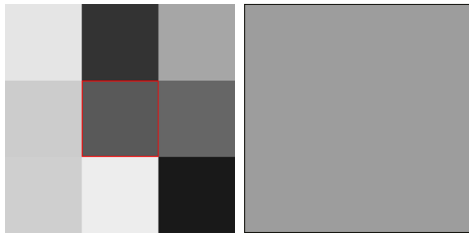
## Filteroperationen

Pixel werden neu berechnet, indem Bildinformationen von benachbarten Pixeln oder sogar aus dem gesamten Bild berücksichtigt werden.

- Beispiel: Weichzeichner, der den neuen Tonwert eines Pixels aus dem Mittelwert des Originalpixels (roter Rand in Abbildung 5.2.2) und den 8 Nachbarpixeln berechnet.

$$TW_{\text{neu}}(x,y) = 1/9 \cdot \sum_j \sum_i TW(j, i)$$

mit  $j=x-1 \dots x+1$ ;  $i = y-1 \dots y+1$



Der Tonwert für das rot umrandete mittlere Pixel (links) wird unter Berücksichtigung der 8 angrenzenden Pixel neu berechnet (Ergebnis rechts).

Abb. 5.2.2 | Anwendung eines Weichzeichners.

</Ø>

## 5.3 Histogramm/Tonwertkorrekturen

Das Histogramm stellt die Häufigkeitsverteilung der in einem Bild vorkommenden Helligkeitswerte dar (Aufruf: Steuerung-L). Das obere Feld (Tonwertspreizung) zeigt auf der X-Achse die Werte von 0 (minimale Helligkeit) bis 255 (maximale Helligkeit). Auf der Y-Achse (ohne Maßstab) ist die Häufigkeit der einzelnen Helligkeitswerte aufgetragen.

Das Beispiel in Abbildung 5.3.1 zeigt ein typisches Histogramm einer Landschaftsaufnahme mit automatischer Belichtungssteuerung unter unproblematischen Lichtverhältnissen.

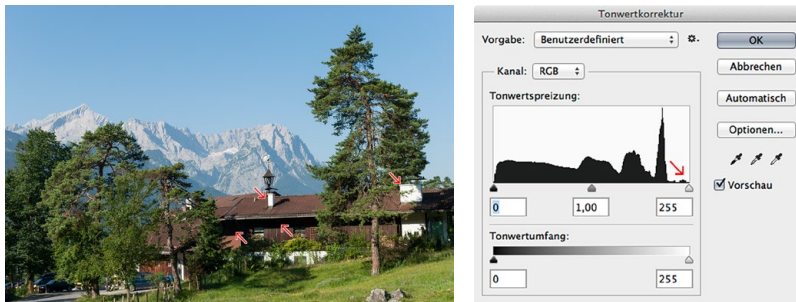


Abb. 5.3.1 | Histogramm einer unkomplizierten Landschaftsaufnahme mit Belichtungsautomatik. Im Histogramm kommen alle Tonwerte vor und es findet kein nennenswertes Clipping statt. Die hellsten Stellen sind die Vorderflächen der Kamine (↘, kleine Zacken am rechten Rand des Histogramms), die dunkelsten (↖) der Schattenstreifen unter dem Dach.

### Hinweis:

Clipping bezieht sich nur auf den linken und rechten Rand des Histogramms. Clipping bedeutet, dass die Häufigkeit der Tonwerte 0 und 255 nicht gegen 0 geht, sondern die Kurve abgeschnitten ist. Dies zeigt an, dass das Bild nicht alle im ursprünglichen Motiv vorhandenen Tonwerte umfasst. Nicht um Clipping handelt es sich, wenn Peaks im Histogramm *nach oben* beschnitten sind. Das Fenster ist zu niedrig, um den Peak komplett anzuzeigen (◻ Abb. 5.3.2).

Abbildung 5.3.2 veranschaulicht, dass es „das ideale“ Histogramm (und damit die „ideale Belichtung“) nicht gibt. In einem bei Tageslicht aufgenommenen Schnappschuss wie der Landschaft aus Abbildung 5.3.1 kommen üblicherweise alle Tonwerte vor. Ein helles Motiv (oberer Teil der Abb. 5.3.2) in diffuser Beleuchtung hat dagegen wenige Tiefen und viele Lichter (Highkey), während eine Aufnahme bei Dunkelheit (unterer Teil der Abb. 5.3.2) wenige oder keine hellen Passagen aufweist (Lowkey).



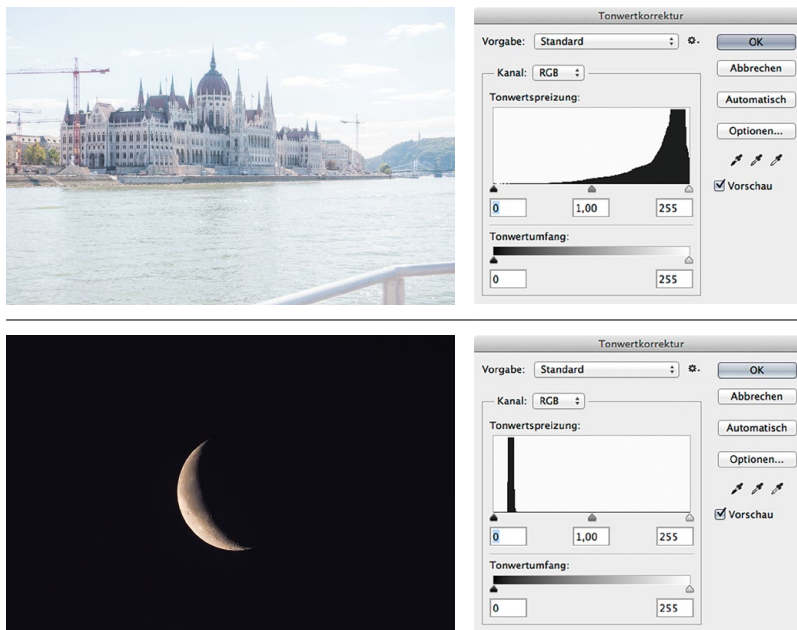
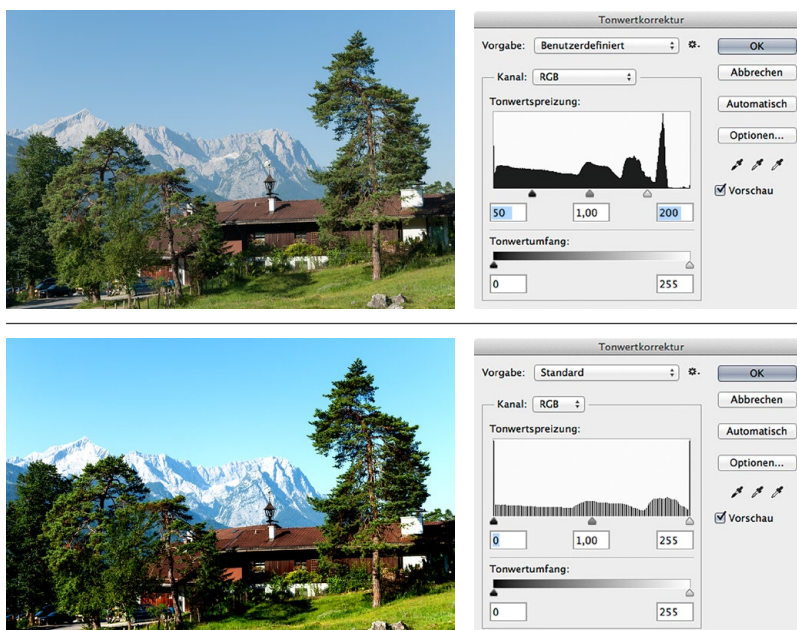


Abb. 5.3.2 | Histogramm einer Highkey-Aufnahme (oben) und einer Lowkey-Aufnahme (unten).

Ob Clipping akzeptabel ist, hängt vom Motiv ab. In einem Porträt sollte im Bereich des Gesichts keinerlei Clipping auftreten, d. h. weder Glanzstellen noch Schatten dürfen ausbrechen. Bei anderen Motiven stören kleine ausbrechende metallische Reflexpunkte (Spitzlichter) oder kleine zeichnungslose Schattenpartien dagegen weniger. Wegen des begrenzten Dynamikumfangs von Film/Sensor muss solches Clipping oft hingenommen werden, um die Helligkeitsbalance zu erhalten (z. B. Gegenlichtaufnahmen).

## Tonwertspreizung/Tonwertumfang

In dem Fenster „Tonwertkorrektur“ gibt es zwei Einstellbereiche: Tonwertspreizung (oben) und Tonwertumfang (unten). Abbildung 5.3.3 zeigt die Anwendung einer Tonwertspreizung.



Wendet man auf das linke Bild die rechts eingestellte Tonwertspreizung (50/200) an, erhält man das Bild unten links.

Das zugehörige Histogramm zeigt die Lücken (Tonwertabrisse), die durch das Abschneiden von insgesamt 105 Tonwerten (50 am linken Rand des Histogramms, 55 am rechten Rand) entstanden sind.

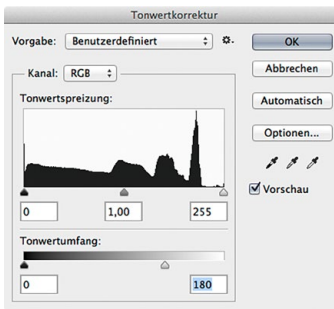
Abb. 5.3.3 | Tonwertspreizung.

Mit den beiden äußeren Anfassern der Tonwertspreizung kann man den unteren und oberen Grenzwert für eine Tonwertspreizung einstellen. Schiebt man z. B. (Abb. 5.3.3) den linken Regler auf 50 und den rechten auf 200 (linkes Bild), werden die zwischen 50 und 200 liegenden Werte so gespreizt, dass alles unter 50 den Wert 0 bekommt, alles über 200 den Wert 255. Öffnet man das Histogramm nach dem Anwenden der Tonwertspreizung erneut, sieht man die beiden hohen Peaks bei 0 und 255 und die durch das Spreizen der 150 verbliebenen Tonwerte bedingten Lücken im Histogramm.

Die Tonwertspreizung hat das ursprüngliche Bild stark verändert. Das kann künstlerisch gewollt sein, hat aber nichts mit dem Optimieren eines Bildes für den Druck zu tun.

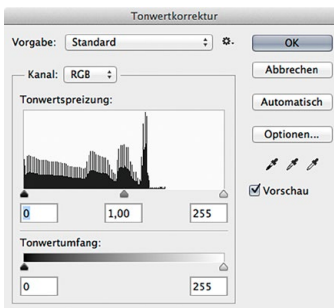
Ist ein Bild stark unterbelichtet, kann die Tonwertspreizung dies ausgleichen. In Abbildung 5.3.4 wurde Unterbelichtung simuliert, indem zuerst der Tonwertumfang der korrekt belichteten Aufnahme eingengt wurde. Das resultierende Histogramm des nun sehr dunklen Bildes zeigt, dass keine Tonwerte über 180 mehr vorhanden sind. Wendet man jetzt eine Tonwertspreizung an, durch die alles oberhalb von 180 auf den Wert 255 gesetzt wird, erhält die Aufnahme ihren ursprünglichen Tonwertumfang zurück. Allerdings zeigt das jetzige Histogramm die Lücken, die durch den Verlust der 75 Tonwertstufen zwischen 180 und 255 entstanden sind. Auch wenn dies hier im Beispiel kaum erkennbar ist, hat die zweimalige entgegengesetzte Tonwertkorrektur zu einem Qualitätsverlust geführt.

Reihe 1



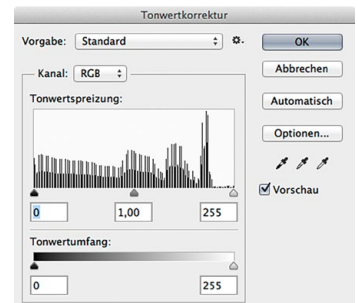
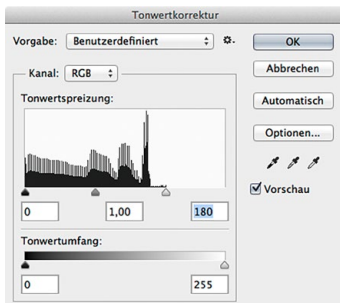
Unterbelichtung wird hier simuliert, indem der Tonwertumfang des linken Bildes nach oben auf 180 begrenzt wird (Ergebnis: Reihe 2).

Reihe 2



Das Bild erscheint jetzt deutlich unterbelichtet. Im Histogramm kommen keine Tonwerte über 180 vor.

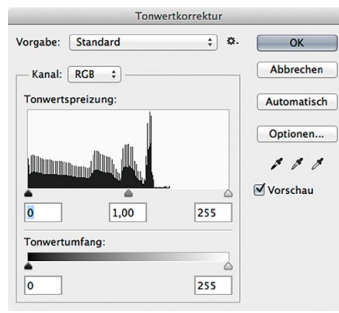
Reihe 3



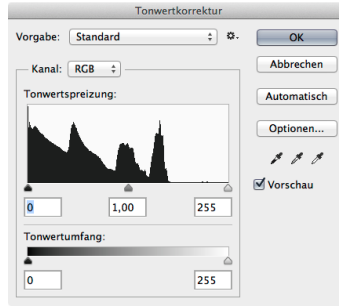
Wird jetzt eine Tonwertspreizung (unteres Bild links) angewendet, die alle Werte ab 180 auf 255 setzt, erhält das Bild seinen ursprünglichen Tonwertumfang zurück (unteres Bild Mitte). Das aktuelle Histogramm (unteres Bild rechts) zeigt aber die zu erwartenden Lücken, denn es fehlen die 75 Tonwertstufen zwischen 180 und 255.

Abb. 5.3.4 | Korrektur einer Unterbelichtung durch Tonwertspreizung.

Bei einer tatsächlich unterbelichteten Aufnahme würde das Histogramm am linken Rand Clipping zeigen. Simuliert man die Unterbelichtung der Landschaftsaufnahme im Raw-Konverter, (Abb. 5.3.5) weist das Histogramm das zu erwartende Clipping auf.



Simulierte Unterbelichtung durch Einengen des Tonwertumfangs (vgl. Abb. 5.3.4, Reihe 2).



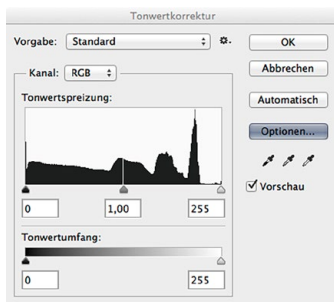
„Echte“ Unterbelichtung mit Clipping am linken Rand des Histogramms.

Abb. 5.3.5 | Histogramm einer von vorneherein unterbelichteten Aufnahme.

#### Hinweis:

Eine Tonwertspreizung lässt sich im RGB-Modus sehr gut kontrollieren, wenn schon vor dem Verschieben der Anfasser die Alt-Taste gedrückt und gehalten wird. Man sieht dann sofort, wo die hellsten bzw. dunkelsten Stellen im Bild liegen und wie stark sie beschnitten werden.

Die Tonwertspreizung ist auch automatisch möglich (Abb. 5.3.6).



Klickt man im Einstellfenster für Tonwertkorrektur (links) auf den Menüpunkt „Optionen“, werden die rechts gezeigten Möglichkeiten verfügbar.

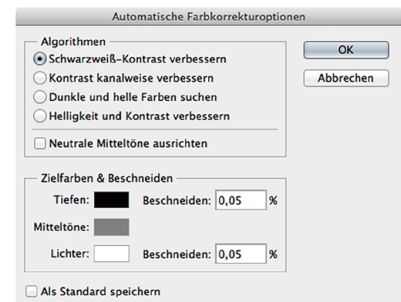


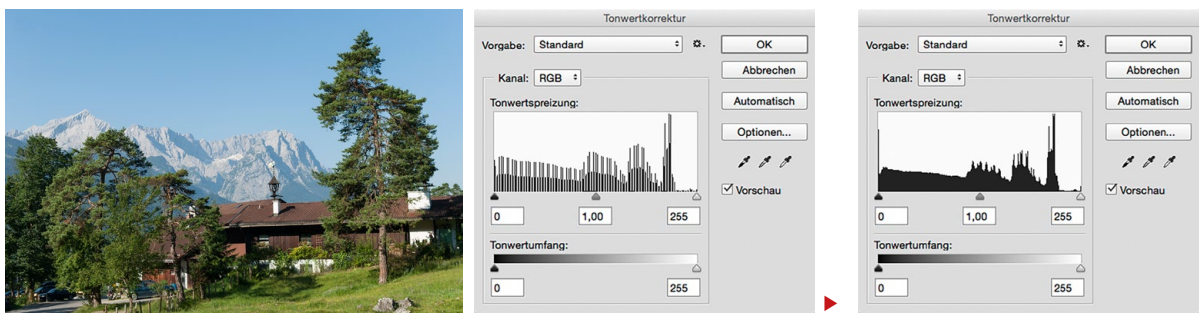
Abb. 5.3.6 | Menüpunkt „Optionen“ im Fenster Tonwertkorrektur.

Der Klick auf „Optionen“ öffnet ein weiteres Fenster. Um nur die Helligkeit ohne Farbänderungen einzustellen, den Punkt „Schwarzweiß-Kontrast verbessern“ wählen. Je höher die Grenzen für Beschneiden gewählt werden (im Beispiel 0,05 %), desto stärker wird das Clipping am jeweiligen Histogrammende. Wählt man „Helligkeit und Kontrast verbessern“, findet kein Clipping statt und es wird über das Gamma auch die Gradation des Bildes geändert. „Dunkle und helle Farben suchen“ stellt die dunkelste bzw. hellste Stelle im Bild auf die unter „Zielfarben & Beschneiden“ eingestellten Farben für Tiefen bzw. Lichter und erlaubt auch das Einstellen von Zielen für Farbkorrekturen im Mitteltonbereich (► 5.7, einfache Farbstichkorrekturen, Seite 64).

#### Hinweis:

Werden Tonwertkorrekturen ohne Einstellungsebene (► Seite 60) vorgenommen (d. h. mit irreversiblen Verändern der Originaldaten), sollte dies auf jeden Fall vor dem eventuellen Herunterrechnen der Auflösung für die Druckausgabe erfolgen. Durch das Neuberechnen beim Verkleinern lassen sich, wie Abb. 5.3.7 zeigt, Tonwertlücken abmildern.





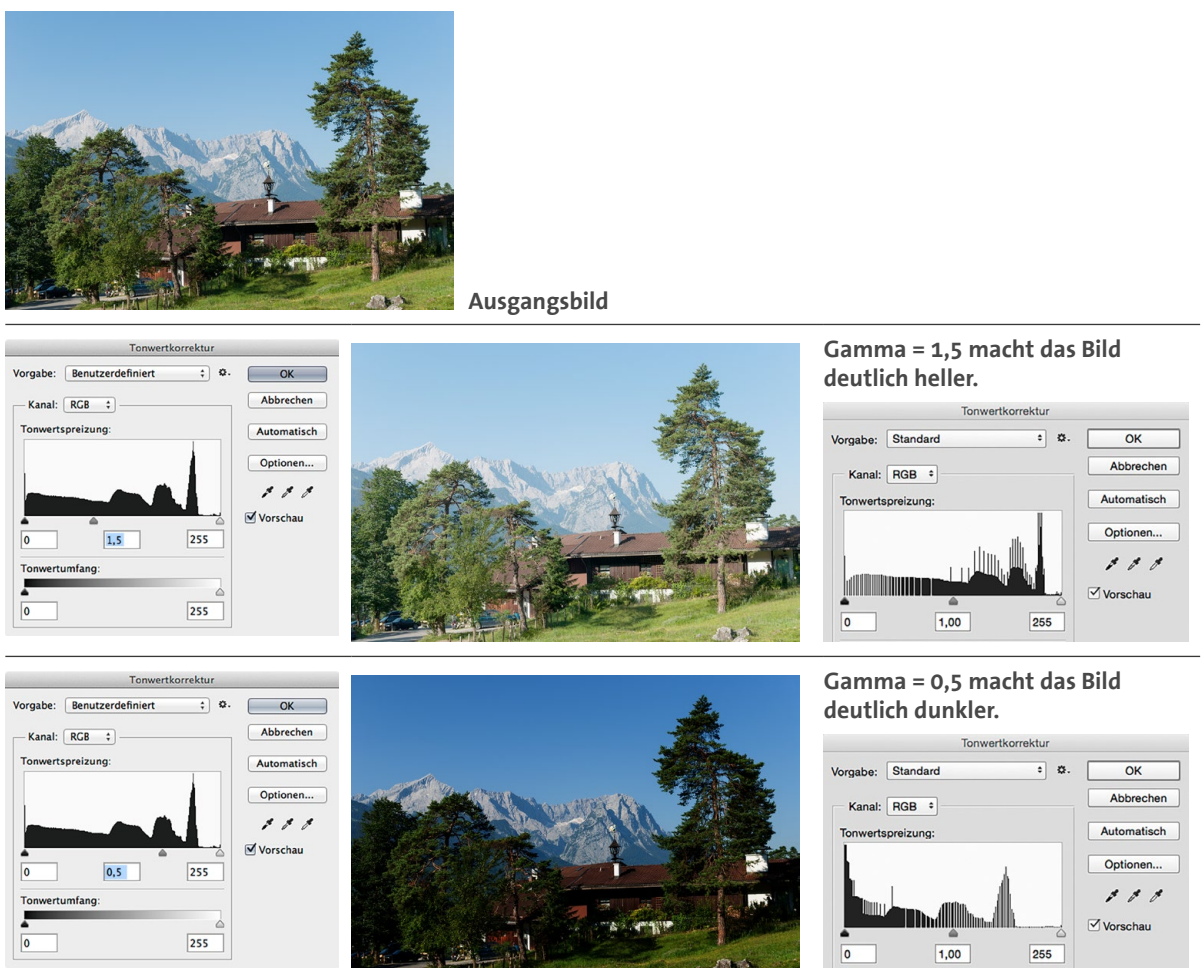
**Abb. 5.3.7 | Glätten eines Histogramms mit Tonwertabrisen durch Neuberechnen beim Verkleinern eines Bildes für die Zielauflösung. Ein Bild (links), das im Histogramm (Mitte) deutliche Lücken aufweist, zeigt nach Verringern der Auflösung auf 75 % („bikubisch schärfer [Verkleinerung]“) ein deutlich glatteres Histogramm ohne Lücken (rechts).**

## Das Gamma

Im Einstellfenster Tonwertkorrektur gibt es im Bereich Tonwertspreizung in der Mitte einen weiteren Anfasser bzw. ein weiteres Eingabefenster. Beim Öffnen des Fensters hat dieser Wert immer den Wert 1. Es handelt sich um den Gammawert der Gradationskurve (5.4, Gradationskurven, Seite 60). Er steuert die Steilheit der Gradationskurve:

- ▶ Ein Gamma = 1 bedeutet einen linearen Verlauf (Eingabe = Ausgabe).
- ▶ Ein Gamma < 1 bedeutet einen steilen Verlauf (Eingabe < Ausgabe, Abdunkelung).
- ▶ Ein Gamma > 1 bedeutet einen flachen Verlauf (Eingabe > Ausgabe, Aufhellung).

Abbildung 5.3.8 zeigt die Auswirkungen zweier unterschiedlicher Gammawerte.



**Abb. 5.3.8 | Tonwertkorrektur durch Ändern des Gammawerts. Ganz rechts jeweils das resultierende Histogramm.**

## Einstellungsebenen

Jede Tonwertkorrektur führt zum Verlust von Tonwertstufen. Bei 8-bit-Bildern kann das die Qualität sichtbar verringern, z. B. durch Tonwertsprünge in Verläufen (Abb. 5.3.7, 5.3.8). Werden Tonwertkorrekturen mithilfe von Einstellungsebenen (Abb. 5.3.9) vorgenommen, lassen sie sich ohne Qualitätsverlust rückgängig machen. Die Originaldaten bleiben unverändert (daher die Bezeichnung „zerstörungsfrei“) und die Tonwertkorrektur lässt sich nachträglich verlustfrei modifizieren, etwa durch Ändern der Deckkraft der Einstellungsebene.

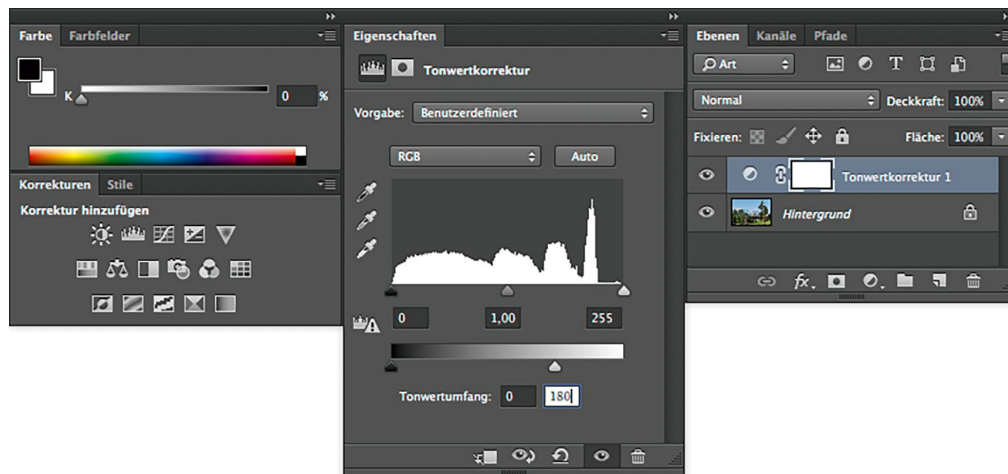


Abb. 5.3.9 | Tonwertkorrektur mithilfe einer Einstellungsebene.

In Einstellungsebenen können auch Masken angelegt werden. In Abbildung 5.3.10 ist die Wirkung der Tonwerteinschränkung auf 180 durch eine Maske mit Verlauf kontinuierlich bis auf null abgeschwächt. Wenn die Einstellungsebene im Ebenenmenü ausgewählt ist, erscheint sie im Kanälenmenü als Alphakanal, der frei bearbeitbar ist (► 5.8, Korrektur eines Beleuchtungscontrasts, Seite 65).

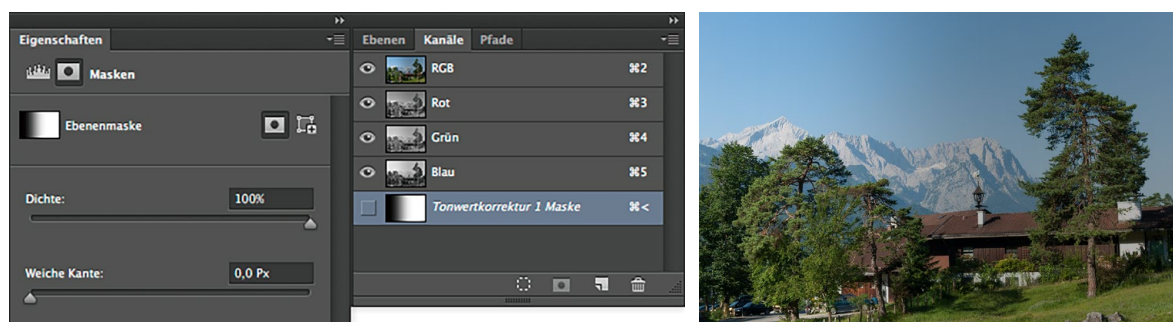


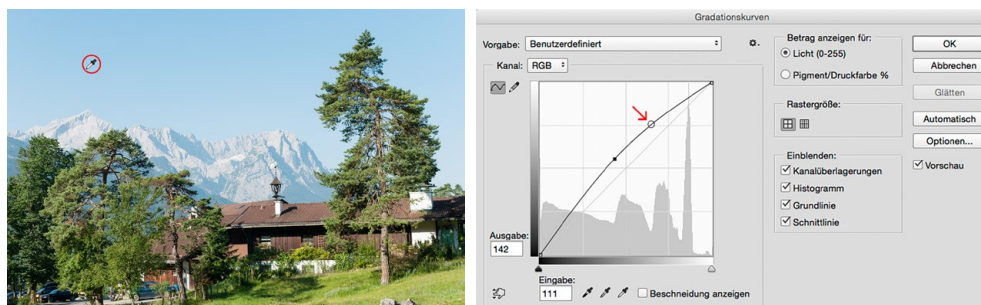
Abb. 5.3.10 | Tonwertkorrektur mithilfe einer Einstellungsebene, für die eine Ebenenmaske erstellt wurde. Die in der Kanälepalette ganz unten erkennbare Maske bewirkt einen Helligkeitsverlauf (Bild rechts)

## 5.4 Gradationskurven

Während der Gammawert in der Tonwertkorrektur nur eine pauschale Regelung von Helligkeit bzw. Kontrast erlaubt, ist mithilfe der Gradationskurve (Aufruf: Steuerung-M) eine gezielte Einstellung möglich, z. B. in den Mittel-, Viertel- oder Dreivierteltönen.

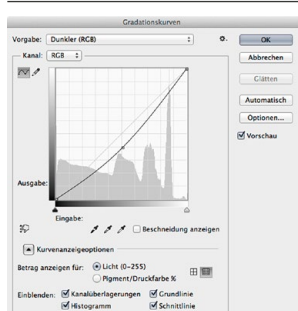
- ✂ Kontrasterhöhung in einem bestimmten Tonwertbereich geht immer zulasten des Kontrasts in anderen Helligkeitsbereichen.

Abbildung 5.4.1 zeigt das Dialogfenster „Gradationskurven“. Man kann wählen, ob die Kurve Helligkeitswerte von 0 bis 255 (wie in RGB) oder Pigmentwerte (wie in CMYK) anzeigt. Wölbung der Kurve nach oben bedeutet im ersteren Fall Aufhellung, im letzteren Abdunkelung. Die hellgraue Diagonale entspricht einer unveränderten Gradation (Eingabe = Ausgabe); klickt man bei geöffnetem Gradationsfenster auf eine Stelle im Bild, wird der Helligkeitswert angezeigt und ein kleiner Kreis markiert dessen Lage auf der Gradationskurve.

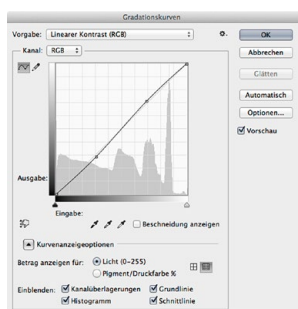


**Abb. 5.4.1 | Dialogfenster „Gradationskurven“.** Gewählt ist die Anzeige „Licht (0–255)“. Die nach oben gewölbte Gradationskurve bewirkt daher eine Aufhellung. Klickt man auf eine Stelle im Bild (rot umrandete Pipette im linken Bild), zeigt ein Kreis (roter Pfeil im rechten Bild) die Lage auf der Gradationskurve an.

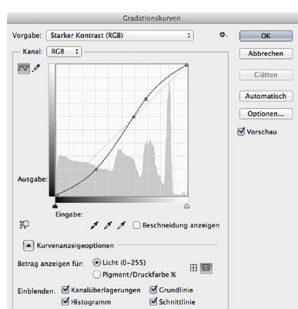
Abbildung 5.4.2 zeigt einige Presets für Gradationskurven aus Photoshop.



Gradationskurve „dunkler“



Gradationskurve „linearer Kontrast“



Gradationskurve „starker Kontrast“

**Abb. 5.4.2 | Voreingestellte Werte (Presets) aus Photoshop für typische Gradationskurvenänderungen.**



## <Ø> Vorsicht vor Tonwertkorrekturen im CMYK-Modus

- Größere Tonwertkorrekturen sollten generell *vor* der Farbseparation zu CMYK stattfinden, sofern man nicht überhaupt mit RGB-Bildern arbeitet (► RGB oder CMYK?, Seite 54).

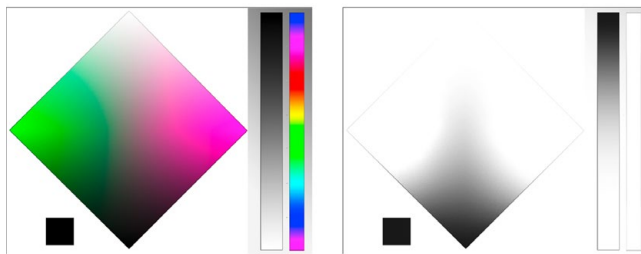
### Begründung:

In Druckprofilen entscheidet der Einsatz von Schwarz wesentlich über das Druckergebnis. Der relative Anteil von Schwarz ist nicht über den ganzen Tonwertbereich konstant, sondern im Allgemeinen bei hellen Tonwerten geringer. Bei Tonwertänderungen im CMYK-Modus werden aber alle Farbanteile proportional erhöht bzw. verringert. Besonders kritisch wirkt sich das in dem Bereich aus, in dem der Einsatz von Schwarz beginnt (Abb. 5.4.3).

#### ISO Coated v2

Gesamtfarbauftrag = 330%  
Maximales Schwarz = 95%  
Schwarzstart = 10%  
Schwarzbreite = 90  
Schwarzaufbau = GCRSmooth

Total Ink = 330%  
Maximum Black = 95%  
Black Start = 10%  
Black Width = 90  
Black Generation = GCRSmooth

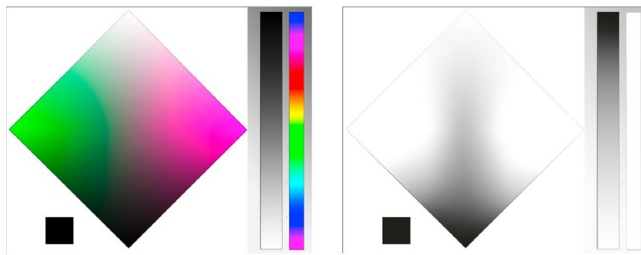


Das derzeit für den Bogenoffsetdruck auf gestrichenen Papieren weit verbreitete Farbprofil ISO Coated v2 verwendet im hellen Bereich (oberer Teil der rechten Raute) gar kein Schwarz (auch neutrale Töne werden nur aus CMY erzeugt). Nach unten zu dunkleren Bereichen setzt die Verwendung von Schwarz sanft ein, um den Grauanteil nicht nur durch CMY zu erzeugen, sondern zunehmend durch Schwarz zu ersetzen. Das Ersetzen von CMY-Grau durch Schwarz wird als Grey Component Replacement (GCR) bezeichnet. Der Gesamtfarbauftrag für die dunkelsten Stellen beträgt in diesem Farbprofil maximal 330%.

#### ISO Newspaper v4 26

Gesamtfarbauftrag = 230%  
Maximales Schwarz = 95%  
Schwarzstart = 0%  
Schwarzbreite = 100  
Schwarzaufbau = GCRMaxSmooth

Total Ink = 230%  
Maximum Black = 95%  
Black Start = 0%  
Black Width = 100  
Black Generation = GCRMaxSmooth



Im Zeitungsdruck (Profil ISO Newspaper v4 26) muss der Gesamtfarbauftrag verringert werden – hier auf 230 % – weil die Farbe sonst nicht rechtzeitig trocknet. Entsprechend stärker wird CMY-Grau durch Schwarz ersetzt, was hier als „GCRMaxSmooth“ bezeichnet wird.

Abb. 5.4.2 | Der Anteil von Schwarz beim Aufbau einzelner Farben ist nicht über den ganzen Helligkeitsbereich konstant. Tonwertkorrekturen im CMYK-Modus verändern daher den im Farbprofil festgelegten Schwarzaufbau.  
Quelle: European Color Initiative (ECI)

</Ø>



## 5.5 Helligkeits-/Kontrastregler

Die beiden Regler erlauben nur eine simple Einstellung von Helligkeit und Kontrast. Beide Regler sind so ausgelegt, dass ihre Wirkung schwächer wird, je näher ein Pixel dem Wert 0 bzw. 255 ist. Pixel mit den Werten 0 und 255 werden nicht verändert. Dadurch wird verhindert, dass Clipping entsteht.

Mit Anklicken des Kästchens „früheren Wert verwenden“ ändert sich diese Charakteristik und die Helligkeitswerte werden linear verschoben. Dies führt zu Clipping. Abbildung 5.5.1 zeigt die Auswirkung einer Helligkeitskorrektur.

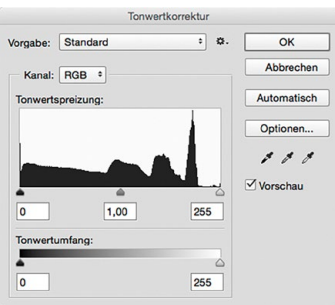
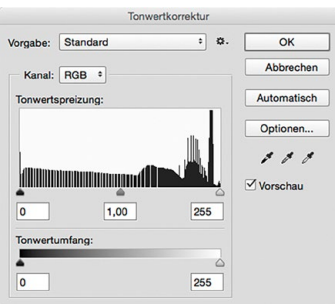
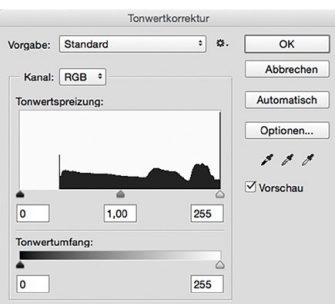
|                                                                                     |                                                                                      |                                                                                     |                                                                                                                                                                          |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |    |  | <b>Helligkeit um 50 erhöhen.</b>                                                                                                                                         |
|   |   |                                                                                     | <b>Ergebnis. Das Histogramm zeigt die erwarteten Lücken, aber kein verstärktes Clipping.</b>                                                                             |
|  |  |                                                                                     | <b>Wird das Kästchen „früheren Wert verwenden“ aktiviert, werden sämtliche Werte des Histogramms um 50 nach rechts verschoben. Die rechten 50 Werte fehlen komplett.</b> |

Abb. 5.5.1 | Wirkung des Einstellung von Helligkeit/Kontrast.

## 5.6 Filter „Tiefen/Lichter“

Mit diesem Werkzeug lassen sich Helligkeitsanpassungen in den Lichtern und Vierteltönen bzw. den Tiefen und Dreivierteltönen vornehmen, die mithilfe von Gradationskurven deutlich aufwendiger wären. Der Filter eignet sich sehr gut zum Aufhellen von Schattenpartien oder zum Abdunkeln mit Rekonstruktion von Zeichnung in sehr hellen Partien.

Drei Parameter sind einzustellen:

► Stärke

► Tonbreite:

Je kleinere Werte hier eingestellt werden, desto mehr beschränkt sich die Änderung auf sehr dunkle bzw. sehr helle Partien. Je größer die Tonbreite gewählt wird, desto mehr werden auch Tonwerte im Bereich mittlerer Helligkeit einbezogen.

*Faustregel:* Kleinere Tonbreite bedeutet geringere Tonwertkorrektur.

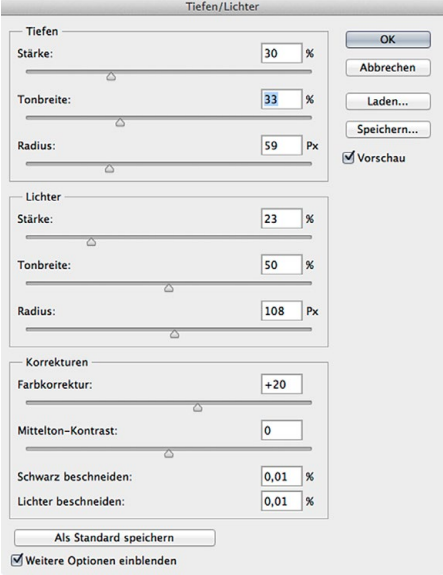
► Radius:

Dieser Wert legt die Größe des Bereichs fest, der beurteilt wird, um zu bestimmen, ob ein Bildbereich zu den zu den verändernden Tiefen bzw. Lichtern gehört. Kleiner Radius bedeutet, dass z. B. schon wenige dunkle Pixel genügen, um einen Bereich aufzuhellen, bei einem großen Radius bleiben solche Bildbereiche unverändert.

*Faustregel:* Kleiner Radius bedeutet ausgedehnte Veränderungen.

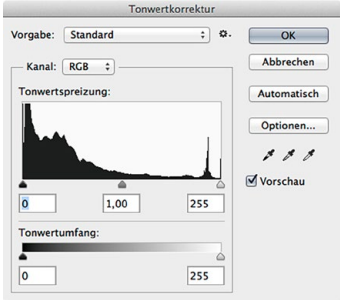
Abbildung 5.6.1 zeigt den Einsatz des Filters an einer Aufnahme mit sehr hellen Lichterbereichen und sehr dunklen Schattenpartien.

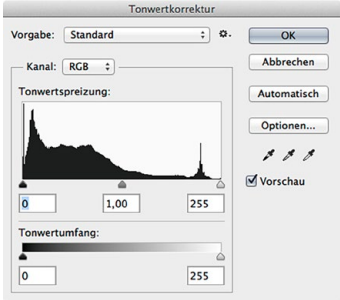




**Änderungen an der Aufnahme links oben durch Anwenden des Tiefen/Lichter-Filters mit nebenstehenden Einstellungen bewirkt Dämpfen der Lichter und Aufhellen der Schatten.**

**Da das Aufhellen der Tiefen und das Abdunkeln der Lichter auf Kosten des Mitteltonkontrasts geht, gibt es den Regler „Mittelton-Kontrast“, um diesem Effekt gegen-zusteuern.**





**Der Vergleich der Histogramme vorher (links) und nachher (rechts) zeigt, dass die Peaks an den Rändern etwas nach innen verschoben wurden.**

Abb. 5.6.1 | Wirkung des Filters „Tiefen/Lichter“

<Ø>

## 5.7 Einfache Farbstichkorrekturen

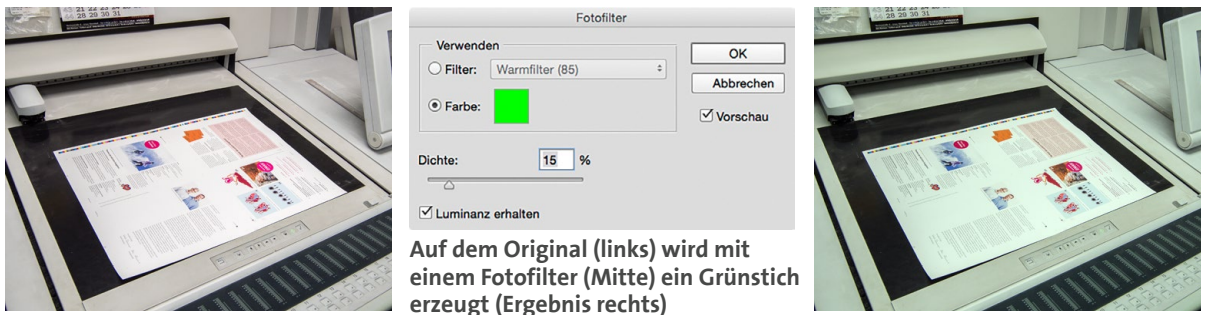
Zur groben Korrektur eines Farbstichs gibt es viele Möglichkeiten. Eine Auswahl:

- Im Menü „Farbbalance“ lassen sich Farbstiche nach Augenmaß sehr gut für Lichter, Mitten und Tiefen getrennt korrigieren (Abb. 5.7.1, Reihe 2).
- Automatikfunktion: Tonwertkorrektur (Steuerung-L)/Optionen/dunkle und helle Farben suchen (mit oder ohne Ausrichten neutraler Mitteltöne). Hierbei wird die hellste Stelle automatisch als Weiß angenommen, die dunkelste als Schwarz (Abb. 5.7.1, Reihe 3, linkes Bild).
- Pipettenwerkzeug: Mit der Tiefen-, Lichter- oder Mitteltonpipette auf einen Bildbereich klicken, der weiß, schwarz bzw. neutralgrau sein sollte. Doppelklick auf die jeweilige Pipette ermöglicht genaue Einstellung des Zielwerts (Abb. 5.7.1, Reihe 3, Mitte und rechtes Bild).



- Fotofilter: Warm- oder Kaltfilter einsetzen, um Farbtemperatur zu korrigieren. Wenn eine Stelle bekannt ist, die Weiß bzw. Hellgrau sein sollte: Farbe mit Pipette aufnehmen, im HSB-Farbmodell (Hue, Saturation, Brightness) die Komplementärfarbe einstellen (H-Wert + bzw. – 180°) und diesen Farbfiler anwenden (Abb. 5.7.1, Reihe 4).

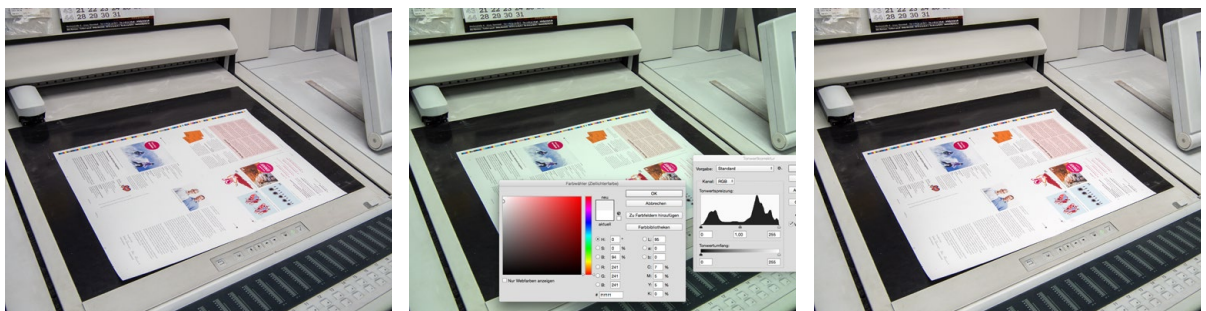
Reihe 1



Reihe 2



Reihe 3



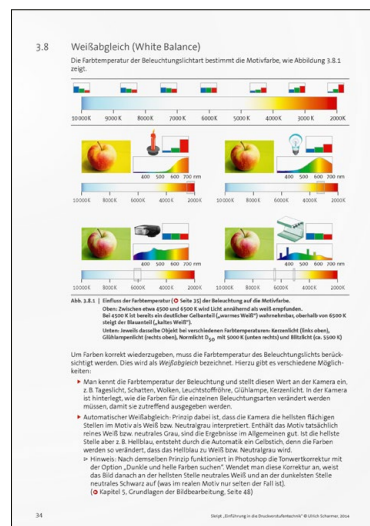
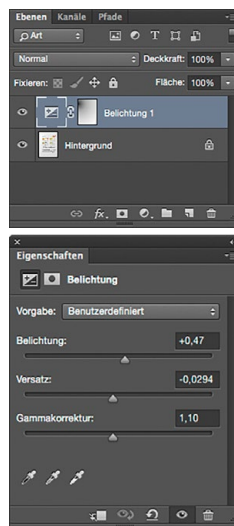
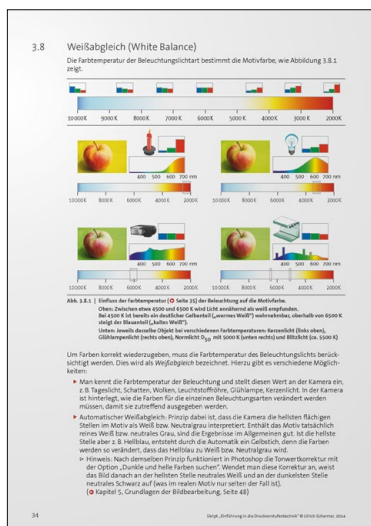
Reihe 4



Abb. 5.7.1 | Beispiele für das Erzeugen (Reihe 1) und Korrigieren (Reihen 2 bis 4) eines Farbstichs.

## 5.8 Korrektur eines Beleuchtungskontrasts

Ziel ist es, eine Maske anzulegen, deren Deckkraftverlauf dem (vermuteten bzw. mit der Pipette orientierend gemessenen) Beleuchtungsgradienten entspricht. Abbildung 5.8.1 zeigt das am Beispiel einer mit einem Beleuchtungskontrast abfotografierten Seite aus diesem Skript.



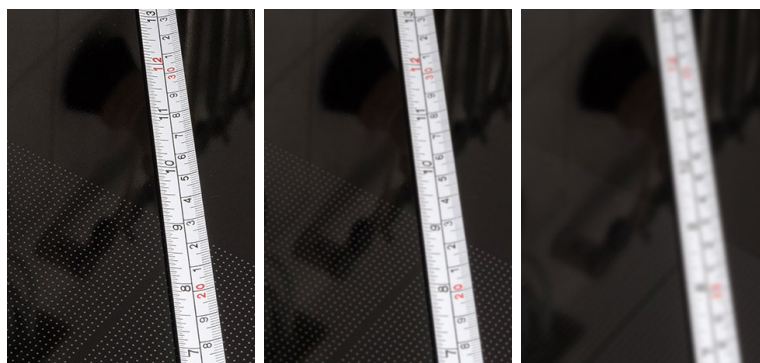
Die abfotografierte Seite (links) ist nicht gleichmäßig ausgeleuchtet, d. h. es besteht ein diagonal verlaufender Beleuchtungsgradient (Gradient): Links oben ist das Bild am hellsten, rechts unten am dunkelsten. Zunächst wird eine Einstellungsebene für „Belichtung“ erstellt und darin ein Verlaufsmaske angelegt (Mitte oben), die so gut wie möglich dem Helligkeitsverlauf im Foto entspricht. Dann werden die Eigenschaften der Einstellungsebene (Mitte unten) so eingestellt, dass das Foto überall gleich hell ist (Ergebnis rechts).

Abb. 5.8.1 | Korrektur eines Beleuchtungsgradienten.

</Ø>

## 5.9 Globale Schärfekorrekturen

Die folgenden Beispiele (Abbildungen 5.9.1 und 5.9.2) zeigen die Auflösungsabhängigkeit von Weichzeichnen und Unschärfmaskieren.

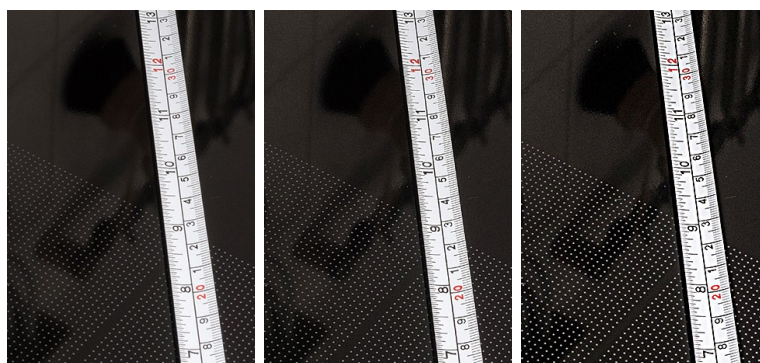


Links: Originalaufnahme.

Mitte: Auflösung 1000 dpi, Weichzeichnung mit einem Radius von 3 Pixel.

Rechts: Auflösung 300 dpi, ebenfalls Weichzeichnung mit einem Radius von 3 Pixel. Die Weichzeichnung wirkt sich bei gleicher Bildgröße deutlich stärker aus als im mittleren Bild.

Abb. 5.9.1 | Abhängigkeit der Effektstärke eines Weichzeichners von der Auflösung.



Links: Originalaufnahme.

Mitte: Auflösung 1000 dpi, Unschärfmaskierung mit einer Stärke von 150 % und einem Radius von 3 Pixel.

Rechts: Auflösung 300 dpi, ebenfalls Unschärfmaskierung mit einer Stärke von 150 % und einem Radius von 3 Pixel. Die Unschärfmaskierung wirkt sich bei gleicher Bildgröße deutlich stärker aus als im mittleren Bild.

Abb. 5.9.2 | Abhängigkeit der Effektstärke einer Unschärfmaskierung von der Auflösung.

Abbildung 5.9.2 zeigt, dass man beim Unschärfmaskieren nicht nur die Zielauflösung berücksichtigen muss, sondern diesen Filter generell richtig dosiert anwenden sollte. (Hinweis: Auch die Art des Druckprozesses spielt eine Rolle, [Offset, Inkjet usw.]).

## <!> Zusammenfassung „Bildbearbeitung für die Druckvorstufe“

### *Hinweis:*

Einige der in diesem Kapitel genannten Korrekturen werden in der Übung zu diesem Thema gezeigt.

#### ■ Bilddatenformate in der Druckvorstufe

- ▷ JPEG: verlustbehaftet, maximal 8 bit/Farbkanal, keine Alphakanäle, keine Ebenen. Die höheren Qualitätsstufen sind auch für den Druck geeignet, für hochwertigen Druck (feine Raster, frequenzmodulierte Raster) sollten aber verlustfrei komprimierte Daten verwendet werden.
- ▷ TIFF: verlustfreie Kompression möglich, 16 bit/Kanal (RGB: 32 bit), Ebenen und Alphakanäle unterstützt.
- ▷ Photoshop-eigenes Format: Wie TIFF plus weitere Merkmale.

#### ■ Datentiefe: 8 bit oder 16 bit

- ▷ Für den Druck genügen im Allgemeinen 8 bit/Farbkanal.
- ▷ 16 bit erlauben Tonwertkorrekturen ohne Gefahr von Tonwertabrissen (Lücken im Histogramm).
- ▷ Bilder in RAW-Formaten bzw. von guten Scannern liegen mit 16 bit Tiefe pro Farbkanal vor.

#### ■ Histogramm: grafische Darstellung der Häufigkeitsverteilung der in einem Bild vorkommenden Tonwerte. Es gibt Aufschluss darüber, ob

- ▷ alle Tonwerte vorkommen,
- ▷ helle (Highkey) oder dunkle Tonwerte (Lowkey) überwiegen,
- ▷ Clipping in den Lichtern und/oder Tiefen vorliegt. Clipping mit steilem Anstieg der Kurve zum linken und/oder rechten Rand des Histogramms hin bedeutet, dass Tonwerte abgeschnitten wurden und Bildbereiche ohne Zeichnung (Tonwert durchgehend 0 bzw. 255) existieren.

#### ■ Gradationskurven

- ▷ Eine Gradationskurve beschreibt, wie Tonwerte bei einem Prozess (z. B. Belichten/Entwickeln eines Films, Belichten auf Fotopapier) transformiert werden.
- ▷ Die Punkte auf einer Gradationskurve sind die Paare von Eingangs- (x-Achse) und Ausgangswerten (y-Achse).
- ▷ Bleiben alle Tonwerte unverändert, verläuft die Gradationskurve als Diagonale mit 45° durch den Koordinatenursprung.
- ▷ Der Gammawert steuert die Steilheit der Gradationskurve:
  - Gamma = 1 bedeutet einen linearen Verlauf (Eingabe = Ausgabe).
  - Gamma < 1 bedeutet einen steilen Verlauf (Eingabe < Ausgabe, Abdunkelung).
  - Gamma > 1 bedeutet einen flachen Verlauf (Eingabe > Ausgabe, Aufhellung).

#### ■ Einstellungsebenen erlauben zerstörungsfreie Tonwertänderungen.

- Weichzeichnen/Unschärf maskieren werden u. a. über die Eingabe eines Pixelradius gesteuert und hängen daher in ihrer Sichtbarkeit von der Ausgabegröße ab. Beide Korrekturen sollten daher erst vorgenommen werden, wenn die Ausgabegröße feststeht. Werden die Bildebenen zuvor in Smartobjekte umgewandelt, können Weichzeichnen/Unschärf maskieren zerstörungsfrei erfolgen und später an eine veränderte Ausgabegröße angepasst werden.



## 6 Grafikformate

### Übersicht

- 6.1 Einführung
  - Grafiktypen
  - Abgrenzung von Grafik, Bild und Text
- 6.2 Datenformate für Grafiken
  - Pixelformate
  - Vektorformate
- 6.3 Bézierkurven
- 6.4 Ausgabe von Vektorgrafik: Rendern

## 6.1 Einführung

### 6.1.1 Grafiktypen

Hierzu zählen: Logos, Wappen, Piktogramme, Cartoons, CAD-Zeichnungen, Flussdiagramme, Organigramme, Datenplots verschiedenster Form, Karten usw. (Abb. 6.1.1).

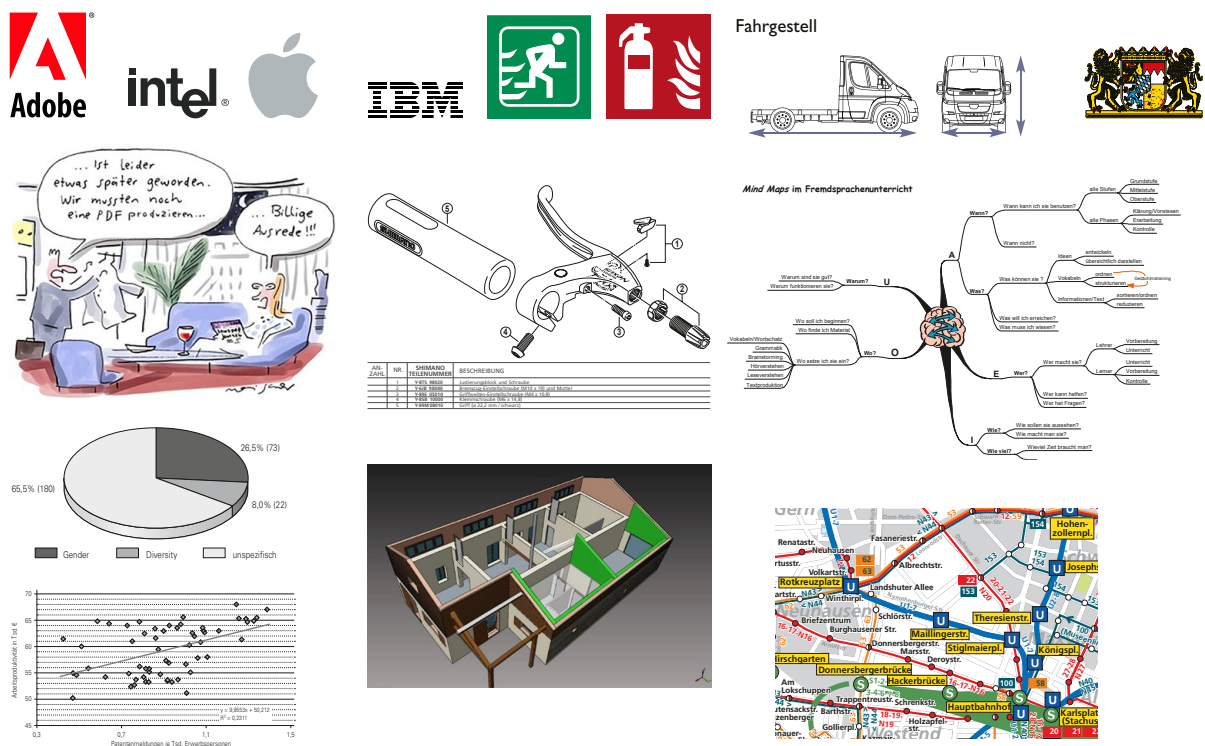


Abb. 6.1.1 | Verschiedene Grafiktypen.

Quellen: Adobe, Intel, Apple, IBM, VdS Schadenverhütung Verlag, Peugeot, Bayer, Staatsregierung, Cartoon: Dirk Meissner, Shimano, Wiebke Heuer/Max Hueber Verlag, IHF München, freecadweb.org, MVV



## 6.1.2 Abgrenzung von Grafik, Bild und Text

### <Ø> Historisch

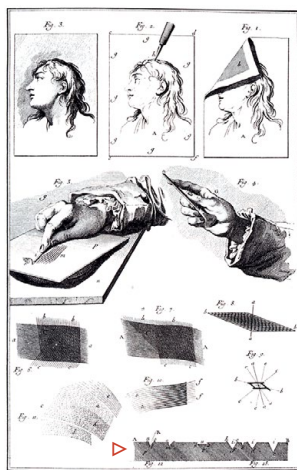
Für diese Abgrenzung gibt es verschiedene Ansätze. Lange galt als Grafik (griech: graphein = schreiben, ritzen) alles, was durch Schreiben, Zeichnen oder Einritzen erzeugt und vervielfältigt wurde, beispielsweise ein Holzschnitt (Hochdruck, Abb. 6.1.2), ein Kupferstich oder eine Radierung (beides Tiefdruck; Abb. 6.1.3). Eine Abgrenzung zwischen Bild und Text ist hier noch willkürlich, denn alles ist „Grafik“.



links: Druckstock, Süddeutschland um 1470–1475

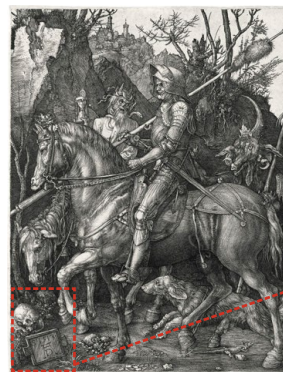
rechts: Druck davon (British Museum)

Abb. 6.1.2 | Holzschnitt „Martyrium des heiligen Sebastian“.



Oben: Darstellung der Kupferstichtechnik in Diderots Encyclopédie (um 1760); beachte die unterschiedliche Tiefe der Gravuren (▷)

Rechts:  
Einer der drei Meisterstiche Albrecht Dürers:  
„Ritter, Tod und Teufel“ (1513) sowie  
ein Detail daraus;  
beachte den Linien-  
raster im Ausschnitt



Links:  
Radierung „El sueño de la razón produce monstruos“  
von Francisco Goya (1799)

Abb. 6.1.3 | Kupferstiche und Radierungen.

### Aktuell

In Abgrenzung zum Bild (z. B. einem Foto) kann Grafik als eine von einem Grafiker/einer Grafikerin erstellte Darstellung eines Objekts oder Sachverhalts definiert werden. In diesem Sinn wären auch Gemälde als Grafiken zu verstehen.

In der Druck- und Medienvorstufe ist es teilweise gebräuchlich, alle Darstellungen, die aus geometrischen Grundformen bestehen, als *Grafik* zu bezeichnen. Entsprechend ist alles, was nicht aus geometrischen Grundformen besteht, ein *Bild*.

</Ø> Eine Möglichkeit, diese Abgrenzungsschwierigkeiten zu umgehen, besteht darin, in der Druck- und Medienvorstufe alles, was nicht reiner Text ist, als **Abbildung** zu bezeichnen.

Einteilung von Grafiken (► Abb. 6.1.1, Seite 68)

- ▶ Dimension: 2-D/3-D
- ▶ Ausgabeziel: Print/Web
- ▶ Aussage: Logo/Piktogramm/CAD-Darstellung/Infografik/Karte usw.
- ▶ Animation: statisch/animiert
- ▶ Datenformat: Pixel-/vektorbasiert (beides ist nebeneinander in einer Grafik möglich)

## 6.2 Datenformate für Grafiken

Für die Praxis wichtig ist das Datenformat, in dem eine Abbildung bzw. ein Teil davon vorliegt. Es gibt zwei Kategorien:

- ▶ Pixelformate
- ▶ Vektorformate

Auswahl gebräuchlicher Formate:

| Name     | <u>V</u> ektor/ <u>P</u> ixel | Anwendung                                       | Anmerkung                                                                  |
|----------|-------------------------------|-------------------------------------------------|----------------------------------------------------------------------------|
| JPEG     | P                             | Fotos, von Druckvorstufe bis Internet           | Variable Kompressionsstufe, die das Ausmaß der Verluste bestimmt           |
| PNG      | P                             | Internetbilder                                  | flexible Farbtiefe, Transparenz über Alphakanal*                           |
| TIFF     | P                             | Druckvorstufe                                   | Containerformat, sehr flexibel anpassbar, Alphakanäle und weitere Features |
| SWF      | V, P                          | Internetanimationen, Spiele                     |                                                                            |
| SVG      | V, P                          | Internet (auch Animationen)                     |                                                                            |
| PS (EPS) | V, P                          | Druckvorstufe                                   | Wird zunehmend von PDF abgelöst                                            |
| PDF      | V, P                          | Druckvorstufe, elektronische Dokumente, Archive | Wichtigstes Datenformat der Druckvorstufe                                  |

### 6.2.1 \*Alphakanal

Ein Alphakanal ist ein zusätzlicher Farbkanal, der ergänzende Informationen über die Transparenz für jedes Pixel enthält. Typische Anwendungen für Alphakanäle sind:

- ▶ Transparenz
- ▶ Freistellung
- ▶ Maskierung

Ein Alphakanal kann dieselbe Farbtiefe wie ein Farbkanal des Bildes haben, z. B. 8 bit. Ein binärer Alphakanal weist nur 1 bit Pixeltiefe auf. Ein Pixel in dem Alphakanal kann dann nur 2 verschiedene Werte annehmen:  $\alpha = 0$  (vollständig transparent) oder  $\alpha = 1$  (nicht transparent). Bei einem 8bit-Alphakanal liegt  $\alpha$  zwischen 0 (vollständig transparent) und 255 (nicht transparent);  $\alpha = 127$  bedeutet dann 50% Transparenz (► Abb. 6.2.1, Seite 71).

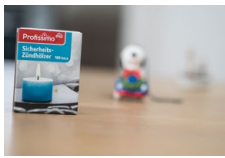


Foto ohne Alphakanal.  
Die Bildbox hat einen blauen Hintergrund, der von dem Foto komplett überdeckt wird.



Ein binärer (▼) Alphakanal bewirkt, dass nur die Streichholzschachtel (keine Transparenz) vor dem blauen Hintergrund der Bildbox zu sehen ist.



Ein 8bit-Alphakanal (▼) bewirkt, dass die Streichholzschachtel mit voller Deckkraft (= keine Transparenz) auf dem blauen Hintergrund steht, der Rest des Bildes mit einer Transparenz von 50% den Hintergrund durchscheinen lässt.



In diesem binären Alphakanal hat  $\alpha$  nur zwei Werte:  $\alpha = 1$  (Streichholzschachtel, keine Transparenz) und  $\alpha = 0$  (Rest des Bildes, volle Transparenz)



In diesem 8bit-Alphakanal hat die Streichholzschachtel den Wert  $\alpha = 255$  (keine Transparenz), der Rest des Bildes den Wert  $\alpha = 127$  (50% Transparenz)

Abb. 6.2.1 | Steuerung der Transparenz mithilfe von Alphakanälen.

## Transparenz in GIF und PNG:

GIF (Graphics Interchange Format): Es liegt kein Alphakanal vor, aber einzelne Pixel können einen Wert aufweisen, der in der Farbtabelle des GIF-Bildes als transparent definiert ist.

PNG (Portable Network Graphics): als PNG8 mit 1-bit für Transparenz, als PNG24 mit 8 bit für Transparenz. Es liegt aber kein in Photoshop editierbarer Alphakanal vor. Erstellen in Photoshop über Ebenenkomposition, dort für jede Ebene Deckkraft einstellen, dann „speichern für Web“ als PNG8 oder PNG24 mit Transparenz.

## 6.2.2 Eigenschaften von Pixel- und Vektorformaten

### Pixelformate

Pixelformate sind auflösungsabhängig. Das bedeutet:

- Dateigröße = Pixelzahl in der Breite x Pixelzahl in der Höhe x Farbtiefe  
Beispiel: 10 Pixel breit, 6 Pixel hoch, 3 Farbkanäle mit je 8 bit plus ein Alphakanal mit ebenfalls 8 bit:  $(10 \cdot 6) \cdot 4 \cdot 8 = 1920 \text{ Bit} = 240 \text{ Byte}$  (reine Bildinformation ohne Header).
- Auflösung = Pixelzahl pro Streckeneinheit (z.B. 72 px/inch)

Verkleinern ist ohne erkennbaren Qualitätsverlust möglich, Vergrößern grundsätzlich mit einem mehr oder minder deutlich erkennbaren Qualitätsverlust verbunden (Abb. 6.2.2).



Beim Skalieren *ohne* Neuberechnung verändert sich die Auflösung umgekehrt proportional zum Skalierungsfaktor.



Beim Skalieren *mit* Neuberechnung werden Pixel neu berechnet (Resampling). „Pixelwiederholung“ (links) ändert an den Stufen der diagonalen Linie nichts, „Details erhalten“ verfeinert sie (rechts).

Abb. 6.2.2 | Skalieren von Pixelbildern.

Pixel sind generell rechteckig, in der Druckvorstufe in der Regel quadratisch.  
Im Videobereich sind auch andere Seitenverhältnisse gebräuchlich:

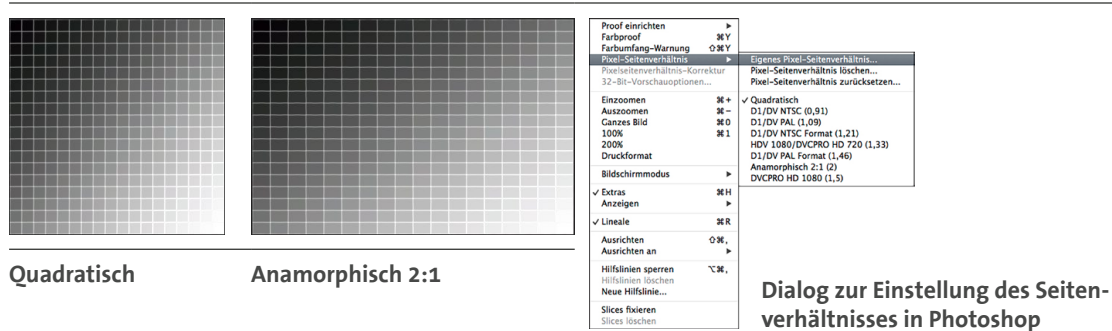


Abb. 6.2.3 | Quadratische sowie rechteckige Pixelformate mit gleichem Seitenverhältnis.

## Vektorformate

In Vektorformaten werden Objekte nicht als Pixel beschrieben, sondern als geometrische Figuren. Ein Kreis z. B. ist definiert durch die Koordinaten des Mittelpunkts und den Radius, ferner durch die Linienstärke/-farbe sowie die Füllung, die als einheitliche Farbe oder als Farbverlauf festgelegt sein kann.

Das bedeutet, dass die Objekte einer Vektorgrafik auflösungsunabhängig beschrieben werden. Das bedingt:

- ▶ Vektoriell beschriebene Objekte sind ohne jeglichen Verlust frei skalierbar (► Abb. 6.2.4).
- ▶ Der Speicherbedarf vektorieller Objekte ist verhältnismäßig klein.
- ▶ Für die Ausgabe auf einem Drucker/Raster-Image-Processor (RIP) bzw. für die Darstellung auf einem Monitor müssen Vektorgrafiken gerendert, d. h. in Rasterpunkte umgerechnet werden (► 6.4 Rendern, Seite 75).

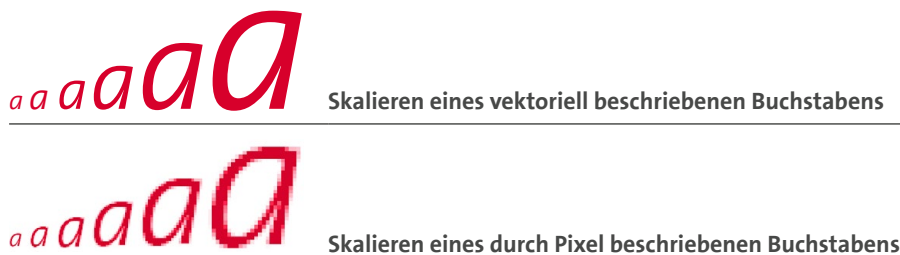


Abb. 6.2.4 | Skalieren vektor- oder pixelabhängiger Objekte.

## 6.2.3 Bearbeitungswerkzeuge für Abbildungen (Auswahl)

- Zeichenprogramme
  - ▷ Illustrator, Corel Draw
  - ▷ CAD(computer-aided design)-Programme (computergestütztes technisches Zeichnen)
- Bildbearbeitung mit Zeichentools
  - ▷ Photoshop, GIMP
- Mindmap-Editoren
- Prozessablauf-Editoren
- Editoren für spezifische Grafikformate, z. B. SVG-Editoren
  - ▷ Illustrator, Inkscape
- Office-Programme
  - ▷ PowerPoint, Excel, OpenOffice



## 6.3 Bézierkurven

Vektorbasierte Grafikprogramme verwenden für die Beschreibung von Linien, Polygonzügen/ Polygonen und Kurven Bézierkurven. Diese sind nach dem französischen Mathematiker Pierre Bézier benannt, der sie zur Beschreibung runder Karosserieformen im Autobau in den 1960er-Jahren entwickelte.

Eine Bézierkurve besteht aus dem Anfangspunkt, dem Endpunkt und einer variablen Zahl von Kontrollpunkten. Die Kontrollpunkte bestimmen die Krümmung der Verbindungslinie zwischen Anfangs- und Endpunkt. Weist eine Bézierkurve keinen Kontrollpunkt auf, definiert sie eine gerade Linie und wird als lineare Bézierkurve bezeichnet, synonym als Bézierkurve ersten Grades. Entsprechend wird eine Bézierkurve mit 1 Kontrollpunkt als quadratisch (= zweiten Grades), eine Kurve mit 2 Kontrollpunkten als kubisch (= dritten Grades) bezeichnet. Es gibt auch Bézierkurven höheren Grades (d. h. mit mehr als zwei Kontrollpunkten). In Programmen für die Druckvorstufe werden sie nicht verwendet.

Abbildung 6.3.1 zeigt Bézierkurven ersten, zweiten und dritten Grades. Die grau gezeichneten Polygone verbinden Anfangs- und Endpunkt und ggf. die dazwischen liegenden 1 oder 2 Kontrollpunkte. Man erkennt, dass der zugehörige Kontrollpunkt immer auf der konvexen Seite der Kurve liegt.

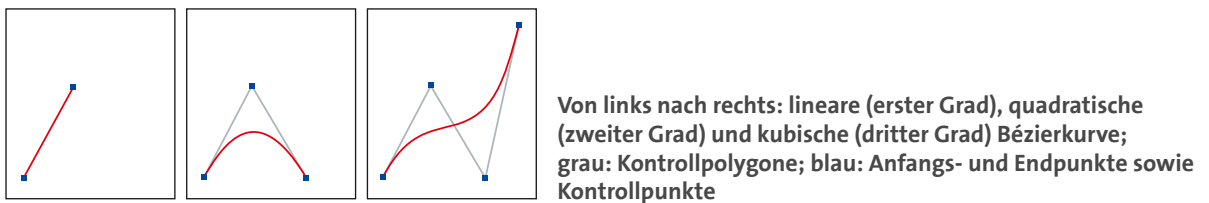


Abb. 6.3.1 | Bézierkurven ersten bis dritten Grades.

Werden mehrere Bézierkurven aneinandergereiht, entstehen zusammengesetzte Bézierkurven. Ist der Anfangspunkt der ersten identisch mit dem Endpunkt der letzten, entsteht eine geschlossene zusammengesetzte Bézierkurve (auch als Beziergon oder Bezigon bezeichnet).

Ein Beispiel für Anwendung zusammengesetzter Bézierkurve ist die Beschreibung der Buchstabenformen in PostScript- und TrueType-Schriften. Abbildung 6.3.2 zeigt das am Beispiel des Buchstabens „g“ in der PostScript-Schrift „Meta“.

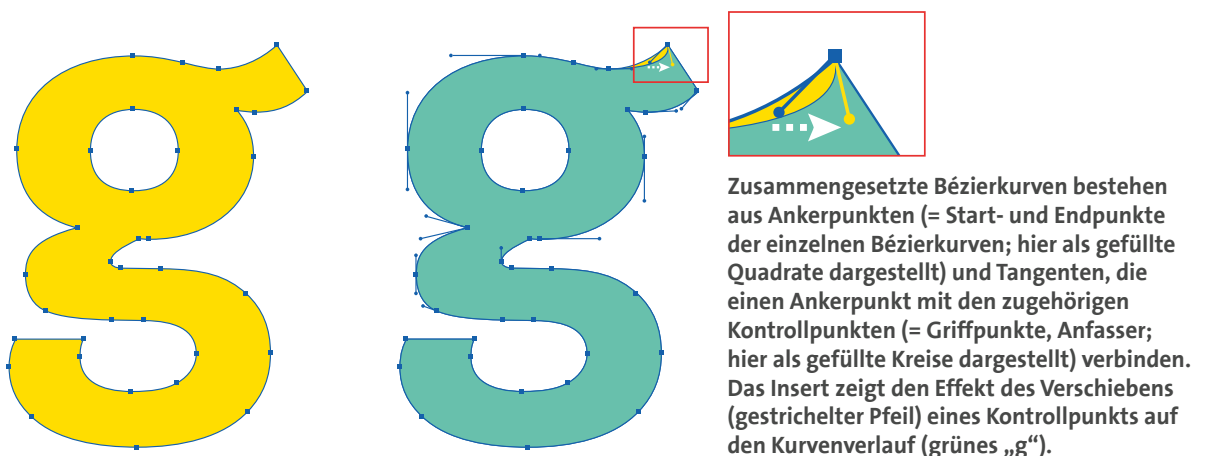


Abb. 6.3.2 | Zusammengesetzte Bézierkurve zur Beschreibung eines Buchstabens.

Hinweis:

Ein technischer Unterschied zwischen TrueType- und PostScript-Schriften besteht darin, dass TrueType nur Bézierkurven ersten und zweiten Grades verwendet, PostScript nur Bézierkurven ersten und dritten Grades. Das hat zur Folge, dass zur Beschreibung einer bestimmten Form in TrueType-Schriften mehr Ankerpunkte (auch als Stützpunkte bezeichnet) nötig sind als in

PostScript-Schriften. Abbildung 6.3.3 zeigt das am Beispiel des großen „B“ aus der „Times New Roman“, einmal als Truetype- (39 Stützpunkte), einmal PostScript-Schrift (22 Stützpunkte).

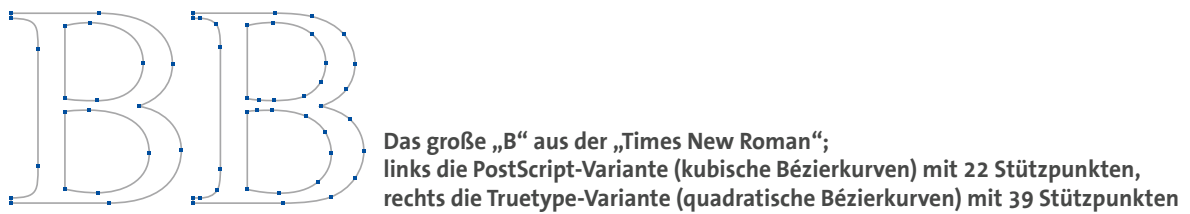


Abb. 6.3.3 | Quadratische Bézierkurven erfordern mehr Stützpunkte als kubische.

Die verbreitete Ansicht, dass Truetype-Schriften wegen der Verwendung quadratischer Bézierkurven und der daraus resultierenden größeren Zahl von Stützpunkten generell schlechter seien als PostScript-Schriften, ist falsch. Der schlechte Ruf der Truetype-Schriften rührt daher, dass das Format häufig für billige Schriften verwendet wird, die schlecht digitalisiert wurden und Schwächen bei der Zurichtung (z. B. fehlende Kerning-Tabellen) aufweisen.

Ein Kreis lässt sich mit Bezierkurven nicht exakt beschreiben. Üblich ist, einen Kreis aus 4 gleich großen kubischen Bézierkurven zusammenzusetzen. So ein Kreis weist somit 4 Stützpunkte mit zusammen 8 Kontrollpunkten (2 pro Bézierkurve) auf. Um den Kreis mit derselben Genauigkeit durch quadratische Bézierkurven zu beschreiben, sind 8 Elemente nötig, d. h. 8 Stützpunkte mit zusammen ebenfalls 8 Kontrollpunkten (1 pro Bézierkurve). Abb. 6.3.4 zeigt das am Beispiel des Zeichens „Bullet“ aus den beiden Times-New-Roman-Varianten.

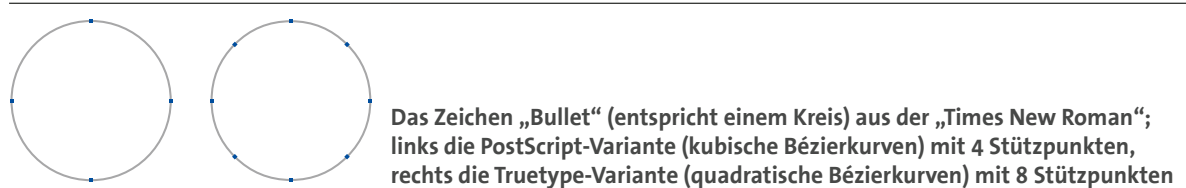


Abb. 6.3.4 | Beschreibung eines Kreises mit einer quadratischen und einer kubischen Bézierkurve.

## Splines

Im Zusammenhang mit Bézierkurven wird der Begriff „Spline“ sowohl für einzelne Bézierkurven als auch für zusammengesetzte Bézierkurven verwendet. Der Begriff stammt aus dem Schiffsbau und beschreibt dort den Verlauf von Planken.

Splines sind stückweise aus Polynomen (z. B. einzelnen Bézierkurven) zusammengesetzt. Die Randbedingungen für die erste und zweite Ableitung an einem Anknüpfungspunkt bestimmen, ob eine Ecke oder eine Rundung entsteht.

- ▶ Ecke bedeutet, dass die Kurve an dieser Stelle nicht differenzierbar ist.
- ▶ Eine Rundung entsteht, wenn die Kurve mindestens einmal differenzierbar ist.

Sind erste und zweite Ableitung definiert (sog. C1- und C2-Kriterien), verläuft die Kurve besonders „geschmeidig“. Dies wird z. B. bei der Konstruktion von Achterbahnen und Gleisen genutzt, um ruckartige Bewegungen der Wägen zu vermeiden.

Abbildung 6.3.5 zeigt einen Spline mit 9 Anknüpfungspunkten.

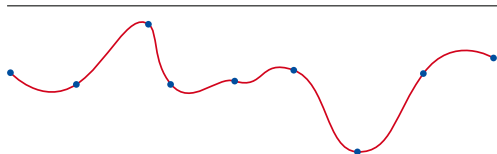


Abb. 6.3.5 | Spline mit 9 Anknüpfungspunkten.



## <Ø> Berechnen des Verlaufs von Bézierkurven

Die Form einer kubischen Bézierkurve ist durch den Start- und Endpunkt sowie 2 Kontrollpunkte außerhalb der Kurve bestimmt. Daraus ergibt sich folgende Parameterform:

$$x(u) = x_0 (1-u)^3 + x_1 3u (1-u)^2 + x_2 3u^2 (1-u) + x_3 u^3$$

$$y(u) = y_0 (1-u)^3 + y_1 3u (1-u)^2 + y_2 3u^2 (1-u) + y_3 u^3$$

Der Kurvenparameter  $u$  läuft von 0 bis 1 und wird entlang der Kurve erhöht.

Abbildung 6.3.6 zeigt das Ergebnis für eine kubische Bézierkurve (2 Kontrollpunkte).

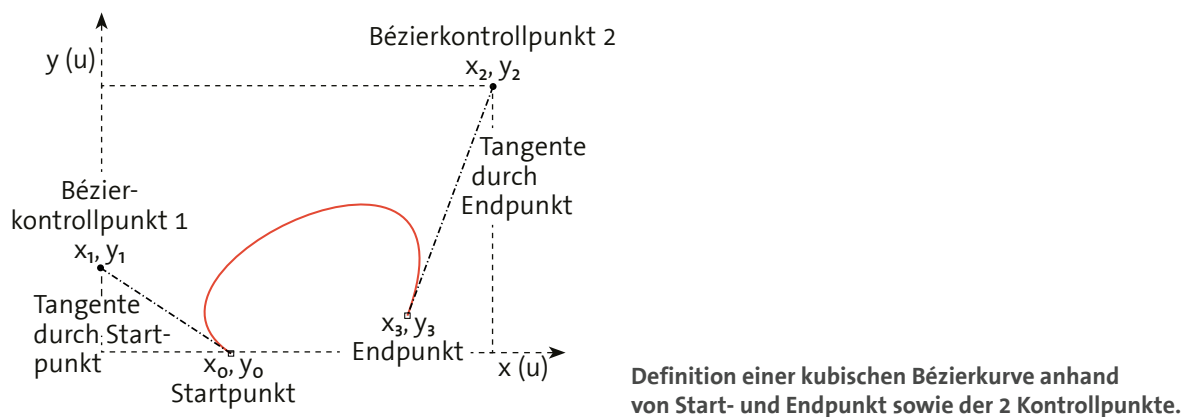


Abb. 6.3.6 | Zeichnen einer kubischen Bezierkurve mit 2 Kontrollpunkten.

Abbildung 6.3.7 zeigt einzelne Schritte des Zeichnens von Bézierkurven ersten, zweiten und dritten Grades anhand von Ausschnitten aus animierten GIFs.

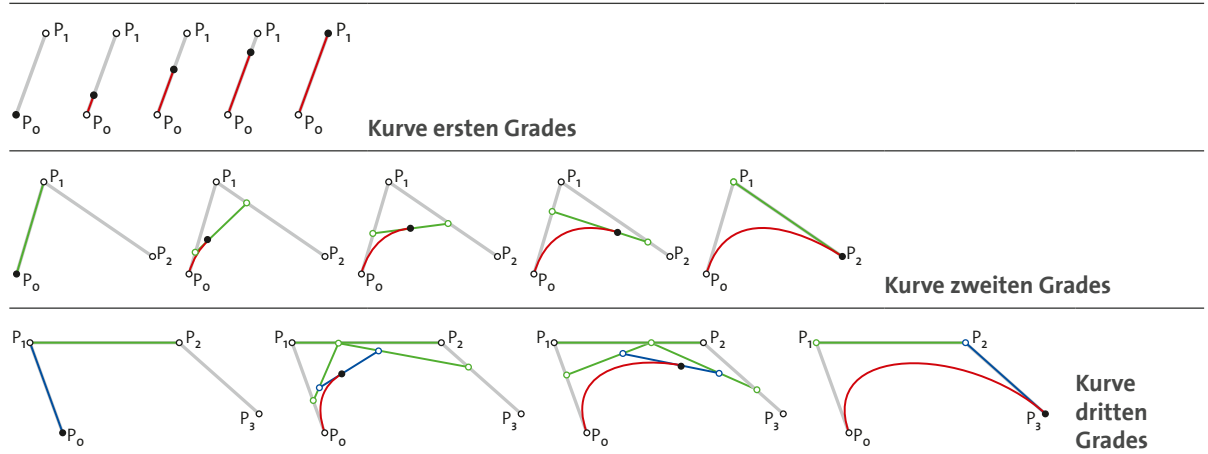


Abb. 6.3.7 | Konstruieren von Bézierkurven.

Quelle der animierten GIFs: <http://de.wikipedia.org/wiki/Bézierkurve>

</Ø>

## 6.4 Rendern (= Pixelausgabe) von Vektorgrafiken

Rendern umfasst folgende Schritte:

- ▶ Koordinatensystem für Pixelausgabe berechnen
- ▶ Geometrische Transformationen in das und auf dem Ausgabesystem
- ▶ Interpolationskurven für gekrümmte Linien konstruieren
- ▶ Berechnen der Pixel
- ▶ Optimierung der Ausgabe, z. B. durch Antialiasing

## Koordinatensystem

Typischerweise liegt der Ursprung des Ausgabekoordinatensystems oben links, weil digitale Bilder von oben links nach unten rechts gespeichert werden. Die Zählung beginnt bei 0, sodass der Koordinatenursprung die x- und y-Werte 0,0 hat (Abbildung 6.4.1).

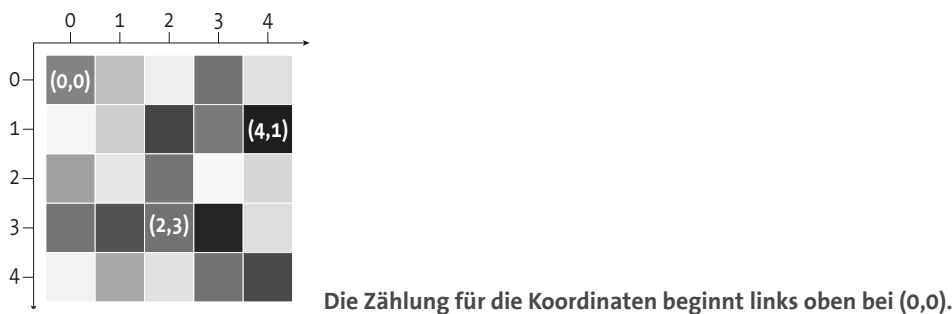


Abb. 6.4.1 | Typisches Koordinatensystem für die Ausgabe.

## Geometrische Transformationen

Bei der Darstellung von Grafiken auf einem Monitor sind laufend Neuberechnungen erforderlich, z. B. beim Verändern der Ausgabefenstergröße. Viele Neuberechnungen erfolgen durch lineare Koordinatentransformationen:

- ▷ Translation
- ▷ Rotation
- ▷ Skalierung
- ▷ Scherung

Rechnerisch sind dazu Matrixmultiplikationen erforderlich, z. B. mit folgender Matrix für die Drehung um den Winkel  $\alpha$ :

$$\begin{pmatrix} x_{\text{neu}} \\ y_{\text{neu}} \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x_{\text{alt}} \\ y_{\text{alt}} \end{pmatrix} = \begin{pmatrix} x_{\text{alt}} \cos \alpha - y_{\text{alt}} \sin \alpha \\ x_{\text{alt}} \sin \alpha + y_{\text{alt}} \cos \alpha \end{pmatrix}$$

## Interpolationskurven für gekrümmte Linien

Bézierkurven mit unterschiedlichen Krümmungen werden durch sehr viele kurze gerade Liniensegmente angenähert. Je kleiner die Liniensegmente, desto glatter werden die Kurven ausgegeben. Aber auch der Rechenaufwand im Raster-Image-Processor (RIP) steigt.

<Ø> Gesteuert wird die Genauigkeit der Approximation in PostScript-Dateien (▶ Seite 92) anhand des Parameters „Flatness“ (*num setflat*; gültige Werte für *num* reichen von 0.2 bis 100). Flatness ist die maximale Abweichung vom idealen Verlauf, angegeben in Ausgabegerätepixeln. Beispiel: 30 Ausgabegerätepixel sind bei einem Drucker mit 300 dpi 1/10 inch, gleich 2,54 mm – eine deutlich sichtbare Abweichung. Ein Kreis würde mit einem großem Wert für die Flatness als Polygon mit z.B. 32, 16 oder sogar nur 8 Ecken ausgegeben. Bei einem Belichter mit 2540 dpi dagegen entsprechen 30 Gerätepixel nur etwa 0,3 mm.

Der Parameter Flatness wird beim Umwandeln in eine PDF-Datei ignoriert. Im PDF-Workflow wird die Flatness in Abhängigkeit von der Ausgabeauflösung korrekt festgelegt. </Ø>

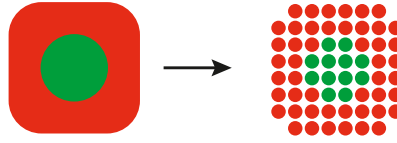
## Berechnen der Pixel

Nachdem der Pfad einer Vektorgrafik in den vorangegangenen Schritten definiert worden ist, wird eine Regel festgelegt, die entscheidet, ob ein Pixel zum Pfad zählt.

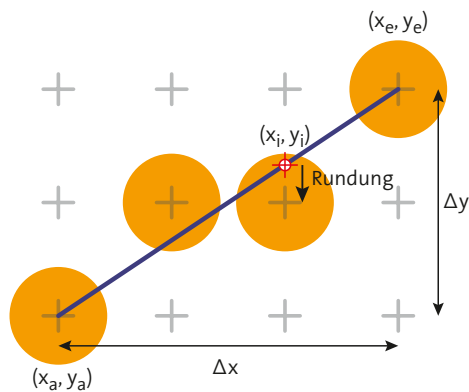
Beispiele: Alle Pixel, die vom Pfad berührt werden, oder alle Pixel, die mit mehr als 50% ihrer Fläche innerhalb des Pfads liegen, zählen zum Pfad. Pixel, die die gestellte Bedingung erfüllen, werden eingefärbt bzw. als druckend definiert.

Es folgen Beispiele für das Rendern einer SVG-Grafik (Abb. 6.4.2) und das Rendern einer Geraden (Abb. 6.4.3).

```
<rect width="8" height="8"
  rx="2" ry="2"
  fill="#C6312A"/>
<circle cx="4" cy="4" r="2"
  fill="#33CC20"/>
```



**Abb. 6.4.2 | Rendern einer SVG-Grafik (Kapitel 7 SVG-Grafik, Seite 79).**  
(Nach <http://commons.wikimedia.org/wiki/File:Scan-conversion.svg>)



1. Schritt:  
Anhand der allgemeinen Geradengleichung  
 $y = m(x - x_a) + y_a$   
werden exakte x,y-Koordinaten berechnet.
2. Schritt:  
Die Rundung auf die nächstgelegenen ganzzahligen x,y-Werte  
ergibt die einzufärbenden Pixel.

**Abb. 6.4.3 | Rendern einer Geraden.**  
(Nach [http://commons.wikimedia.org/wiki/File:Line\\_scan-conversion\\_rounding-de.svg](http://commons.wikimedia.org/wiki/File:Line_scan-conversion_rounding-de.svg))

Abbildung 6.4.3 lässt erkennen, dass das Runden zu Problemen führt, die eine Optimierung erfordern:

- ▷ Treppeneffekte (Aliasing, „Jaggies“),
- ▷ Identische Linien werden bei unterschiedlicher Steigung oder Lage nicht gleich lang bzw. gleich dick dargestellt,
- ▷ je nach Rundung kann eine Gerade eine unterschiedliche Steigung haben.

Diese Rundungsfolgen fallen vor allem bei niedriger Auflösung auf, z. B. Darstellung am Monitor oder Druck mit Auflösungen von weniger als etwa 600 dpi. Wird Text auf Druckern mit niedriger Auflösung ausgegeben, leidet das Schriftbild durch Runden: Gleiche Buchstaben, z. B. ein „i“, fallen unterschiedlich breit aus oder Buchstabenkonturen werden deformiert.

TrueType-Schriften und PostScript-Typ-1-Schriften enthalten „Hints“, d. h. Hinweise für die Rundung auf Ausgabegeräten mit niedriger Auflösung. Das verbessert die Ausgabequalität wesentlich. Allerdings gehen diese Hints verloren, wenn man Schriften im Layout- oder Grafikprogramm in Pfade umwandelt (wie das manche Online-Druckdienste fordern, um Schriftenprobleme grundsätzlich auszuschließen). Für hochauflösende Drucksysteme (Film- oder Plattenbelichter) ist das kein Problem. Betrachtet man aber z. B. eine PDF-Datei, die Pfade statt der ursprünglichen Schrift enthält, am Bildschirm, können Rundungsunterschiede wegen der im Vergleich zum Druck niedrigen Auflösung (je nach Monitor rund 100 dpi) erkennbar werden.

## Antialiasing

Abbildung 6.4.4 zeigt das Abmildern von Rundungsfolgen durch Antialiasing. Dabei werden angrenzende Pixel entsprechend dem Grad ihrer Beteiligung an der gerenderten Struktur gefärbt, um die Kontur glatter aussehen zu lassen. Diesem Prinzip folgt z. B. die Schriftenglättung an einem Monitor. Je nach Intensität der Kantenglättung können die Objekte dadurch fetter und leicht unscharf wirken. (Neben Antialiasing kann zur Schriftenglättung am Bildschirm auch noch das Subpixel-Rendering eingesetzt werden, Seite 103.)

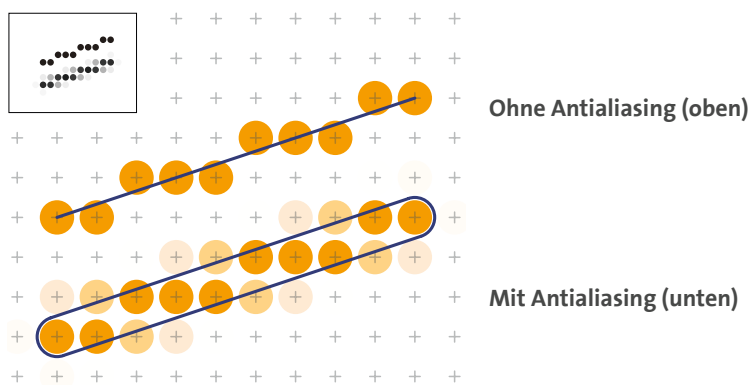


Abb. 6.4.4 | Antialiasing zur Abmilderung von Treppeneffekten.  
(Nach [http://commons.wikimedia.org/wiki/File:Line\\_scan-conversion.svg](http://commons.wikimedia.org/wiki/File:Line_scan-conversion.svg))

## <!> Zusammenfassung „Grafikformate“

### ■ Abgrenzung von Grafik, Bild und Text

- ▷ In der Druck- und Mediovorstufe ist es teilweise gebräuchlich, alle Darstellungen, die aus geometrischen Grundformen bestehen, als *Grafik* zu bezeichnen. Entsprechend ist alles, was nicht aus geometrischen Grundformen besteht, z. B. ein Foto, ein *Bild*.
- ▷ Diese Abgrenzungsschwierigkeiten lässt sich umgehen, indem alles, was nicht reiner Text ist, als **Abbildung** bezeichnet wird.

### ■ Datenformate für Grafiken

- ▷ Pixelformate (u. a. JPEG, PNG, TIFF): auflösungsabhängig, nicht verlustfrei skalierbar. Alphakanal: zusätzlicher Farbkanal, der ergänzende Informationen über die Transparenz für jedes Pixel enthält; von JPEG nicht unterstützt.
- ▷ Vektorformate (u. a. SVG, PostScript): auflösungsunabhängig, verlustfrei skalierbar.

### ■ Bézierkurven

- ▷ Vektorbasierte Grafikprogramme verwenden für die Beschreibung von Linien, Polygonzügen/Polygonen und Kurven Bézierkurven.
- ▷ Eine Bézierkurve besteht aus Anfangs- und Endpunkt sowie einer variablen Zahl von Kontrollpunkten.
- ▷ In der Druckvorstufe kommen nur lineare (0 Kontrollpunkte), quadratische (1 Kontrollpunkt) und kubische (2 Kontrollpunkte) Bézierkurven vor.
- ▷ Mehrere Bezierkurven bilden eine zusammengesetzte Bézierkurve, z. B. zur Definition der Kontur von Buchstaben in einer Schriftendatei (PostScript, Truetype).
- ▷ Beide Schriftarten verwenden lineare Bézierkurven plus quadratische (Truetype) bzw. kubische Bézierkurven (PostScript); das hat keinen Einfluss auf die Qualität einer Schrift.
- ▷ Als Spline wird im Zusammenhang mit Bézierkurven sowohl eine einzelne als auch eine zusammengesetzte Bézierkurve bezeichnet.

### ■ Rendern (= Pixelausgabe) von Vektorgrafiken

- ▷ Für die Ausgabe auf einen Monitor oder Drucker muss Vektorgrafik gerendert werden. Schritte sind:
  - Koordinatensystem berechnen: beginnt üblicherweise links oben mit (0,0).
  - Geometrische Transformationen, z. B. Skalieren, Rotieren, Nigen usw.
  - Interpolationskurven für gekrümmte Linien konstruieren: Annäherung durch zahlreiche kurze geraden Liniensegmente.
  - Berechnen der Pixel: Welche Pixel werden für die Ausgabe der Grafik gefärbt bzw. als druckend definiert?
- ▷ Beim Rendern muss immer auf- oder abgerundet werden. Bei niedriger Auflösung führt das zu sichtbaren Problemen wie Treppeneffekten und ungenauer Darstellung von Linien und Kurven. Antialiasing mildert diese Folgen. PostScript- und Truetype-Schriften enthalten hierfür sog. Hints, die beim Umwandeln der Schrift in Pfade verloren gehen.

## 7 SVG (Scalable Vector Graphics)

### Übersicht

Vektorgrafiken für das Web werden in SVG erstellt. Für die Druckvorstufe werden die Vektorformate PostScript (► Kapitel 8, Seite 92) bzw. das darauf basierende PDF-Format (► Kapitel 9, Seite 105) genutzt. Dieses Kapitel beschäftigt sich mit SVG.

### 7.1 Einführung

– Was ist SVG, was kann es?

### 7.2 Kurzübersicht zu Markup-Sprachen

– Von SGML zu HTML, XML und HTML5

– HTML

– XML

### 7.3 Cascading Style Sheets (CSS) in HTML, XML und SVG

### 7.4 SVG: Grundlagen, Beispiele

---

### 7.1 Einführung

#### Merkmale von SVG

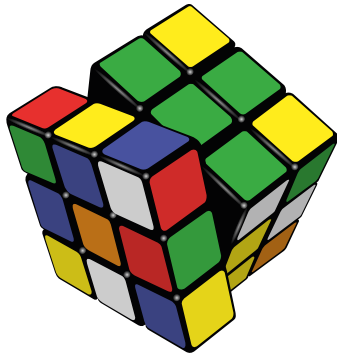
- SVG ist eine textorientierte Auszeichnungssprache (Markup Language, ► Seite 80), um zweidimensionale skalierbare Grafiken zu beschreiben.
- Die Grafiken können mit einfachen Texteditoren erstellt/bearbeitet werden.
- SVG-Grafiken lassen sich auch in gängigen Grafikprogrammen erstellen und bearbeiten, z. B. Adobe Illustrator. Inkscape ist eine Open Source Software für SVG-Grafiken und auf verschiedenen Plattformen verfügbar (<http://inkscape.org/>)
- SVG beruht auf XML (Extensible Markup Language, ► Seite 84).
- Neben Vektorgrafiken können auch Bilder und Texte enthalten sein.
- Stylesprachen (z. B. CSS, ► Seite 85) werden unterstützt.
- Scriptsprachen (z. B. JavaScript, ECMAScript) werden ebenfalls unterstützt.
- Darstellen lassen sich SVG-Grafiken mit Browser-Plugins oder eigenständigen SVG-Viewern.

#### Grafische Möglichkeiten von SVG

- Geometrische Grundformen (Linie, Kreis usw.)
- Unterstützung für Text und Rastergrafiken (z.B. als PNG-Bilder).
- Farben, Farbverläufe und Muster als Füllung für SVG-Objekte
- Elemente können als „Entitäten“ definiert und beliebig oft aufgerufen werden.
- SVG-Objekte können Interaktiv sein und mit Hyperlinks hinterlegt sein.
- Set von Filtereffekten (Gaußscher Weichzeichner, Schatten usw.)
- Animationen (Syntax aus SMIL: Synchronized Multimedia Integration Language)



SVG wurde entwickelt, um verlustfrei skalierbare Grafiken mit kleiner Dateigröße für das Web zu ermöglichen. Der Quellcode für das Beispiel in Abbildung 7.1.1 umfasst weniger als 16 kb. Nach Umwandeln in das Format des „Illustrators“ (CC 2014) für die Darstellung in diesem Skript (Indesign CC 2014 und Quark XPress 10 unterstützen SVG-Grafiken nicht) hat die Grafik eine Größe von 3,1 MB, als EPS (Level 3, ohne VorschauBild) 318 kb, als PDF (kleinste Dateigröße) 113 kb.



Dateigröße: knapp 16 kb

Abb. 7.1.1 | „Rubik's Cube“ als SVG-Grafik

Quelle: [http://commons.wikimedia.org/wiki/File:Rubik%27s\\_cube\\_v3.svg](http://commons.wikimedia.org/wiki/File:Rubik%27s_cube_v3.svg)

## 7.2 Kurzübersicht zu Markup-Sprachen

Charakteristisch für Markup-Sprachen ist, dass Text mithilfe von Tags („Abgrenzer“) ausgezeichnet wird. Tags weisen einem Textabschnitt ein Merkmal zu. *Semantische* Tags (engl.: *descriptive markup*) strukturieren den Text. Beispiel: das Tag „Ort“ in einer Adressensammlung. Tags können ferner dazu genutzt werden, das *Aussehen*, z. B. Fettdruck, zu steuern (Auszeichnung für die Darstellung, engl.: *procedural markup*).

<Ø>

**Ab hier bis zum Ende von Kapitel 7 nicht Stoff des Semesters!**

### 7.2.1 Von SGML zu HTML, XML und HTML5

SGML (Standard Generalized Markup Language) wird häufig als „Mutter“ aller Markup-Sprachen bezeichnet. Noch weiter zurück reicht jedoch die bei IBM Ende der 1960er-Jahre entwickelte GML (Generalized Markup Language). Ein Beispiel zeigt Abbildung 7.2.1.1.

```
:h1.Chapter 1: Introduction
:p.GML supported hierarchical containers, such as
:ol
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
as well as simple structures.
:p.Markup minimization (later generalized and formalized in SGML),
allowed the end-tags to be omitted for the "h1" and "p" elements.
```

Die Tags in diesem GML-Beispiel sind in Blau dargestellt. Die Nähe zu den heute z. B. in HTML verwendeten Tags ist gut erkennbar.

Abb. 7.2.1.1 | Beispiel aus der ersten Markup-Sprache: IBM GML (Generalized Markup Language)

Quelle: [http://en.wikipedia.org/wiki/IBM\\_Generalized\\_Markup\\_Language](http://en.wikipedia.org/wiki/IBM_Generalized_Markup_Language)

Ein Ziel der Entwicklung von SGML war, den Austausch umfangreicher maschinenlesbarer Dokumente z. B. zwischen Regierungen und Industrieunternehmen zu ermöglichen (insbesondere: US-Verteidigungsministerium und Rüstungsindustrie). Außerdem sollte die Lesbarkeit der Dokumente für lange Zeit gewährleistet sein. SGML ist seit 1986 ein ISO-Standard.

- SGML ist eine Markup-Sprache mit sehr komplexen Regeln. Diese Komplexität war nötig, um die Dokumente möglichst klein zu halten, denn Speicherplatz war Anfang der 1970er-Jahre noch knapp.

- SGML trennt die Dokumentstruktur vom Dokumentlayout und garantiert damit Systemunabhängigkeit der Dokumente. Je nach Ausgabebedingungen können unterschiedliche Layouts zugewiesen werden.
- Die Dokumentstruktur wird mithilfe der DTD (Document Type Definition) festgelegt. Die Strukturelemente des Dokuments werden mit Tags kenntlich gemacht.
- SGML-Dokumente und DTD bestehen aus ASCII (American Standard Code for Information Interchange; mittlerweile sind auch ISO-Zeichensätze sowie UTF-8 möglich).

Abbildung 7.2.2 zeigt die Weiterentwicklung wichtiger Markup-Sprachen, die von SGML ausging.

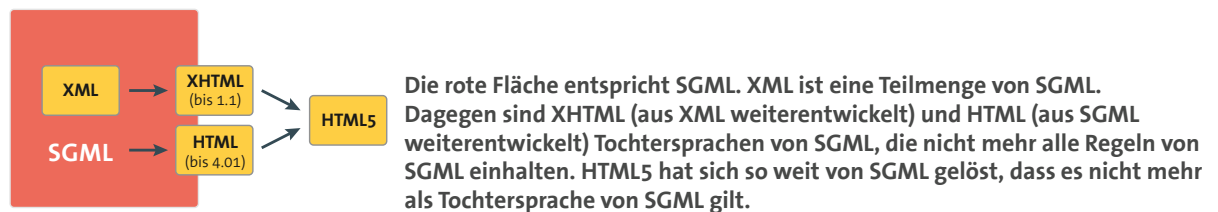


Abb. 7.2.2 | Weiterentwicklungen, die von SGML ausgingen.

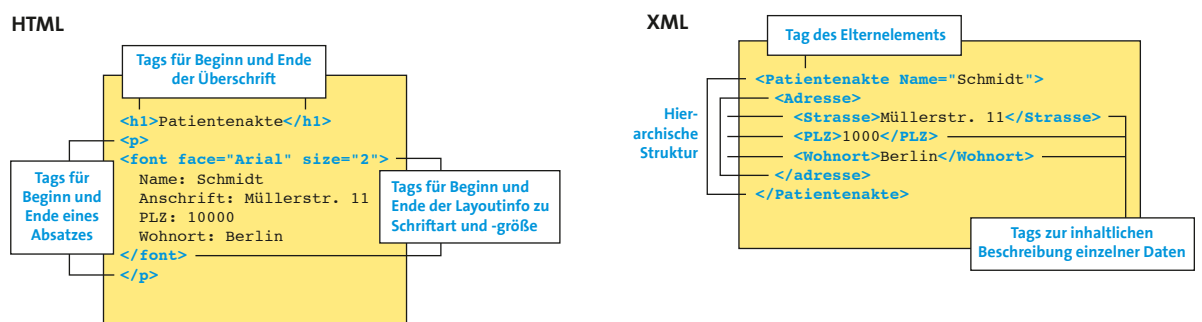
## HTML (Hypertext Markup Language, bis Version 4.01)

SGML wäre aufgrund seines knappen Codes gut für das Web geeignet, aber das Regelwerk ist für die meisten Zwecke zu komplex. Daher wurde ab etwa 1990 HTML aus SGML entwickelt. Da HTML nicht alle Konventionen von SGML einhält, ist es keine Teilmenge von SGML.

## XML (Extensible Markup Language)

HTML erwies sich bald als sehr formatlastig (z.B. `<b>`, `<i>` und viele weitere Tags). Man begann daher mit der Entwicklung von XML, einer „schlanken“ Version von SGML. Schlank bezieht sich auf den Umfang des Regelwerks. Während die Dokumentation von SGML über 500 Seiten umfasst, genügen für XML weniger als 30 Seiten. Da mit XML einige Neuerungen gegenüber SGML eingeführt wurden, hat man SGML nachträglich ergänzt (Annex J und K), damit XML eine Teilmenge von SGML bleibt. Die Erweiterungen von SGML erlauben neben ASCII auch ISO- und Unicode-Zeichensätze und lockern einige Restriktionen.

- XML erlaubt, strukturierte Daten (z. B. eine Datenbank) als Textdatei zu speichern.
- Wie HTML nutzt auch XML Tags und Attribute, um Strukturen kenntlich zu machen. Während HTML genau spezifiziert, was ein Tag bzw. Attribut bedeutet, nutzt XML die Tags nur zur Strukturierung und überlässt die Interpretation der Daten der Anwendung, die die XML-Daten liest. Den Unterschied verdeutlicht Abbildung 7.2.3.



Anders als in der HTML-Darstellung erkennt man in der XML-Variante einer Patientenakte die Struktur des Texts anhand der verschachtelten Tags. Die Formatierung kann z. B. mithilfe einer CSS-Datei festgelegt werden.

Abb. 7.2.3 | Vergleich von HTML und XML.

Quelle: Nach Max Dahms, 2004, Veränderung der Dokumentation in der klinischen Forschung durch XML, EDC und CDISC-Standards, München, Diplomica Verlag GmbH

- Die Darstellung von XML-Daten kann z. B. per CSS (🔴 Seite 85) gesteuert werden.
- SVG (🔴 Seite 87) folgt den XML-Standards und unterstützt CSS.
- XML ist lizenzfrei und plattformunabhängig (Hardware und Betriebssystem).
- XML-Daten eignen sich für Database Publishing, z. B. von Katalogen (über Scripte oder Plugins u. a. in InDesign).

## XHTML (Extensible Hypertext Markup Language, bis Version 1.1)

Die Entwicklung von XHTML war als minimalistische Neuformulierung von HTML auf der Basis von XML mit dem Ziel begonnen worden, für mobile Geräte besonders knappen Code schreiben zu können. Die Entwicklung wurde mit Version 1.1 zugunsten von HTML5 eingestellt.

## HTML5

HTML5 ist die Weiterentwicklung von HTML (Version 4.01). Die fertige Spezifikation wurde am 28. Oktober vom World Wide Web Consortium (W3C) vorgelegt. HTML5 lässt die Syntax von HTML 4.01 und von XHTML 1.0 zu, aber (noch) nicht alle Browser unterstützen das.

### 7.2.2 HTML, ausgewählte Tags

HTML mit seinen verschiedenen Versionen (einschließlich HTML5) ist die Dokumentbeschreibungssprache für Seiten im Web. HTML ist leicht zu lernen und Dokumente können mit einem einfachen Texteditor erstellt werden. Ein wesentliches Merkmal sind externe und interne Hyperlinks. Die Daten sind plattformunabhängig.

## Tag, Element, Attribut

Ein Tag („Abgrenzer“) ist eine Auszeichnung eines Text- bzw. Datenbereichs. Für Tags gilt Kleinschreibung.

Beispiel:

```
<h1>Einführung in HTML</h1>
```

`<h1>` ist das öffnende Tag, `</h1>` das schließende.

Wichtig:

Die meisten Tags müssen wieder geschlossen werden. Ausnahmen sind die sog. leeren Tags, beispielsweise ein Bild: ``. (Hinweis: In XHTML 1.0 muss jedes Tag geschlossen werden, also auch ein Bild: `</img>`.)

Öffnendes und schließendes Tag mit dem dazwischen stehenden Inhalt bilden ein Element.

Elemente dürfen unter Wahrung der Ebenen verschachtelt sein. Man spricht dann von Eltern- und Kindelementen.

Beispiel:

```
<h1>Einführung in <b>sauberes</b>HTML</h1>
```

Einführung in **sauberes** HTML

Block-Level-Elemente erzeugen am Ende einen Zeilenumbruch (z.B. nach `</div>`, `</h1>`, `</p>`), Inline-Elemente wie `<b>...</b>` aus dem Beispiel oben nicht. Eine gewollte Zeilenschaltung innerhalb eines Block-Level-Elements wird mit `<br>` erzeugt. Ein Zeilenumbruch im Quelltext wird vom Browser als Leerschritt ausgegeben. (In XHTML muss auch dieses Tag geschlossen werden: `<br></br>`.)

Mithilfe von Attributen werden Eigenschaften eines HTML-Elements im Starttag festgelegt.  
Beispiel:

```
<p align="center">
```

Hier bestimmt das Attribut `align` (Ausrichtung) mit dem Attributwert `center`, dass der Absatz zentriert wird. Attribute und Attributwerte sind in HTML (anders als in XML) nicht „case sensitive“, d. h. Klein- oder Großschreibung spielt keine Rolle. Sie werden aber meistens kleingeschrieben. Die Anführungszeichen für den Attributwert sind üblich, aber nicht Pflicht.

Es gibt drei Arten von Werten für Attribute:

- ▷ normiert (aus einer festen Auswahl)
- ▷ Zahlenwert (absolut oder als %-Angabe)
- ▷ variable Werte

Das vorhergehende Beispiel (`align="center"`) ist ein normiertes Attribut. Zur Auswahl stehen nur die Attributwerte `left`, `center`, `right` und `justify`.

Die Angabe der Breite einer Tabelle ist ein Zahlenwert:

```
<table width="400px">    Tabellenbreite 400 Pixel
<table width="30%">      Tabellenbreite 30% der Bildschirmbreite
```

Ein variabler Attributwert ist z. B. ein Link:

```
  Link zu dem Bild mit dem Dateinamen bild_01.jpg
<a href="www.pm.hm.edu"> (Externer) Link
```

- ⌘ Viele Attribute (z. B. `align`, siehe oben) werden in HTML5 zwar akzeptiert und angezeigt, aber sie gelten als „deprecated“, d. h. abgelehnt, überholt.  
Grund: Sie gehören in die CSS-Datei!

## Grundstruktur von HTML-Dokumenten

Jedes Dokument nach HTML5 muss mit einer Document Type Definition (DTD) beginnen:

```
<!DOCTYPE html>
```

Darauf folgt das umschließende HTML-Element, das den gesamten HTML-Teil einleitet und beendet:

```
<html>
```

Dazwischen steht der HTML-Anteil der Seite.

```
</html>
```

Innerhalb des Elements `html` gibt es maximal zwei mögliche Kindelemente:

- ▶ Kopf: `<head></head>`

Der Kopf (Header) kann Informationen zu Autor/Autorin, Beschreibungen des Inhalts und weitere Angaben (sog. Metatext) enthalten. Er ist im Browser nicht sichtbar, mit einer Ausnahme:

Das Element Title `<title></title>` ist zwingend, auch wenn es leer bleibt. Der Text zwischen den beiden Tags wird im Browser als Seitentitel angezeigt. Erfolgt keine Angabe zum Titel, zeigt der Browser im Allgemeinen den Dateinamen der HTML-Datei an.

- ▶ Rumpf: `<body></body>`. Das ist der im Browser sichtbare Teil der HTML-Seite.

## HTML-eigene Zeichen (Entities) und Sonderzeichen

Die HTML-eigenen Zeichen `<`, `>`, `"`, `'` und `&` werden als Entitäten bezeichnet. Wenn diese Zeichen im sichtbaren Text vorkommen, müssen sie auf spezielle Weise eingegeben werden, damit der Browser sie von Entitäten unterscheiden kann. Es gibt zwei Notationen: die HTML-eigene und die nach Unicode-Standard. Beide folgen dem Prinzip `&zeichen;`, d. h. am Beginn steht ein `&`, am Ende ein `;`.

Beispiel: Das Zeichen „<“ (kleiner als) kann als `&lt;` geschrieben werden (HTML-eigen) oder als `&#60;` (Unicode). HTML verwendet zur Codierung eine Abkürzung des englischen Zeichennamens (hier: lower than, abgekürzt als `lt`), Unicode die Nummer des Zeichens (hier `#60`).

| Zeichen | HTML                    | Abgeleitet von | Unicode                |
|---------|-------------------------|----------------|------------------------|
| <       | <code>&amp;lt;</code>   | lower than     | <code>&amp;#60;</code> |
| >       | <code>&amp;gt;</code>   | greater than   | <code>&amp;#62;</code> |
| "       | <code>&amp;quot;</code> | quotation mark | <code>&amp;#34;</code> |
| '       | <code>&amp;apos;</code> | apostrophe     | <code>&amp;#39;</code> |
| &       | <code>&amp;amp;</code>  | ampersand      | <code>&amp;#38;</code> |

Was in HTML als Sonderzeichen gilt, hängt vom Zeichensatz ab, der im Head definiert wurde. Üblich ist heute der Unicode-Zeichensatz UTF-8:

```
<meta charset="utf-8">
```

UTF-8 kann alle deutschen Sonderzeichen wie Umlaute und ß darstellen. Das gilt auch für die Zeichensätze iso-8859-1 bis iso-8859-10. In Zeichensätzen ohne deutsche Sonderzeichen müssen Umlaute als Sonderzeichen eingegeben werden:

```
Ä = &Auml;
ä = &auml;
ß = &szlig;
```

Hinweis:

Beim Erstellen von HTML-Seiten darauf achten, dass im Texteditor derselbe Zeichensatz eingestellt ist wie der für die HTML-Datei im Head definierte.

Ein weiteres Sonderzeichen ist das geschützte Leerzeichen (non-breaking space), das einen Zeilenumbruch verhindert:

```
&nbsp;
```

Da mehrere normale Leerzeichen hintereinander immer nur als ein Leerzeichen ausgegeben werden, müssen mehrere Leerzeichen als `&nbsp;` eingegeben werden.

Aber: Leerzeichen sollten nicht dazu dienen, Text auszurichten!

## Beispiele für Textauszeichnungen in HTML

|                            |                             |               |
|----------------------------|-----------------------------|---------------|
| <b>fett</b>                | <code>&lt;b&gt;</code>      | (bold)        |
|                            | <code>&lt;strong&gt;</code> | (strong)      |
| <i>kursiv</i>              | <code>&lt;i&gt;</code>      | (italic)      |
| gesperrt                   | <code>&lt;tt&gt;</code>     | (teletype)    |
| <u>unterstrichen</u>       | <code>&lt;u&gt;</code>      | (underline)   |
| <del>durchgestrichen</del> | <code>&lt;strike&gt;</code> | (strike)      |
|                            | <code>&lt;del&gt;</code>    | (delete)      |
| groß                       | <code>&lt;big&gt;</code>    | (big)         |
| klein                      | <code>&lt;small&gt;</code>  | (small)       |
| hochgestellt <sup>2</sup>  | <code>&lt;sup&gt;</code>    | (superscript) |
| tiefgestellt <sub>2</sub>  | <code>&lt;sub&gt;</code>    | (subscript)   |

### 7.2.3 XML, ausgewählte Tags

XML weist einige Besonderheiten gegenüber HTML auf, u. a.:

- Tags sind frei definierbar (beginnen mit einem Buchstaben, Unterstrich oder Doppelpunkt), Attributnamen und Attribute unterscheiden Groß-/Kleinschreibung („case sensitive“).
- Start- und Endtag sind immer erforderlich.

- Es muss exakt verschachtelt werden.
- Jeder Fehler führt zu einer Fehlermeldung mit Abbruch der Interpretation! (Anders als bei HTML, bei dem Browser mit Fehlern rechnen.)
- Jedes XML-Dokument enthält genau ein Rootelement.
- Als Zeichensatz können ISO-Zeichensätze sowie Unicode utf-8 und utf-16 verwendet werden.
- Validität („Gültigkeit“) erfordert u.a.:
  - ▷ Grammatik, die durch eine Doctype Definition (DTD) oder ein XML-Schema festgelegt ist:
 

```
<!DOCTYPE root SYSTEM „meine_DTD.dtd“>
```
  - ▷ Validität kann unter [validator.w3.org](http://validator.w3.org) getestet werden.

Beispiel für eine XML-Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE root SYSTEM "meine_DTD.dtd">
<!-- Hier kann ein Kommentar stehen, der keine doppelten Bindestriche
enthalten darf. -->
<root>
  <Inhalt>Beispiel für ein XML-Dokument</Inhalt>
  <Text>Das erste Beispiel für ein wohlgeformtes XML-Dokument</Text>
</root>
```

Hinweise:

- ▷ Statt „root“ kann jeder andere Begriff gewählt werden, z.B. wurzelement oder svg.
- ▷ Wird kein Zeichensatz angegeben, wird utf-8 vorausgesetzt.
- ▷ Wenn keine DTD existiert oder diese im Dokument enthalten ist, kann das in der XML-Deklaration (oberste Zeile) durch `standalone="yes"` ausgedrückt werden.

## 7.3 Cascading Style Sheets (CSS) in HTML, XML und SVG

Cascading Style Sheets bedeutet „stufenförmig aufeinander aufbauende Stilvorlagen“. Die Idee hinter der Einbindung von CSS-Dateien ist die Trennung von Inhalt und Layout. Durch Zuweisen einer anderen CSS-Datei lässt sich die Darstellung komplexer Websites einfach ändern. Außerdem kann über Media-Queries die Darstellung an das Ausgabemedium (Beispiele: Druck oder Bildschirm; maximale Breite des Bildschirms) angepasst werden (responsives Webdesign, RWD). CSS-Dateien sind kombinierbar mit HTML, XML und SVG.

### Zusammenspiel von HTML/XML und CSS

- HTML/XML legt die Struktur fest: Auszeichnungssprache.
- CSS bestimmt das Aussehen: Stylesheet-Sprache.

Beispiel:

```
<h1>Grundlagen Druckvorstufentechnik</h1>
```

Der Inhalt dieser Zeile ist der Text „Grundlagen Druckvorstufentechnik“.

Die Struktur wird durch das Überschriften-Tag `<h1>` zugewiesen.

Selbst ohne weitere Angaben würde diese Zeile von einem Browser als Überschrift dargestellt und ein anderes Aussehen haben als nachfolgender Text zwischen den Tags `<p>` und `</p>`.

Eine Layoutanweisung (z.B. fett, rote Textfarbe) könnte direkt in das öffnende h1-Tag geschrieben werden (Style-Attribut, „Inline-Style“):

```
<h1 style="color:red;font-weight:bold">Grundlagen Druckvorstufentechnik
</h1>
```

Ergebnis:

**Grundlagen Druckvorstufentechnik**



Diese Formatierung gilt nur für diese eine Textpassage in dem HTML-Dokument.

Mithilfe von CSS besteht die Möglichkeit, solche Formatierungen Dokument- oder Website-übergreifend zu steuern. Nachfolgend wird ein Beispiel gezeigt, wie man die Überschrift `<h1>` generell auf „bold“ setzt und für ausgewählte Überschriften mit dem Tag `<h1>` die Textfarbe zusätzlich auf Rot ändert. Möchte man die Farbe auf Blau ändern, muss nur eine einzige Anweisung in der CSS-Datei geändert werden.

Ein Eintrag in einer CSS-Datei besteht immer aus einem oder mehreren Selektoren und einem Deskriptor. Der Selektor legt fest, auf welche HTML-Elemente die CSS-Anweisung anzuwenden ist, der Deskriptor beschreibt die Anweisung.

Beispiel, um allen `h1`-Überschriften den Schriftschnitt „fett“ zuzuordnen:

```
h1 {  
    font-weight:bold;  
}
```

Vor den geschweiften Klammern steht der **Selektor**, zwischen den Klammern der **Deskriptor**.

Um HTML-Elemente gezielt in einer CSS-Anweisung anzusprechen, gibt es weitere Selektoren. Im Beispiel wird ein Klassenselektor (beginnt in der CSS-Datei mit einem Punkt) eingeführt:

```
.color {  
    color:red;  
}
```

Alle HTML-Elemente, die die Klasse `color` aufweisen, werden damit angesprochen und rot gefärbt. Damit die Klassenauszeichnung `color` wirksam wird, muss dem `h1`-Element mit dem Text „Grundlagen Druckvorstufentechnik“ diese Klasse zugewiesen werden:

```
<h1 class="color">Grundlagen Druckvorstufentechnik</h1>
```

Ergebnis (wie zuvor bei der Formatierung direkt im `h1`-Tag):

**Grundlagen Druckvorstufentechnik**

## Einbinden von CSS-Information

CSS-Styleanweisung können entweder im Head eingebunden (`style`-Element) oder in einer separaten CSS-Datei (`link`-Element, ► Seite 1) gespeichert werden.

### Styleanweisung im Head

Beispiel

```
<head>  
    <title>Styleanweisung im Head</title>  
    <style type="text/css">  
        p {  
            color:red;  
            font-weight:bold;  
        }  
    </style>  
</head>
```

Diese Anweisung weist allen Absätzen `<p>` Fettschrift zu.

Die im Head auf diese Weise hinterlegte CSS-Anweisung gilt für das ganze HTML-Dokument.

## Styleanweisung in CSS-Datei

Wird eine eigenständige CSS-Datei angelegt und per [link](#)-Element eingebunden, gelten deren Anweisungen für alle HTML-Dokumente, die auf diese CSS-Datei verweisen.

Das Einbinden externer CSS-Dateien umfasst 2 Schritte:

1. Im Head der HTML-Datei wird auf die CSS-Datei (hier: „cssmuster.css“) verwiesen.

```
<head>
  <title>Styleanweisung in eigener Datei</title>
  <link rel=stylesheet type="text/css"
    href="cssmuster.css">
</head>
```

2. Anlegen der CSS-Datei mit der Regel für das HTML-Element [h1](#) und die Klasse [.color](#)

```
h1 {
  font-weight:bold;
}
.color {
  color:red;
}
```

## 7.4 SVG: Grundlagen und Beispiele

- SVG-Dateien folgen dem Prinzip von XML-Dateien. Sie bestehen aus einem Head und einem Rootelement (hier: `<svg>...</svg>`).
- CSS wird unterstützt.
- CSS-Anweisungen können nach 3 Methoden angelegt werden (► 7.3, Seite 85):
  - ▷ direkt im Geometrie- oder Textelement ([style](#)-Attribut, „Inline-Style“),
  - ▷ im Head ([style](#)-Element),
  - ▷ in einer CSS-Datei ([link](#)-Element).
- Es können Entitäten definiert werden, die sich später im Dokument wiederholt aufrufen lassen (z. B. mehrfach verwendete Grafikelemente). (Ähnlich: indirekte Objekte in einer PDF-Datei, ► Seite 109)
- Die Elemente einer Datei werden wie bei PostScript (► Seite 92) der Reihe nach gezeichnet. Nachfolgend beschriebene Elemente werden daher über die vorhergehenden gelegt. Im Gegensatz zu PostScript unterstützt SVG Angaben zur Transparenz (Attribut `opacity`), sodass es möglich ist, darunterliegende Elemente durchscheinen zu lassen.
- Der Koordinatenursprung ist links oben (PostScript: links unten).
- Längeneinheiten wie in den CSS-Spezifikation:
  - em (Buchstabenhöhe bei Standardschriftgröße), ex (Höhe des Buchstaben X),
  - px,
  - mm, cm, in, pt (72 pt = 1 in), pc (6 pc = 1 in; 1pc = 12 pt)
- Wenn keine Einheit angegeben wird, gelten die Angaben in px.
- Transformationen: Skalierung, Translation, Rotation, Schrägstellen sowie Matrixoperationen.
- Fast jedes Objekt kann beschnitten werden oder als Beschneidungspfad dienen (auch Textobjekte).
- Pfade können Löcher enthalten und aus mehreren nicht verbundenen (disjunkten) Objekten bestehen (z. B. ein Land mit Inseln).
- An Stützpunkten einer Linie können Markersymbole stehen, an den Endpunkten Pfeile.

- **Linieeneigenschaften:**  
Breite, Farbe, Opazität, Strichelungsart, Endtyp (butt [stumpf], square [um die halbe Linienstärke verlängert], round [runder Abschluss, Radius = halbe Linienstärke], Verbindungstyp (miter [Gehrung], bevel [abgeflacht], round)
- **Füllungseigenschaften:**  
gleichförmiger Farbwert (solid fill), Farbverlauf (linear oder radial; unterschiedliche Stoppwerte und Ausbreitungsmethoden), Opazität (zum Erzeugen von Transparenz), Füllungsregel für Löcher im Pfad (non-zero winding, even-odd, ► Seite 102).
- **Text:**  
Kann wie andere Basisgeometrien formatiert werden (s.o.).  
Unicode, bidirektional, rechts-links möglich.  
Schriftfestlegungen: Art, Gewicht, Schnitt, Lage, Größe, Spationierung, Kerning.  
Wichtig: SVG 1.0 erlaubt nur einzeiligen Text.  
Glyphen oder ganze Schriften können eingebunden werden.  
Unterschiedliche Grundlinien/Text auf Pfad sind möglich, einzelne Zeichen sind rotierbar.
- **SVG kennt Filtereffekte (auf Pixel und Vektoreffekte):**  
Lichteffekte, Schärfe-/Unschärfefilter, Überblenden, Kombinieren, Fluten, Zusammenfügen, Morphologie, Turbulenz usw.
- **SVG kennt Animationen:**  
Farbe, Geometrie, Transformation, Ort usw. können verändert werden.  
Objekte können an einem Pfad entlang wandern.  
Syntax: SMIL: Synchronized Multimedia Integration Language
- **SVG-Grafiken können interaktiv sein.**
- **SVG-Objekte können gruppiert werden:** <g>...</g>

Die Abbildungen 7.4.2 bis 7.4.4 zeigen Beispiele für Grafik, Text und Animation in SVG.

|                                                                                     |                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <code>&lt;circle cx="25" cy="25" r="20" fill="red"/&gt;</code>                                                                                                                                                | <b>Kreis:</b> cx,cy (Mittelpunkt), r (Radius)                                                                                                                                                                                                                          |
|  | <code>&lt;rect x="5" y="5" width="40" height="40" fill="red"/&gt;</code>                                                                                                                                      | <b>Rechteck:</b> x,y (linke obere Ecke), width, height (Breite, Höhe)                                                                                                                                                                                                  |
|  | <code>&lt;rect x="5" y="5" rx="5" ry="5" width="40" height="40" fill="red"/&gt;</code>                                                                                                                        | <b>Rechteck mit abgerundeten Ecken:</b> rx,ry (Eckenradius horizontal/vertikal)                                                                                                                                                                                        |
|  | <code>&lt;ellipse cx="25" cy="25" rx="20" ry="10" fill="red"/&gt;</code>                                                                                                                                      | <b>Ellipse:</b> cx,cy (Mittelpunkt), rx,ry (horizontaler/vertikaler Radius)                                                                                                                                                                                            |
|  | <code>&lt;line x1="5" y1="5" x2="45" y2="45" stroke="red"/&gt;</code>                                                                                                                                         | <b>Linie:</b> x1,y1 (Anfangs-), x2,y2 (Endpunkt); ohne Definition von stroke und stroke-width: schwarz, 1px                                                                                                                                                            |
|  | <code>&lt;polyline points="5,5 45,45 5,45 45,5" stroke="red" fill="none"/&gt;</code>                                                                                                                          | <b>offener Polygonzug:</b> x- und y-Koordinaten der Punkte; ohne fill="none" schwarz gefüllt                                                                                                                                                                           |
|  | <code>&lt;polygon points="5,5 45,45 5,45 45,5" stroke="red" fill="none"/&gt;</code>                                                                                                                           | <b>geschlossener Polygonzug:</b> x- und y-Koordinaten der Punkte; ohne fill="none" schwarz gefüllt                                                                                                                                                                     |
|  | <code>&lt;path d="M5,5 C5,45 45,45 45,5" stroke="red" fill="none"/&gt;</code>                                                                                                                                 | <b>kubische Bézierkurve:</b> M5,5 (Startpunkt; M = moveto); C5,45 5,45 (2 Kontrollpunkte, blau; C = curveto) 45,5 (Endpunkt); großes C, M: absolute Koordinaten; kleines c, m: relativ zum vorherigen Punkt                                                            |
|  | <code>&lt;defs&gt;<br/>  &lt;rect id="rect" width="15" height="15" fill="red"/&gt;<br/>&lt;/defs&gt;<br/>&lt;use x="5" y="5" xlink:href="#rect"/&gt;<br/>&lt;use x="30" y="30" xlink:href="#rect"/&gt;</code> | <b>Elemente, die mehrfach verwendet werden, werden zunächst definiert (&lt;defs&gt;) und erhalten eine id (hier: id="rect"). Zur Verwendung werden sie mit &lt;use&gt; abgerufen; nach den Koordinaten der linken oberen Ecke folgt der Link (xlink:href="#rect").</b> |

Abb. 7.4.2 | Grafikelemente in SVG

|                                                                                   |                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Text</b>                                                                       | <code>&lt;text x="0" y="13" fill="red" text-anchor="start"&gt;Text&lt;/text&gt;</code>                                                                                                                                                                                                                                                | <b>x,y</b> sind die Koordinaten des ersten Buchstabens. Da <b>y</b> die Grundlinie ist, darf der Wert nicht 0 sein, weil der Text sonst nach oben aus dem Viewport (🔴 Seite 90) ragt. Textausrichtung: links (start, left), zentriert (middle), rechts (end). Kein Blocksatz, weil Text nur einzeilig ist. Rechtsbündig: x darf nicht 0 sein, weil der Text sonst nach links aus dem Viewport ragt. |
| <b>Text</b>                                                                       | <code>&lt;text x="25" y="13" fill="red" text-anchor="middle"&gt;Text&lt;/text&gt;</code>                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Text</b>                                                                       | <code>&lt;text x="50" y="13" fill="red" text-anchor="end"&gt;Text&lt;/text&gt;</code>                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Zeile 1</b><br><b>Zeile 2</b>                                                  | <code>&lt;text x="0" y="13" fill="red"&gt;<br/>  &lt;tspan&gt;Zeile 1&lt;/tspan&gt;<br/>  &lt;tspan x="0" dy="1em"&gt;Zeile 2&lt;/tspan&gt;<br/>&lt;/text&gt;</code>                                                                                                                                                                  | <b>Kodierung für mehrzeiligen Text</b>                                                                                                                                                                                                                                                                                                                                                              |
|  | <code>&lt;defs&gt;<br/>  &lt;path id="textPath"<br/>    d="M10,50 C10,0 90,0 90,50" /&gt;<br/>&lt;/defs&gt;<br/>&lt;use xlink:href="#textPath"<br/>  stroke="blue" fill="none" /&gt;<br/>&lt;text fill="red"&gt;<br/>  &lt;textPath xlink:href="#textPath"&gt;<br/>    Text auf Pfad<br/>  &lt;/textPath&gt;<br/>&lt;/text&gt;</code> | <b>Text auf Pfad:</b> Zunächst wird eine kubische Bézierkurve als Textpfad definiert (path id="textpath") und über use aufgerufen, um sie blau zu färben. Dann wird dem Text der Textpfad zugewiesen.                                                                                                                                                                                               |

Abb. 7.4.3 | Textelemente in SVG

|                                                                                     |                                                                                                                                                                                                                         |                                                                                        |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
|   | <code>&lt;rect y="45" width="10" height="10" fill="red"&gt;<br/>  &lt;animate attributeName="x" from="0" to="90" dur="10s" repeatCount="indefinite" /&gt;<br/>&lt;/rect&gt;</code>                                      | <b>horizontal wanderndes Rechteck</b>                                                  |
|  | <code>&lt;rect y="45" width="10" height="10" fill="red"&gt;<br/>  &lt;animate attributeName="y" from="0" to="90" dur="10s" repeatCount="indefinite" /&gt;<br/>&lt;/rect&gt;</code>                                      | <b>vertikal wanderndes Rechteck</b>                                                    |
|  | <code>&lt;rect x="-5" y="-5" width="10" height="10" fill="red"&gt;<br/>  &lt;animateMotion<br/>    path="M10,50 C10,0 90,0 90,50"<br/>  dur="10s" repeatCount="indefinite" /&gt;<br/>&lt;/rect&gt;</code>               | <b>auf Bézierkurve wanderndes Rechteck, Ausrichtung konstant</b>                       |
|  | <code>&lt;rect x="-5" y="-5" width="10" height="10" fill="red"&gt;<br/>  &lt;animateMotion<br/>    path="M10,50 C10,0 90,0 90,50"<br/>  dur="10s" rotate="auto" repeatCount="indefinite" /&gt;<br/>&lt;/rect&gt;</code> | <b>auf Bézierkurve wanderndes Rechteck, Ausrichtung automatisch an Kurve angepasst</b> |

Abb. 7.4.4 | Animationen in SVG

## Die 2 Koordinatensysteme in SVG: Viewport und View Box

SVG kennt 2 Koordinatensysteme:

- ▶ Viewport (Gerätekoordinatensystem)
- ▶ View Box (Userkoordinatensystem)

### Viewport

SVG-Elemente können eine beliebige Größe haben. Sichtbar ist aber nur der Teil, der im Viewport definiert wird:

```
<svg width="500px" height="300px"></svg>
```

Man sieht in diesem Beispiel 500 x 300 Pixel der SVG-Elemente.

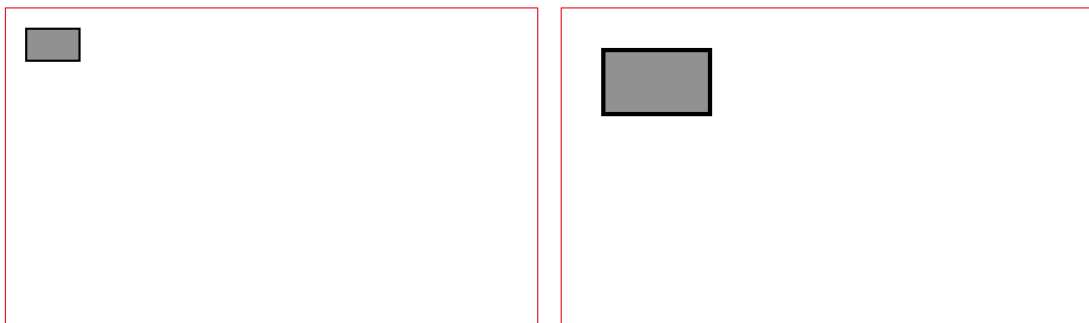
Das folgende Rechteck liegt somit komplett im Viewport (Abbildung 7.4.1, links):

```
<rect x="20px" y="20px" width="50px" height="30px" stroke="#000000"
stroke-width="2px" fill="#999999" />
```

### View Box

Wird zusätzlich eine View Box definiert, sieht die SVG-Datei z. B. so aus (Abbildung 7.4.1, rechts):

```
<svg width="500" height="300" viewBox="0 0 250 150">
  <rect x="20px" y="20px" width="50px" height="30px" stroke="#000000"
  stroke-width="2px" fill="#999999" />
</svg>
```



Ohne ViewBox wird das graue Rechteck mit seinen ursprünglichen Koordinaten in dem Viewport (rot umrandet) ausgegeben.

Die Definition einer ViewBox bewirkt hier eine Skalierung um den Faktor 2. Auch die Linienstärke wird verdoppelt. Der Viewport hat dieselbe Größe wie links.

Abb. 7.4.1 | Einfluss einer ViewBox auf die Ausgabe einer Grafik in SVG.

Mithilfe einer View Box (beachte die Schreibweise `viewBox` im Quellcode!) kann eine Grafik z. B. ohne Neuberechnung skaliert werden. Beide Koordinatensysteme können unterschiedliche Einheiten verwenden.

</Ø>

## <!> Zusammenfassung „SVG“

- SVG ist eine textbasierte Auszeichnungssprache (Markup Language), mit der sich zweidimensionale skalierbare Grafiken erstellen lassen. Neben Vektorelementen können Pixelbilder und Textelemente enthalten sein. SVG beruht auf XML. Skript- und Stylesprachen werden unterstützt.
- Im Unterschied zu PostScript können Objekte in SVG transparent sein.
- SVG unterstützt Animationen.

### <∅>

- Der ISO-Standard SGML schließt XML komplett ein. XHTML und HTML sind Tochtersprachen zu SGML. HTML5 gilt nicht mehr als Tochtersprache zu SGML.
- HTML ist die Sprache für das Web. Der im Oktober 2014 fertiggestellte Standard HTML5 (aus HTML 4.01 und XHTML 1.0 weiterentwickelt) hat das Ziel, die *strukturelle (=semantische)* Auszeichnung im HTML-Dokument von der Beschreibung des *Aussehens* in einer CSS-Datei zu trennen. Syntaxfehler im HTML-Code werden von Browsern teilweise toleriert.
  - ▷ HTML-Dokumente beginnen mit einer Document Type Definition.
  - ▷ Darauf folgt das Element `<html>`, das den ganzen Rest der Datei umschließt.
  - ▷ Kindelemente dieses Elements sind `<head>` und `<body>`. `<head>` enthält Metainformationen (z. B. die Angabe zum Zeichensatz) und zwingend das Element `<title>`.
- XML ist eine Teilmenge von SGML. Tags werden ausschließlich zur Strukturierung des Texts verwendet, z.B. in einer XML-Datenbank. Syntaxfehler führen zum Abbruch der Interpretation. XML-Datenbanken eignen sich zur automatischen Katalogproduktion (mithilfe von Skripts oder Plugins zu InDesign und Quark Xpress).
  - ▷ Die Syntax von XML (bzw. XHTML) weicht teilweise von HTML ab. Beispiel: Jedes Tag muss in XML geschlossen werden, während leere Tags in HTML kein Endtag erfordern. HTML5 erlaubt die Syntax beider Sprachen nebeneinander. Groß- und Klein-Schreibung in Tags werden in XML beachtet, in HTML nicht.
  - ▷ Jedes XML-Dokument enthält genau 1 Rotelement, das beliebig benannt werden darf (in SVG: `<svg>`)
- Die Stylesheet-Sprache CSS wird von HTML und XML (und damit auch SVG) unterstützt. Die Einbindung von CSS-Dateien hat das Ziel, Inhalt einer Seite und Anweisungen für die Darstellung zu trennen.

### </∅>

---



## 8 PostScript

### Übersicht

- 8.1 Einführung
  - 8.2 Struktur einer PostScript-Datei
  - 8.3 Grundlegende Eigenschaften von PostScript
  - 8.4 Programmieren in PostScript (Beispiele)
  - 8.5 EPS-Dateien
  - 8.6 PostScript-Schriften
- 

### 8.1 Einführung

PostScript ist eine Seitenbeschreibungssprache (PDL, Page Description Language). Die erste Version wurde 1984 von Adobe vorgestellt. Im Jahr 1991 kam die erste Erweiterung. Sie wurde als „PostScript Level 2“ bezeichnet. Level 2 ermöglichte u. a. das Einbinden von JPEG-Bildern. Eine weiterer wesentlicher Fortschritt war die Implementierung von Superzellen für die Rasterung (🔴 Seite 132), wodurch sich die geforderten Rasterwinkel und Rasterweiten viel genauer annähern ließen. Die maximale Bittiefe pro Kanal in Bilddaten stieg von 8 auf 12.

„PostScript 3“ (inoffiziell auch als „Level 3“ bezeichnet) kam 1997. Mit dieser Version wurde u. a. das Farbmodell DeviceN aufgenommen, das auch Hexachrom und Duplexfarben unterstützt. Die Rasterung wurde weiter verbessert.

Mit Version PostScript 3 wurde die Weiterentwicklung zugunsten von PDF eingestellt. PDF enthält wesentliche Elemente von PostScript.

- Eine PostScript-Datei (oft auch als PostScript-Programm bezeichnet) kann (wie z. B. SVG) enthalten:
  - ▷ Text,
  - ▷ Vektorgrafik,
  - ▷ Pixelgrafik (Bit-/Bytemap).  
Fotos oder andere digitale Bilder, die in üblichen Formaten wie JPEG, TIFF, PNG usw. vorliegen, können eingebettet werden.  
Die Rasterung von Pixelbildern wird vielfältig unterstützt.
- PostScript-Dateien lassen sich über die Kommandozeile („Eingabeaufforderung“ in Windows, „Terminal“ in MacOS) direkt an PostScript-fähige Drucker/RIPs schicken. Dort werden sie für die Ausgabe gerendert.
- Neben der Seitenbeschreibung (alle Elemente einer Seite, Seitenformat, Informationen zu Farben) können auch Transferkurven, Informationen zu Separation und Rasterung sowie Schriften (soweit die Lizenz des Herstellers das zulässt) eingebunden werden.
- Zum Editieren des Inhalts (z. B. Text auf der Seite) sind PostScript-Dateien nicht geeignet. Ein (sinnvoll nutzbarer) Rückweg ins Layoutprogramm existiert nicht.

- PostScript-Dateien werden üblicherweise aus Layoutsoftware heraus über den Druckdialog erstellt. Sie können geräteunabhängig sein oder durch Auswahl einer PPD-Datei (siehe unten) an das Ausgabegerät angepasst werden.
- PostScript-Dateien können auch manuell in einem Texteditor erstellt werden.

## PPD-Dateien

- Über PPD-Dateien (PostScript Printer Description) können Merkmale des Ausgabegeräts direkt angesprochen werden:
  - ▷ PPD-Dateien enthalten Informationen über PostScript-Drucker und andere PostScript-Ausgabegeräte (insbesondere Belichter), z. B. unterstützte PostScript-Version, Auflösung, Seiten-/Filmformate usw.
  - ▷ PPD-Dateien sind formalisierte Textdateien entsprechend der PPD-Spezifikation von Adobe. Sie sind selbst nicht in PostScript geschrieben.
  - ▷ PPD-Dateien werden zum Teil mit dem Rechnerbetriebssystem installiert, zum Teil sind sie auf der Homepage der Gerätehersteller verfügbar.
  - ▷ Druckertreiber und PPD sind softwaretechnisch zwei Komponenten, die bei der Erstellung einer PostScript-Datei zusammenwirken:
    - Der PostScript-Druckertreiber liest Informationen aus der PPD-Datei.
    - PostScript-Druckertreiber generieren unter Berücksichtigung der PPD-Datei für den jeweiligen Drucker optimierten PostScript-Code.

Hinweis:

Mit dem Prinzip der PPD-Dateien wurde die ursprüngliche Absicht aufgegeben, mit PostScript geräteunabhängigen (vor allem unabhängig von der Auflösung des Ausgabegeräts) Code zu erzeugen.

## PostScript-Verarbeitung im RIP (Raster Image Processor)

Das Prinzip der Verarbeitung von PostScript-Code in einem RIP zeigt Abbildung 8.1.1.

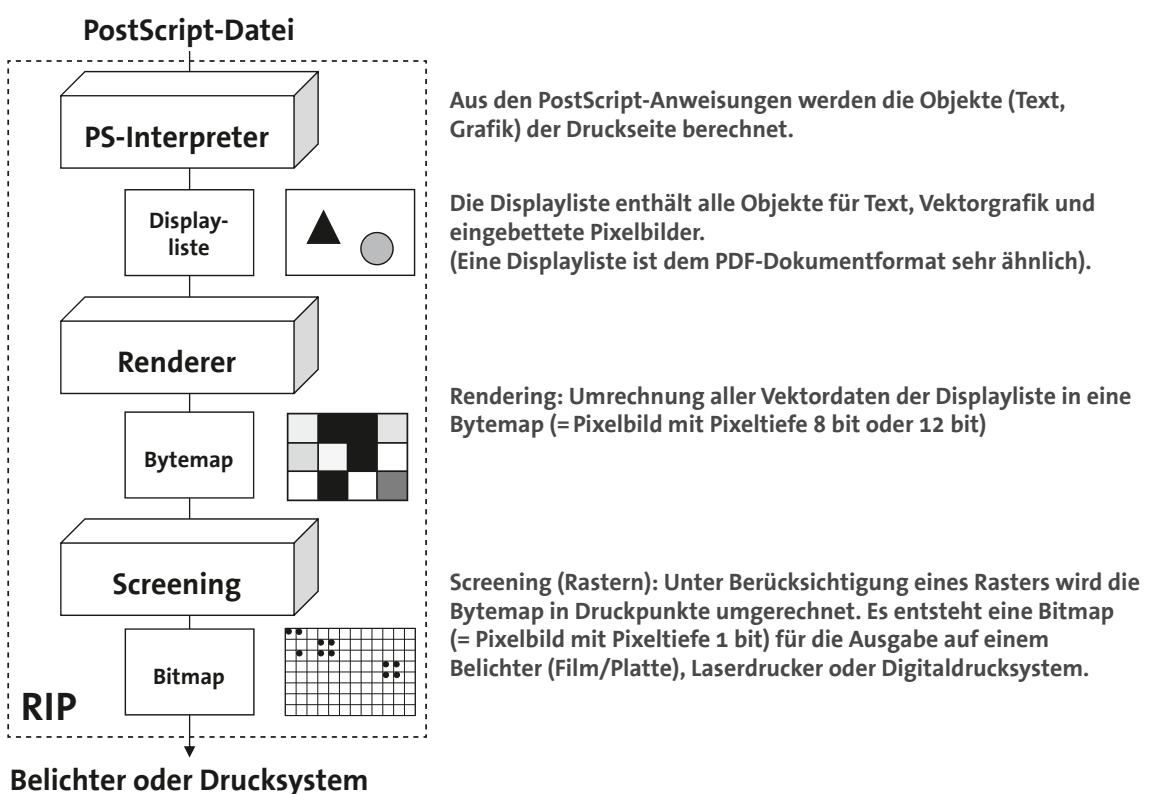


Abb. 8.1.1 | Prinzip der Verarbeitung einer PostScript-Datei im Raster Image Processor (RIP)

## Warum heute noch PostScript lernen?

- Um sich in einer PostScript-Datei zurechtzufinden und mögliche Problemstellen zu erkennen und korrigieren
- PDF enthält einige Elemente von PostScript bzw. baut auf PostScript auf (z. B. Rastertechnologie).
- Ein Teil der Schriften nutzt PostScript für die Beschreibung der Buchstabenformen (🔴 Seite 103)
- Einige Prinzipien von PostScript finden sich z. B. in SVG (Scalable Vector Graphics, ) wieder.

## Literatur zu PostScript

- ▷ Red Book: <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>
- ▷ Green Book: <http://www-cdf.fnal.gov/offline/PostScript/GREENBK.PDF>
- ▷ Blue Book: <http://www-cdf.fnal.gov/offline/PostScript/BLUEBOOK.PDF>
- ▷ First Guide to PostScript, von Peter J. Weingartner:  
<http://www.tailrecursive.org/postscript/postscript.html>  
[http://dept-info.labri.u-bordeaux.fr/~ges/ENSEIGNEMENT/GEOMETRIE/postscript\\_level1.pdf](http://dept-info.labri.u-bordeaux.fr/~ges/ENSEIGNEMENT/GEOMETRIE/postscript_level1.pdf)
- ▷ Information zu PostScript-Schriften:
  - [http://partners.adobe.com/public/developer/en/font/T1\\_SPEC.PDF](http://partners.adobe.com/public/developer/en/font/T1_SPEC.PDF)
  - <http://blogs.adobe.com/typblography/files/typblography/TT%20PS%20OpenType.pdf>



**Ab hier bis zum Ende von Kapitel 8 nicht Stoff des Semesters!**

## 8.2 Struktur einer PostScript-Datei

PostScript-Dateien sollten (müssen aber nicht) strukturiert werden. Typische Strukturelemente sind der Gebrauch von *Document Structuring Conventions* (DSC-Kommentaren) zu Beginn der Datei sowie die Einteilung in *Prolog* und *Script*.

Hinweis:

Jede PostScript-Datei sollte mit

%!

beginnen. Damit wird einem Drucker oder Spooler signalisiert, dass PostScript-Code folgt.

Andernfalls wird u. U. der PostScript-Code als Text ausgegeben, z. B. wenn man eine PostScript Datei direkt an eine Druckerschnittstelle sendet (copy unter Windows, lp am Mac.)

### ■ DSC-Beispiel:

▷ %!PS-Adobe-3.1

Generell werden mit dem Zeichen „%“ Kommentare eingeleitet, die an jeder Stelle beginnen dürfen und vom Interpreter nicht beachtet werden.

Einfache Kommentare beginnen mit „%“, DSC-Kommentare mit „%%“.

▷ %%Title:

%%Creator:

%%CreationDate:

%%BoundingBox: (und viele weitere)

Die BoundingBox beschreibt die Ausdehnung der in der PostScript-Datei enthaltenen Elemente. Beispiel: 0 0 200 140 definiert eine BoundingBox, die bei den Koordinaten 0, 0 (ll = Lower left) beginnt und bei 200, 140 (ur = upper right) endet. Wird eine EPS-Datei (🔴 Seite 102) in einem Layoutprogramm platziert, berechnet das Layoutprogramm den für die Platzierung der EPS-Datei nötigen Rahmen anhand der BoundingBox.

▷ %%BeginProlog

%%EndProlog

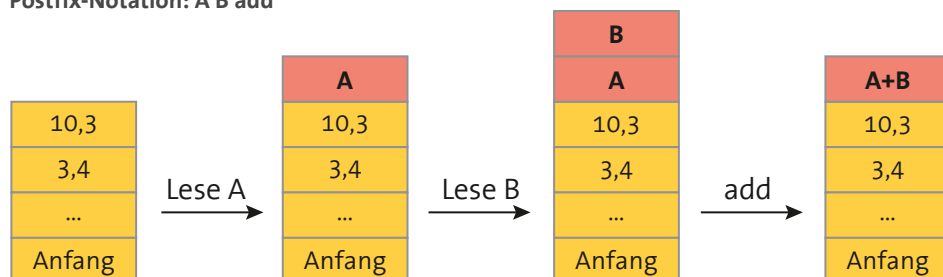
Zwischen den Zeilen „BeginProlog“ und „EndProlog“ stehen allgemeine Definitionen, vor allem Befehlsabkürzungen (Prozeduren). Sie werden in den Speicher geladen und gelten für das ganze Dokument.

- ▷ %%Page: x y (z. B. 1 von 3)
- %%EndPageComments
- %%BeginPageSetup (z. B. PageSize)
- %%EndPageSetup
- Nach dieser Zeile beginnt das Script der Seite.
- Die Seitenbeschreibungsbefehle werden sequenziell abgearbeitet.
- %%PageTrailer
- showpage (Befehl zur Ausgabe der Seite)
- ▷ %%Trailer
- %%Pages: x
- %%DocumentProcessColors: Cyan Magenta Yellow Black
- ...
- ▷ %%EOF (End of File)

## 8.3 Grundlegende Eigenschaften von PostScript, Syntax

- Konzeptioneller Ansatz:  
Ausgangspunkt ist eine imaginäre Fläche (z. B. weiße A4-Seite), auf der mit einem imaginären Stift gezeichnet wird.
- Sprachbefehle:
  - ▷ Programmierung: Vergleiche, Schleifen, Variablen, ...
  - ▷ Zeichnen: Linien, Flächen, Grafiken, Text, Muster, ...
  - ▷ Druckvorstufenprozesse: Rasterung, Farbe, Überfüllung, Datenkompression, ...
- Ein *Dictionary* ist eine Sammlung von Name-Wert-Paaren.
- PostScript arbeitet mit *Stapeln* (*Stacks*).
  - ▷ Es gibt mehrere Stapel, u. a.: operand stack, dictionary stack.
  - ▷ Stapel werden vom Interpreter sequenziell abgearbeitet.
- PostScript folgt dem Prinzip der *Postfix-Notation* (= Reverse Polish Notation, RPN), wie sie von manchen Taschenrechnern oder Rechen-Apps bekannt ist:  
Auf den Stapel werden zuerst die Operanden gelegt, gefolgt von einem Operator (Abb. 8.3.1).

Beispiel mit den beiden Operanden „A“ und „B“ sowie dem Operator „add“:  
Postfix-Notation: A B add



Der Interpreter legt die Variablen „A“ und „B“ auf den Stack. Der Operator „add“ nimmt die beiden oberen Einträge vom Stack und bildet die Summe. Die Summe wird oben auf den Stack gelegt.

Abb. 8.3.1 | Funktion eines Stapels in PostScript.

Da Befehle sequenziell abgearbeitet werden, entscheidet die Reihenfolge, in der Grafikobjekte beschrieben werden, über das Ergebnis (Abb. 8.2.2).

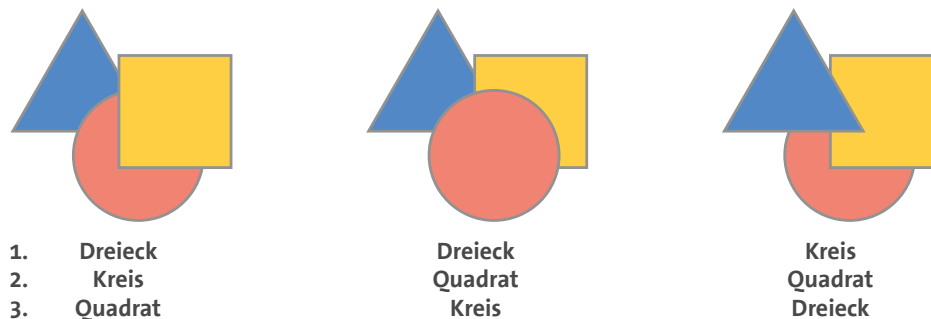


Abb. 8.3.1 | Einfluss der Reihenfolge, mit der die Objekte in der PostScript-Datei geschrieben sind, auf das Ergebnis.

❧ PostScript kann keine Transparenz beschreiben! Vor dem Erstellen einer PostScript-Datei wird Transparenz (z. B. ein Schlagschatten) daher immer reduziert. Im Gegensatz dazu kennt z. B. SVG (🔴 Seite 87) das Attribut „Opacity“.

- Die Basiseinheit ist pt (1/72 Inch = 0,35278 mm)  
 Jede andere Einheit kann mithilfe einer *Procedure* (🔴 Seite 97) definiert werden, z. B. inch:  
 /inch {72 mul} def  
 (Als Procedure wird festgelegt, dass ein Zahlenwert durch „inch“ mit 72 multipliziert wird).  
 Hinweis:  
 Die Einheit PostScript-Point weicht vom typografischen Punkt (Didot-Punkt: 0,376 mm)  
 und vom traditionellen amerikanischen Pica Point (0,351 mm) ab.
- *Koordinatensystem:*
  - ▷ User Space: Geräteunabhängiges kartesisches Koordinatensystem  
 Der Ursprung 0,0 liegt links unten (SVG: links oben).
  - ▷ Device Space: Gerätekoordinatensystem
- *Matrizen:*  
 Wenn nicht anders definiert, arbeitet PostScript mit einer 3 x 3 Matrix, die Rotation, Skalieren und Transformation erlaubt.
- Errechnete *Pfade* sind imaginäre Linien, die erst durch Befehle wie „stroke“ oder „fill“ sichtbar werden.
- Jeder Pfad kann unabhängig von Kontur und Füllung als Beschneidungspfad dienen („clip“). Auch Text kann als Beschneidungspfad dienen.
- *Namen:*  
 Ein Name ist eine Referenz auf einen bestimmten Wert in einem Dictionary des Dictionary Stack.  
 Als Name kann jede beliebige Zeichenfolge fungieren, die nicht als Zahl interpretierbar ist. Nicht als Name verwendet werden dürfen Leerschritte sowie die Zeichen ( ) [ ] { } < > / %  
 Namen dürfen theoretisch auch mit einer Ziffer beginnen. Das sollte aber vermieden werden, weil z. B. 1E10 eine reelle Zahl verkörpert (siehe unten).
- *Zahlen:*  
 In PostScript sind alle reellen („real“) Zahlen möglich. Dazu gehören auch ganze Zahlen („Integer“). Negatives Vorzeichen ist möglich.  
 Zahlen können als Dezimalzahlen (z. B. 0.5) oder in der Form *Mantisse E Exponent* (Scientific

Notation, z. B.  $2340 = 2.34 \times 10^3 = 2.34\text{E}3$ ) eingegeben werden. Ferner sind Zahlenangaben in der Form `2#1011110` oder `16#7FFF` möglich (vor der Raute steht jeweils die Basis).

■ **Strings:**

Als String wird eine Buchstabenfolge bezeichnet, die als Text ausgegeben werden soll. Sie steht zwischen runden Klammern:

(Grundlagen Druckvorstufentechnik)

■ Boolesche Operatoren sind möglich: `true`, `false`

■ Als *Escape-Code* wird Code bezeichnet, der keinen Text, sondern ein Kommando kodiert. Beispiele:

```
\n      new line
\r      carriage return
\tab    Tabulator
```

■ **Arrays:**

Ein Array ist eine Gruppe von Objekten, die zwischen eckigen Klammern stehen: z. B.

```
[12 /box 5]
```

(drei Elemente: die Zahl „12“, der Name „box“ und die Zahl „5“).

■ **Procedures:**

Als Procedure wird ein Array von ausführbaren Befehlen bezeichnet.

Beispiel:

Um den obersten Wert des Stack zu quadrieren, können die beiden Befehle

```
{dup mul}
```

als Array geschrieben werden. Dieses Array wird als neuer Operator „square“ definiert:

```
/square {dup mul} def
```

Diese Definition wird oben auf den Dictionary Stack gelegt.

Anderes Beispiel:

```
/ZeichneSenkrechteLinie {0 200 rlineto} def
```

(Beachte die Ähnlichkeit zu Deskriptoren in CSS, ► Seite 85, die ebenfalls zwischen geschweiften Klammern stehen.)

■ **Variablen:**

```
/A 20 def
```

legt die Variable A mit dem Wert 20 im User Dictionary an.

## 8.4 Programmieren in PostScript (Beispiele)

### 8.4.1 Zeichnen von Pfaden

Für das Zeichnen eines Rechtecks sind folgende Befehle erforderlich:

- ▶ `newpath` (beginnt einen neuen Pfad)
- ▶ `moveto` (verlegt den aktuellen Punkt [current point], Angabe in absoluten Koordinaten)
- ▶ `rmoveto` (verlegt den aktuellen Punkt, Angabe in relativen Koordinaten)
- ▶ `lineto` (zeichnet eine Linie vom aktuellen Punkt aus, Zielangabe in absoluten Koordinaten)
- ▶ `rlineto` (zeichnet eine Linie vom aktuellen Punkt aus, Zielangabe in relativen Koordinaten)
- ▶ `closepath` (verbindet den aktuellen Punkt mit dem ersten Punkt des Pfades auf dem kürzesten Weg)
- ▶ `stroke` (stellt den Pfad dar; ohne weitere Angaben: 1 pt dicke schwarze Linie  
  `2 setlinewidth` setzt die Linienstärke auf 2 pt  
  `0.9 setgray` setzt die Linienfarbe auf 10 % grau.
- ▶ `showpage` (gibt die Seite aus)



Abbildung 8.4.1.1 zeigt das Zeichnen eines Rechtecks.

---

```
%!
/inch {72 mul} def      %definiert inch als Maßeinheit
newpath                %Beginn des neuen Pfades
1 inch 1 inch moveto    %aktueller Punkt auf die Koordinaten 1 inch, 1 inch gelegt
2 inch 1 inch lineto    %Linie vom aktuellen Punkt zum Punkt mit den Koordinaten
                        % 2 inch, 1 inch, d.h. um ein inch nach rechts
2 inch 2 inch lineto    %Linie weiter um 1 Inch nach oben
1 inch 2 inch lineto    %Linie weiter um 1 inch nach links
closepath              %schließt den Pfad
15 setlinewidth         %setzt Linienstärke auf 15 pt
.5 setgray              %setzt Linienfarbe auf 50% Grau
stroke
showpage
```

---



Statt mit `closepath` hätte man das Quadrat auch mit  
`1 inch 1 inch lineto`  
fertigstellen können. Dann wäre der Pfad aber nicht geschlossen.  
Das wirkt sich auf die Form der unteren linken Ecke (► Seite 100) aus:



---

**Abb. 8.4.1.1 | Zeichnen eines Rechtecks mit den Befehlen `moveto` und `lineto`.**

Abbildung 8.4.1.2 zeigt das Zeichnen des Rechtecks mit den Befehlen `rmove to` und `rlineto` statt `moveto` und `lineto`. Mit `rmove to` bzw. `rlineto` werden nicht absolute Koordinaten als Zielpunkt festgelegt, sondern der Abstand zum aktuellen Punkt (current point). (Der Befehl `currentpoint x y` legt den aktuellen Startpunkt auf die Koordinaten `x y`.)

---

```
%!
/inch {72 mul} def      %definiert inch als Maßeinheit
newpath                %Beginn des neuen Pfades
1 inch 1 inch rmove to  %currentpoint von den aktuellen Koordinaten um je 1 inch nach
                        % rechts und oben verschoben
1 inch 0 rlineto        %Linie vom aktuellen Punkt im 1 inch nach rechts
0 1 inch rlineto        %Linie weiter um 1 Inch nach oben
-1 inch 0 rlineto       %Linie weiter um 1 inch nach links
closepath              %schließt den Pfad
15 setlinewidth         %setzt Linienstärke auf 15 pt
.5 setgray              %setzt Linienfarbe auf 50% Grau
stroke
showpage
```

---

**Abb. 8.4.1.2 | Zeichnen eines Rechtecks mit den Befehlen `rmove to` und `rlineto`.**

Mit relative Koordinaten muss gearbeitet werden, um die Box als Objekt zu definieren, das wiederholt abrufbar ist (Abb. 8.4.1.3).

---

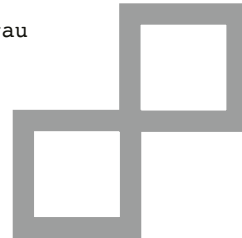
```

%!
/inch {72 mul} def          %definiert inch als Maßeinheit
/box {
    newpath
    moveto
    1 inch 0 rlineto
    0 1 inch rlineto
    -1 inch 0 rlineto
    closepath
} def                        %Definition der Box abgeschlossen
                               %Es folgt der dreimalige Aufruf der Box

15 setlinewidth
.5 setgray                  %setzt Linienfarbe auf 50% Grau
1 inch 1 inch box
stroke
2 inch 2 inch box
stroke
4 inch 4 inch box
stroke
showpage

```

---



**Abb. 8.4.1.3 | Definieren (def) einer Box mit relativen Koordinaten und dreimaliges Aufrufen mit absoluten Koordinaten.**

■ Die Befehle `gsave` und `grestore`:

Speichern z. B. eines Pfades für die spätere Wiederverwendung.

Abbildung 8.4.1.4 zeigt die Anwendung von `gsave` und `grestore`:

Die zuvor beschriebene Box, die nur eine Umrandung (`stroke`) bekommen hat, soll jetzt auch eine Füllfarbe (`fill`) erhalten. Zuerst wird der Pfad beschrieben. Bevor der Befehl `stroke` erfolgt, wird der Pfad mit `gsave` gesichert und nach dem Befehl `stroke` mit `grestore` wieder aufgerufen.

---

```

%!
/inch {72 mul} def
15 setlinewidth
.5 setgray
newpath
1 inch 1 inch moveto
2 inch 1 inch lineto
2 inch 2 inch lineto
1 inch 2 inch lineto
closepath
gsave                        %speichert den Pfad vor dem Ausführen des Befehls stroke
stroke
grestore                     %ruft den zuvor gespeicherten Pfad wieder auf
.2 setgray                  %setzt die Füllfarbe auf 80% Grau
fill
showpage

```

---



**Abb. 8.4.1.4 | Einsatz der Befehle `gsave` und `grestore`.**

■ Liniendefinitionen

▷ Linienenden:

```

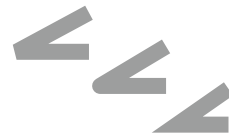
0 setlinecap    % stumpfe Enden (butt)
1 setlinecap    % runde Enden (round)
2 setlinecap    % stumpfe Enden plus halbe
                  Linienstärke (square)

```



#### ▷ Ecken

```
0 setlinejoin    % abgeschrägte Ecken (bevel)
1 setlinejoin    % runde Ecken (round)
2 setlinejoin    % Gehrung (miter)
```



#### ■ Farbdefinitionen

```
0 0 0 setrgbcolor    %schwarz
1 1 1 setrgbcolor    %weiß

0 0 0 0 setcmykcolor  %weiß
.4 .2 0 .1 setcmykcolor %hellblau
```

#### ■ Bézierkurven: PostScript verwendet nur kubische Bézierkurven (2 Kontrollpunkte,

##### ► Seite 73)

```
x1 y1 x2 y2 x3 y3 curveto    %kubische Bézierkurve vom current point
                               aus; x1 y1 und x2 y2 sind die beiden
                               Kontrollpunkte, x3 y3 ist der Endpunkt.
```

(curveto gibt es als rcurveto mit relativen Koordinaten.)

#### ■ Kreis, Kreisbogen

```
x y r angle1 angle2 arc      %zeichnet einen Kreisbogen um den Kreis-
                               mittelpunkt x,y mit dem Radius r von
                               Winkell1 bis Winkel2 gegen den
                               Uhrzeigersinn (arcn: im Uhrzeigersinn)
```

Beispiel Kreisbogen von 0 bis 270°:

```
%!
.5 setgray
10 setlinewidth
newpath
100 100 50 0 270 arc
stroke
showpage
(Für einen geschlossenen Kreis: von 0 bis 360°)
```



## 8.4.2 Text in PostScript

Folgendes Programm gibt das Wort „Druckvorstufe“ in 20 pt Times-Roman aus:

```
%!
/Times-Roman findfont    %findet Times-Roman im Wörterbuch
20 scalefont             %setzt Schrifgröße auf 20 pt
setfont                  %macht die Schrift zur aktuell verwendeten
newpath
72 150 moveto
(Druckvorstufe) show     %Text ist ein String zwischen Klammern;
                           show ist der Befehl für die Textausgabe

showpage
```

### 8.4.3 Beschneidungspfade (clip)

Ein Pfad kann als Beschneidungspfad z. B. für Text definiert werden (Abb. 8.4.3.1).

---

```
%!  
/box {  
  newpath  
  moveto  
  72 0 rlineto  
  0 72 rlineto  
  -72 0 rlineto  
  closepath  
} def  
/Times-Roman findfont 20 scalefont setfont  
gsave                                %Beschneidungspfad sichern  
72 72 box                            %Box aufrufen  
gsave                                %Box speichern, damit sie nach stroke noch  
                                     verfügbar ist  
  
  stroke  
grestore                             %Boxpfad wieder aufrufen  
clip                                 %Boxpfad zum Beschneidungspfad machen  
60 70 moveto                         %Startpunkt für Zeile 1  
(Times-Roman auf Box beschnitten) show  
75 95 moveto                         %Startpunkt für Zeile 2  
(Times-Roman auf Box beschnitten) show  
40 120 moveto                       %Startpunkt für Zeile 3  
(Times-Roman auf Box beschnitten) show  
grestore                             %Beschneidungspfad wieder aufrufen  
showpage                             %Seite ausgeben
```

---



Abb. 8.4.3.1 | Eine Box als Beschneidungspfad für Text.  
Links: Darstellung, wie oben programmiert: rechts: der beschnittene Text in Hellgrau hinterlegt.

### 8.4.4 Einbinden von Rastergrafik

Es müssen angegeben werden: Breite, Höhe, Bittiefe, Matrix, Daten.

Eine Matrix (als Array zwischen eckigen Klammern) ist erforderlich, um zu berücksichtigen, dass Koordinaten in PostScript links unten beginnen, Bilder aber meistens von links oben nach rechts unten beschrieben sind.

Nachfolgend das Beispiel eines Graukeils, der 16 Pixel in der Breite und 1 Pixel in der Höhe hat:

```
%!  
gsave  
216 72 translate                    %linke untere Bildecke bei (216, 72)  
72 72 scale                         %Größe des Bildes 72 x 72 Punkt  
16 1 4                             %Breite 16Pixel, Höhe 1Pixel,  
                                     Bittiefe 4 (erlaubt 16 Graustufen)  
[16 0 0 1 0 0]                    %Array für Transformation der  
                                     Bildpixel  
{<0123456789ABCDEF>}              %Bilddaten in hex (nur 1 Zeile, weil  
                                     nur 1 Pixel hoch)  
image                               %Befehl zur Ausgabe des Bildes  
grestore  
showpage
```



Hinweis zum Array für Transformation der Pixelmap:

Wenn ein Bild wie in PostScript von links unten nach rechts oben beschreiben ist, lautet die Matrix:  $\begin{bmatrix} w & 0 & 0 & h & 0 & 0 \end{bmatrix}$ , wobei  $w$  die Breite und  $h$  die Höhe in Pixel ist.

Wenn (wie üblich) ein Bild von links oben nach rechts unten beschrieben wird, lautet die Matrix:  $\begin{bmatrix} w & 0 & 0 & -h & 0 & h \end{bmatrix}$ .

## 8.4.5 Die Füllbefehle „fill“ und „eofill“

Wenn sich Pfade überlappen, entscheiden die Richtung, in der ein Pfad gezeichnet ist, und die Art des Füllbefehls, welche Teile gefüllt werden:

- `fill` arbeitet nach der Regel „Non-Zero Winding“.
- `eofill` arbeitet nach der Regel „Even-Odd“.

Abbildung 8.4.5.1 zeigt, wie die Füllung ermittelt wird:

Man schickt waagerechte Teststrahlen von links auf die Abbildung; bevor der Strahl die Abbildung trifft, steht der Zähler  $Z$  auf 0; immer wenn der Strahl einen Pfad kreuzt, ändert sich der Zähler  $Z$ ; der aktuelle Zählerstand entscheidet über die Füllung einer Fläche.

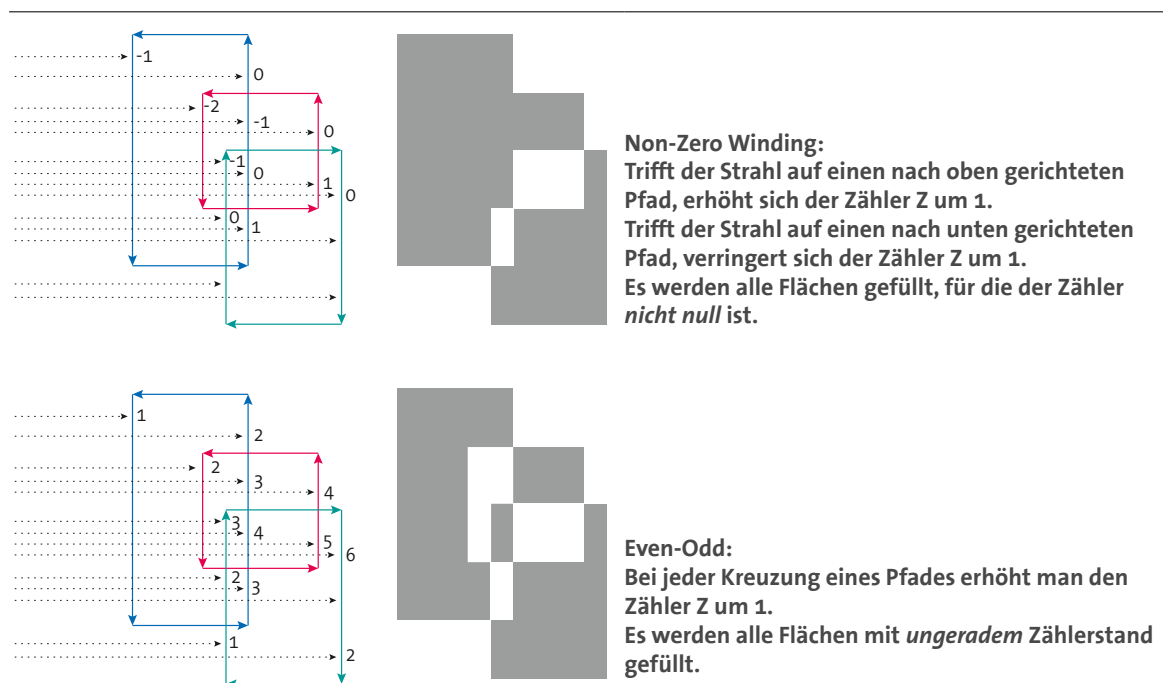


Abb. 8.4.5.1 | Die Füllungsregeln Non-Zero-Winding (`fill`) und Even-Odd (`eofill`).

</Ø>

## 8.5 EPS-Dateien

Wenn eine PostScript-Datei zur Platzierung in einem Layoutprogramm verwendet werden soll, muss sie bestimmte Regeln einhalten. Dieses Format wird als EPS (Encapsulated PostScript) bezeichnet (Dateiendung .eps).

- Die erste Zeile muss lauten: `%!PS-Adobe EPSF-3.0`
- Es muss eine BoundingBox angegeben sein: `%%BoundingBox` (→ Seite 94)
- Der Stack muss unverändert hinterlassen werden.
- Bestimmte Befehle, z. B. zur Festlegung von Rasterfunktionen, sind verboten.
- Auch der Befehl „showpage“ ist verboten.

Hinweis: Das Programm, das EPS-Dateien platziert, muss den Befehl unwirksam machen:  
`/showpage {} def`

(da zwischen den geschweiften Klammern nichts steht, ruft der Befehl `showpage` keine Aktion hervor).

- EPS-Dateien haben immer nur 1 Seite.
- Zeilen in EPS-Dateien sind maximal 255 Zeichen lang.
- Eine EPS-Datei kann ein Vorschaubild enthalten (für Layoutprogramme, die PostScript-Inhalte nicht am Bildschirm darstellen können).
- EPS wurde zeitweise als das Standardaustauschdatenformat für druckfähige Bilder verwendet. Heute ist EPS weitgehend von PDF abgelöst.
- Zu den Nachteilen von EPS-Dateien gehört, dass sie kein Farbmanagement erlauben.

## 8.6 PostScript-Schriften

PostScript-Schriften sind eine spezifische Umsetzung von Vektorgrafiken. Dabei werden die Buchstabenumrisse als Pfade (kubische Bézierkurven, d. h. Bézierkurven dritten Grades) definiert.

Es gibt verschiedene Arten (Typen) von PostScript-Schriften.

- ▶ Am weitesten verbreitet sind Typ-1-Schriften. Sie enthalten sog. Hints, die Rundungsfehler bei der Ausgabe in niedriger Auflösung verringern („hinted fonts“). Abbildung 8.6.1 zeigt den Unterschied:

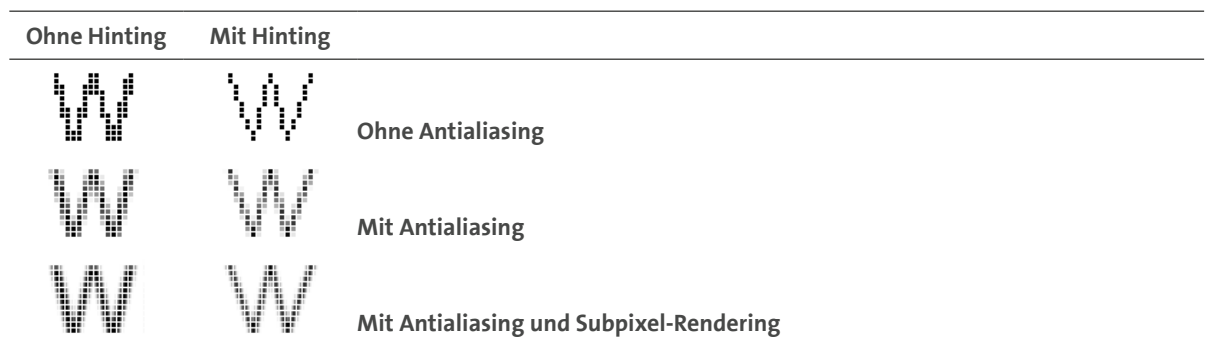


Abb. 8.6.1 | Einfluss von Hinting auf die Darstellung von Schriften.  
Quelle: <http://de.wikipedia.org/wiki/Hint>

- ▶ Typ-3-Schriften nutzen PostScript voll und können daher Angaben zu Farben, Schattierungen und Füllmustern, aber keine Hints enthalten.
- ▶ Typ 42: „PostScript-Adapter“ für TrueType-Schriften (Umrisskurven werden in TrueType als quadratische, in PostScript als kubische Bézierkurven beschrieben. Um TrueType auf einem PostScript-Drucker ausgeben zu können, müssen die Schriften in kubische Bézierkurven umgerechnet werden.)

Typ-1-PostScript-Schriften bestehen aus zwei Komponenten:

- ▶ der Auflistung der Buchstabenbreiten (Dickten = Laufweiten) und Unterschneidungspaare (z. B. „Vo“),
- ▶ der Sammlung der Kurvenformen für die einzelnen Buchstaben.

In *Windows* sind die Informationen einer PostScript-Schrift auf mindestens 2 Dateien verteilt:

- ▶ .pfb (PostScript Font Binary; oder als .pfa: ASCII-Zeichen statt Binary) für die Kurvenformen,
- ▶ .pfm (PostScript Font Metrics) für die Laufweitendaten

Optional:

- ▶ .afm (Adobe Font Metrics) für die Laufweiteninformationen in editierbaren ASCII-Zeichen
- ▶ .inf mit allgemeinen Angaben zur Schrift.



In *MacOS* ist es gebräuchlich, die Laufweiteninformationen für mehrere Schnitte einer Schriftfamilie in einer einzigen Datei (Typ „Font Suitcase“) zusammenzufassen. Die Umrissdefinitionen der einzelnen Schnitte bleiben auf einzelne Dateien verteilt.

Hinweis:

*OpenType-Schriften* (.otf) überwinden die Plattformgrenzen und fassen alle Informationen für einen Schnitt in einer Datei zusammen. Die Zeichenbeschreibung kann als TrueType oder PostScript erfolgen („truetype flavored“ bzw. „postscript flavored“).

Weitere Vorteile von Open Type (u. a.):

Unicode-Zeichensatzunterstützung, bis zu 65.536 Glyphen, Sortierung nach Layoutfeatures (Ligaturen, verschiedene Ziffernformen usw.)

## <!> Zusammenfassung „PostScript“

- Kenntnisse von PostScript sind eine wichtige Grundlage für das Verständnis von PDF. Wie SVG können PostScript-Dateien Text, Vektorgrafik und Pixelgrafik enthalten.

<Ø>

- PostScript verwendet die *Postfix-Notation*. PostScript arbeitet mit Stapeln, die nacheinander abgearbeitet werden. Transparenz von Objekten kann nicht definiert werden.
- Das Koordinatensystem für PostScript beginnt links unten.
- Text wird als *String* bezeichnet und steht zwischen runden Klammern. Der Befehl `show` gibt den Text aus.
- Wichtige Befehle zum zeichnen von Pfaden sind: `newpath`, `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto` und `closepath`. Der Buchstabe „r“ vor `moveto`, `lineto` oder `curveto` legt fest, dass die folgenden Koordinaten relativ zum aktuellen Punkt (`currentpoint`) angegeben sind. Das ist nötig, wenn ein Objekt für die spätere Verwendung beschrieben wird (`def`). Dargestellt wird ein Pfad durch den Befehl `stroke`. `showpage` gibt die fertige Seite aus.
- Als Bézierkurven werden von PostScript nur solche dritten Grades (kubische Bézierkurven, 2 Kontrollpunkte) verwendet. Vom Ausgangspunkt werden nach dem Befehl `curveto` (`rcurveto`) die x,y-Koordinaten der beiden Kontrollpunkte und des Endpunkts.
- Jeder Pfad kann als Beschneidungspfad für andere Objekte definiert werden.
- Das „Encapsulated PostScript Format“ (EPS) muss bestimmten Regeln folgen. Es wird heute in der Druckvorstufe kaum noch verwendet. PDF hat die Funktion von EPS übernommen.
- PostScript-Schriften nutzen kubische Bézierkurven zur Beschreibung der Buchstabenkonturen. Typ-1-Schriften enthalten „Hints“, die die Ausgabequalität bei niedriger Auflösung (einfache Drucker, Bildschirm) verbessern. PostScript-Schriften liegen für Windows und MacOS in verschiedenen Formaten vor. Das OpenType-Format überwindet die Plattformgrenzen und erlaubt große Zeichensätze.

</Ø>

---

## 9 Portable Document Format (PDF)

### Übersicht

- 9.1 Von PostScript zu PDF
  - 9.2 Erstellen von PDF-Dateien
  - 9.3 Struktur und Objekte einer PDF-Datei
  - 9.4 PDF-Versionen von 1.3 bis 1.7 (Ausblick: 2.0)
  - 9.5 PDF-Standards (A, E, X, UA, VT)
  - 9.6 Preflight
- 

### 9.1 Von PostScript zu PDF

PDF ist heute das Standardformat für den Datenaustausch in der Druckvorstufe und hat PostScript bzw. Encapsulated PostScript (EPS) weitestgehend abgelöst. Die Entwicklung nahm ihren Ausgang 1991 mit einer Veröffentlichung, die den Titel „Camelot“ hatte. Darin skizzierte John E. Warnock, Mitbegründer von Adobe, das Konzept einer plattformübergreifenden Darstellung von PostScript-Dateien, die auf dem Bildschirm leicht umsetzbar ist und zeigt, wie das Dokument gedruckt aussieht.

Das Format PDF 1.0 wurde 1993 vorgestellt. Die zugehörige Software Acrobat 1.0 bestand aus Reader (damals noch kostenpflichtig), Exchange und Distiller.

PDF war zunächst wenig erfolgreich. Der Standard wurde Schritt für Schritt erweitert. Interessant für die Druckvorstufe wurde PDF ab 1999 mit der Version 1.3. Die aktuelle Version ist PDF 1.7, das seit Juli 2008 ISO-Standard 32000-1 ist.

Die folgende Übersicht listet Unterschiede zwischen PostScript und PDF auf.

| PostScript                                                                                                | PDF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Die Interpretation von PostScript-Dateien ist rechenintensiv und zeitaufwendig.                           | PDF-Dateien sind kompakter und enthalten die Informationen als „Display List“.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| PostScript ist ein Streaming-Format. Daher ist der Zugriff auf bestimmte Seiten bzw. Objekte umständlich. | PDF erlaubt den schnellen Zugriff auf Seiten bzw. Objekte (Random Access).                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|                                                                                                           | Je nach Version zusätzlich möglich (u. a.): <ul style="list-style-type: none"><li>– Hypertext</li><li>– Metainformationen</li><li>– eingebettete Multimediaelemente (Audio, Video, SWF/Flash)</li><li>– eingebettete Job-Description (JDF)</li><li>– neben den PostScript-Farbräumen auch ICC-Farbräume (Farbmanagement)</li><li>– Kompression für Pixelbilder, Vektorgrafik und Text</li><li>– Tags, z. B. zur automatischen Verarbeitung der Textstruktur</li><li>– Interaktive Dokumente/Formulare mit JavaScript</li></ul> |

PDF und PostScript weisen einige Gemeinsamkeiten auf:

- ▶ Gleiches Grafikmodell, daher sind ähnliche Verarbeitungsfunktionen verfügbar.
- ▶ Halbtonerzeugung (Rasterung) folgt dem gleichen Prinzip.
- ▶ Für einige PDF-Funktionen wird PostScript-Code verwendet.

## Literatur zu PDF

Der ISO-Standard 32000 ist bei Adobe frei herunterladbar (Inhalt identisch mit dem offiziellem Dokument):

[http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000\\_2008.pdf](http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf)

Dazu gehören 2 Supplements:

[http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/adobe\\_supplement\\_iso32000.pdf](http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/adobe_supplement_iso32000.pdf)

[http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/adobe\\_supplement\\_iso32000\\_1.pdf](http://www.images.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/adobe_supplement_iso32000_1.pdf)

Das offizielle ISO-Dokument (198,00 CHF) kann bezogen werden bei:

[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=51502](http://www.iso.org/iso/catalogue_detail.htm?csnumber=51502)

Leonard Rosenthol. Developing with PDF. Erste Auflage, Oktober 2013. O'Reilly Media Inc.

(Hinweis: Das erste Kapitel des Buchs (Syntax von PDF-Dateien) ist als Sampler kostenlos erhältlich: [http://cdn.oreillystatic.com/oreilly/booksamplers/9781449327910\\_sampler.pdf](http://cdn.oreillystatic.com/oreilly/booksamplers/9781449327910_sampler.pdf)

[http://www.adobe.com/content/dam/Adobe/en/technology/pdfs/PDF\\_Day\\_A\\_Look\\_Inside.pdf](http://www.adobe.com/content/dam/Adobe/en/technology/pdfs/PDF_Day_A_Look_Inside.pdf)

## 9.2 Erstellen von PDF-Dateien

Es gibt mehrere Möglichkeiten.

- ▶ Zuerst wird eine PostScript-Datei erstellt: direkt mit einem Text-Editor oder aus einer Anwendung heraus über den PostScript-Druckertreiber.

Im zweiten Schritt wird diese PostScript-Datei mit Acrobat Distiller (Alternative: Ghostscript) in eine PDF-Datei umgewandelt.

In Distiller werden die Einstellungen zum Erzeugen von PDF-Dateien in Dateien mit der Extension .joboptions gespeichert. Sie fassen z. B. die Einstellungen für bestimmte PDF-Standards zusammen. Viele Druckereien stellen solche Dateien zu Verfügung, die auf den eigenen Workflow optimiert sind.

Hinweis:

PostScript unterstützt einige Features nicht, die in aktuellen PDF-Versionen möglich sind. Dazu gehören Transparenz, Ebenen, Hyperlinks und Tags. PDF-Dateien mit solchen Features können daher nicht über PostScript-Dateien erzeugt werden.

- ▶ PDF-Erstellung mithilfe von Software-Bibliotheken
  - ▷ Direkter Export aus einer Anwendung: CS-Programme (u. a. InDesign), QuarkXpress
  - ▷ Andere Software-Bibliotheken wie
    - <http://www.pdflib.com>
    - <http://itextpdf.com>
    - SDK (Software Developer Kit) für Acrobat
- ▶ Aus einer XML-Datei mit einem Prozessor für die XML-Formatierungssprache XSL-FO (Extensible Stylesheet Language-Formatting Objects), z. B.
  - RenderX (<http://www.renderx.com>),
  - FOP (Formatting Objects Processor, <http://xmlgraphics.apache.org/fop>)

## 9.3 Struktur und Objekte einer PDF-Datei

Eine typische PDF-Datei besteht aus 4 Abschnitten:

|                                           |                                                                                                                                                                                                                                                    |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Header</b>                             | <ul style="list-style-type: none"><li>▷ 1. Zeile: Information über die PDF-Version (z. B.: %PDF-1.7)</li><li>▷ 2. Zeile: Falls binäre Daten enthalten sind: %-Zeichen plus 4 Zeichen binärer Code (ASCII &gt; 128), z. B.: %aaïö</li></ul>         |
| <b>Body</b><br>(🔴 9.3.1)                  | <ul style="list-style-type: none"><li>▷ Objekte des Dokuments, d. h. der darzustellende Inhalt<br/>Beginnt mit dem ersten Objekt, z. B.:<br/>1 0 obj<br/>endobj</li><li>▷ Reihenfolge frei, Objekte über Cross-Reference Table zu finden</li></ul> |
| <b>Cross-Reference Table</b><br>(🔴 9.3.2) | <ul style="list-style-type: none"><li>▷ Einsprungsadressen für die indirekten Objekte (🔴 Seite 109) aus dem Body. Die Tabelle beginnt mit<br/>xref</li></ul>                                                                                       |
| <b>Trailer</b><br>(🔴 9.3.3)               | <ul style="list-style-type: none"><li>▷ U. a. die Adresse der Cross-Reference Table und die Einsprungsadresse zum Wurzelement (Root) der Objekte</li></ul>                                                                                         |

### 9.3.1 Bestandteile von Body

Es werden **8 Objekttypen** unterschieden:

#### ■ Boolean

Boolesche Werte: true, false

#### ■ Number

Der Adressraum für Zahlen ist 64 bit groß.

Die Vorzeichen + oder – sind möglich; Dezimalpunkt.

Im Gegensatz zu PostScript ist keine Scientific Notation erlaubt.

▷ Integer (ganze Zahlen)

▷ Real (reelle Zahlen)

#### ■ Name

Namen fangen immer mit / an, z. B.:

/Helvetica

/Blue#20Green

Hinweise:

▷ Groß-/Kleinschreibung wird unterschieden.

▷ Der Hexcode #20 steht für den Leerschritt (Dezimal: 32) und ist ein *white-space character*, Diese dienen zum Trennen von Namen und Zahlen.

▷ Kommentare beginnen mit % und werden bei der Ausgabe ignoriert.

#### ■ String

Ein String ist eine beliebig lange Folge von 8-Bit-Zeichen, die in runden Klammern (Parentheses) stehen, z. B.:

(Hallo Druckvorstufe)

(A+B)

%ist identisch mit (A\053B), wobei 053 der Octalcode für das Pluszeichen ist (dezimal: 43).

Hinweis:

Die Zeichen ( , ), \, [ und ] müssen so generiert werden:

\(, \), \\, \133 und \135

### ■ Array

Ein Array ist eine eindimensionale Gruppierung von Objekten zwischen eckigen Klammern (Brackets), z. B.:

```
/MediaBox [0 0 612 792]      %Das sind die Maße für die MediaBox,
                                Letterformat. Wie in PostScript gilt
                                die Maßeinheit pt = 1/72 Inch

[ (Text) -20.5 (End) 4]      %Ein Array aus String, Real, String,
                                Integer

[ [ 1 2 3 ] [ 4 5 6 ] ]      %Ein Array kann Arrays bestehen.
```

### ■ Dictionary

Ein Dictionary (Verzeichnis) ist eine assoziative Tabelle aus Paaren von: Zugriffsschlüssel (*Key*) und zugehörigem Wert (*Value*).

Der Schlüssel ist immer eine Name, der Wert kann jeder Objekttyp sein, z. B. eine Zahl, ein Name oder wiederum ein Verzeichnis.

Dictionary-Objekte sind die häufigsten Objekte in PDF-Dateien, weil sie die Basis fast aller übergeordneten Objekte sind.

Die Reihenfolge der Einträge spielt keine Rolle, die Zahl ist unbegrenzt.

Ein Dictionary steht zwischen doppelten spitzen Klammern (Angle Brackets). Beispiel:

```
<<
  /Type /Example              %Dem Schlüssel Type wird der Wert
                              /Example zugeordnet
  /Version 0.01               %Dem Schlüssel /Version wird der Wert
                              0.01 zugeordnet
>>
```

Hinweis: eine kompakte Schreibweise ist:

```
<</Type/Example/Version 0.01>>
```

Es gibt zwei Sonderformen für Verzeichnisse:

▷ *Name Tree*

▷ *Number Tree*

Beide haben dieselbe Funktion, aber die Keys sind entweder *String*-Objekte (*Name Tree*) oder *Number*-Objekte (*Number Tree*). Sie müssen sortiert sein (nach dem Unicode Collation Algorithm).

Der Wurzelknoten (Root Node) besteht aus einem Key: entweder *Names* oder *Numbers*, wenn der Baum einfach (= nicht weiter verzweigt) ist, oder *Kids*, wenn der Baum komplex (= weiter verzweigt) ist. Bei komplexen Bäumen heißt jeder Verzweigungsknoten *Kids*, am Ende jedes Astes stehen die *Names* bzw. *Numbers*.

**Beispiel** für einen **einfachen** (nicht verzweigten) *Name Tree*:

```
1 0 obj                      %Beginn der Definition eines indirekten
                                Objekts (🔴 Seite 109)
                                <<
      /Names [                %sortiert, daher A, N, Z
        (Apple) (Cox)
        (Name 1) 1            %Als Wert sind alle Objekttypen möglich.
        (Name 2) /Value2
        (Zebra) << /A /B >>
      ]
    >>
endobj                      %Ende der Beschreibung eines indirekten
                                Objekts
```

### Beispiel für einen **komplexen** (verzweigten) *Name Tree*:

```
1 0 obj
<<
  /Kids [
    2 0 R      %Aufruf eines indirekten Objekts
    3 0 R
  ]
>>
endobj
```

Wenn der Baum nur *eine* Verzweigung hat, haben die Objekte 2 0 und 3 0 den Wurzelknoten */Names*, wenn eine *weitere* Verzweigung besteht, den Wurzelknoten */Kids*.

#### ■ *Stream*

Ein Stream ist in einer PDF-Datei ein Folge von beliebig vielen 8-bit-Zeichen, die komprimiert oder codiert sein können. In Streams werden vorwiegend größere Datenmengen standardisierter Formate abgelegt, z. B. Bilddaten, Schriftdaten oder XML-Strukturen.

Ein Stream beginnt mit dem Stream-Dictionary, das die Eigenschaften des Streams auflistet. Ein Stream beginnt mit `stream` und endet mit `endstream`.

Beispiel:

```
<<
  /Type /XObject
  /Subtype /Image
  /Filter /FlateDecode      %Filter gibt an, ob bzw. wie die Daten
                             komprimiert sind
  /Length 496
  /Height 32
  /Width 32
>>
stream                      %Hier muss ein Zeilenende stehen!
.....                     %hier stehen die 496 Bytes des Streams,
                             gefolgt von einem Zeilenende
endstream
```

Hinweis:

Der Eintrag „Length“ muss immer vorhanden sein.

Kompressionsalgorithmen (u. a.):

FlateDecode = ZIP

DCTDecode = JPEG

JPXDecode = JPEG2000

#### ■ *Null*

Das Null-Objekt wird hier nur der Vollständigkeit halber erwähnt. Es hat keinen Wert und ist anonym. Theoretischer Hintergrund: Wenn auf ein indirektes Objekt (siehe unten) verwiesen wird, das nicht definiert ist, gilt das als Verweis auf das Null-Objekt.

## Unterschied zwischen direkten und indirekten Objekten

*Direktes Objekt* bedeutet, dass ein Objekt unmittelbar (direkt) erhalten wird, wenn die zugehörigen Zeilen der Datei gelesen werden.

*Indirekte Objekte* dagegen werden an einer beliebigen Stelle beschrieben und dort, wo sie benötigt werden, aufgerufen. Damit sie aufgerufen (referenziert) werden können, müssen sie eindeutig benannt werden. Dies geschieht in der ersten Zeile der Definition. Daher beginnt die Beschreibung aller indirekten Objekte mit:

`ON GN obj`

Dabei ist `ON` die (bezogen auf die vorliegende PDF-Datei) eindeutige Objektnummer (ID).

Die `ON` ist eine ganze nicht negative Zahl.



Die GN (ebenfalls eine ganze nicht negative Zahl) ist die Versionsnummer (generation number) des Objekts. Sie weicht nur dann von 0 ab, wenn ein inkrementelles Update der Datei gespeichert wurde (dabei bleiben die Originale von Body, Cross-Reference Table und Trailer erhalten und es werden Abschnitte mit den Änderungen angefügt).

- Beispiel für die Beschreibung und den Aufruf eines indirekten Objekts:

```
17 0 obj          %Beginn der Beschreibung von
                  Objekt 17, Version 0

.....
endobj           %Ende der Beschreibung des Objekts

17 0 R           %Aufruf des Objekts (R für Reference)
```

Beispiel für ein *String*-Objekt, einmal als direktes, einmal als indirektes Objekt beschrieben:

- ▷ Direktes Objekt:

```
(Hallo Druckvorstufe)
```

- ▷ Indirektes Objekt:

```
12 0 obj
(Hallo Druckvorstufe)
endobj
```

Dieses String-Objekt kann an jeder Stelle mit 12 0 R aufgerufen werden.

- Bedeutung bzw. Vorteil indirekter Objekte:  
Es müssen nur die Objekte gelesen werden, die für die Darstellung der aktuellen Seite benötigt werden.

### 9.3.2 Cross-Reference Table (Querverweistabelle)

Um ein indirektes Objekt zu lesen, muss der Reader an die Stelle springen, wo es beschrieben wird. Um dies zu beschleunigen, sind die Adressen aller indirekten Objekte in der Cross-Reference Table aufgelistet. Diese Tabelle steht an vorletzter Stelle des Dokuments vor dem Trailer. Das hat u. a. den Vorteil, dass ein PDF in einem Durchgang erstellt werden kann. Nachteil ist, dass erst das ganze Dokument geladen sein muss, bevor mit der Darstellung der ersten Seite begonnen werden kann (🔴 Linearization, Seite 115).

Beispiel:

```
xref
0 7              %Nummer des ersten Objekts, Zahl der Objekte
0000000000 65535 f  %Ab hier die 7 Objekte (Nummer 0 bis 6)
0000000060 00000 n
0000000228 00000 n
0000000424 00000 n
0000000145 00000 n
0000000333 00000 n
0000000009 00000 n
```

- Erklärung zur Cross-Reference Table:

- ▷ Der erste Eintrag in der Tabelle hat immer die Form  
0000000000 65535 f  
denn ein Objekt mit der Nummer 0 gibt es nicht. Das f am Zeilenende zeigt an, dass das Element nicht existiert und daher (theoretisch) frei (f) ist. Obige Tabelle beschreibt somit 6 aufrufbare Objekte.
- ▷ Die erste Ziffernfolge ist der Byte-Offset des Objekts im Dokument:  
Das Objekt 2 (dritte Zeile) beginnt nach Byte 228.
- ▷ Die zweite Ziffernfolge (hier durchgehend 0) gibt die Version (Generation Number) an.
- ▷ Das n am Ende sagt, dass die Objektnummer besetzt, d. h. nicht (n) frei ist.

### 9.3.3 Trailer, Root Object

Der Trailer ist ein *Dictionary*, das mit der Zeile `trailer` direkt nach der Cross-Reference Table beginnt. Zwei Einträge sind Pflicht:

- ▶ die Zahl der Objekte in der Cross-Reference Table und
- ▶ die ID des Root-Objekts, d. h. des Verzeichnisses (Dictionary) mit dem Namen `catalog`. Dieser Katalog enthält Referenzen auf die Objekte, die den Inhalt des Dokuments ausmachen. Wichtig: Das Root Object muss als indirektes Objekt beschrieben sein, damit es im Trailer referenzierbar ist. Abb. 9.3.3.1 zeigt die hierarchische Baumstruktur einer PDF-Datei, die in dem Objekt „catalog“ festgelegt wird.

Weitere Einträge sind möglich (z. B. zu Verschlüsselung, IDs, Infos usw.)

Beispiel:

```
trailer
<<
  /Size 23          % 23 Einträge in der Cross-Reference Table
  /Root  5 0 R      % Das Dictionary „Catalog“ hat die Referenz  5 0 R
>>
```

<Ø>

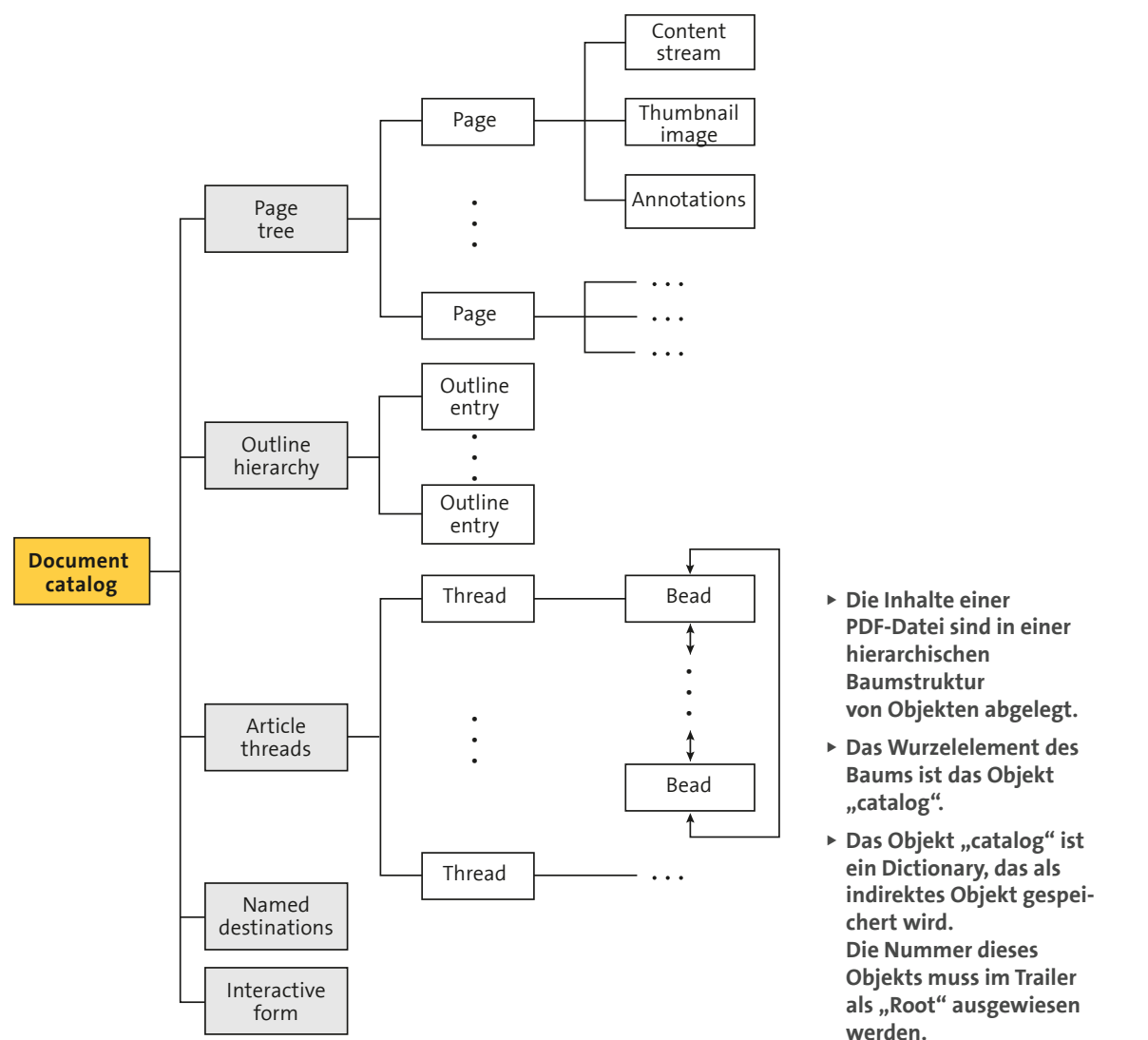


Abb. 9.3.3.1 | Hierarchische Baumstruktur einer PDF-Datei, die im Katalog-Objekt festgelegt ist.

Hellgrau hinterlegt: erste hierarchische Ebene der Dokumentstruktur.

Quelle:

[http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000\\_2008.pdf](http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf)

Die Elemente der ersten hierarchischen Ebene (hellgrau hinterlegt) in der Dokumentstruktur aus Abb. 9.3.3.1 haben folgende Funktionen:

- ▶ *Page tree* (ISO 32000-1:2008 , 7.7.3)
  - ▷ Die Seiten eines Dokuments sind unter einem Knoten zusammengefasst. Jede Seite ist als indirektes Objekt definiert. Die Struktur des Page-tree-Knotens, also die Anordnung der Seiten, ist unabhängig von der Gliederungsstruktur des Dokuments.
- ▶ *Outline hierarchy* (12.3.3)
  - ▷ Logische Gliederung des Dokumentes (Kapitel, Unterkapitel usw.)
  - ▷ Für die Bestandteile der Gliederung gibt es Lesezeichen (Bookmarks), die in einer Baumstruktur angeordnet sind.
- ▶ *Article threads* (12.4.3)
  - ▷ Navigationsunterstützung für zusammenhängende Inhaltsteile, die aber nicht physikalisch aufeinander folgen und auch nicht der Gliederungsstruktur folgen.  
Z. B. ein Zeitungsartikel, der auf der Titelseite startet und auf Seite xy fortgesetzt wird.
- ▶ *Named destinations* (12.3.2)
  - ▷ Unterstützung für spezifische Sichten auf ein PDF-Dokument, z. B. Inhaltsverzeichnis mit Seitenvorschaubildern. Gespeichert werden Zieladressen von Objekten, die direkt angewählt und mit bestimmten Vorgaben angezeigt werden sollen.
- ▶ *Interactive form*
  - ▷ Dieser Knoten enthält Eingabefelder und Formulare.

</Ø>

## 9.3.4 Beispiel einer einfachen PDF-Datei

Die nachfolgende PDF-Datei (Abb. 9.3.4.1, nächste Seite) ist manuell erstellt. Kein Programm würde sie so schreiben.

Das Beispiel zeigt, dass die einzelnen Objekte in beliebiger Reihenfolge stehen können. Im Beispiel steht der `Catalog` (das Root-Objekt, auf das der Trailer verweist, hier mit ① markiert) ganz oben im Body. In den erklärenden Kommentaren in der rechten Spalte ist mit den roten Zahlen im roten Kreis markiert, wo die Folgeobjekte zu finden sind.

|                       |                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Header                | %PDF-1.4                                                                                                                                                              | %Header (im Fall einer Binärdatei als zweite Zeile:<br>%ääïÖ (vier Zeichen Binärcode, ASCII > 128)                                                                                                                                                                                                                                                                                                                                                                                              |
|                       | ① 6 0 obj<br><<<br>/Type /Catalog<br>/Pages 5 0 R<br>>><br>endobj                                                                                                     | %Beginn von Body.<br>%Hier wird als indirektes Objekt das Katalog-Objekt<br>(als „Root“ im Trailer ausgewiesen) beschrieben. Es<br>verweist auf den Knoten „Pages“ mit der Referenz<br>5 0 R → ②                                                                                                                                                                                                                                                                                                |
|                       | ③ 1 0 obj<br><<<br>/Type /Page<br>/Parent 5 0 R<br>/MediaBox [ 0 0 612 792 ]<br>/Resources 3 0 R<br>/Contents 2 0 R<br>>><br>endobj                                   | %1 0 obj ist die (einzige) Seite dieses Dokuments.<br>Angabe ist die MediaBox (hier: Letterformat),<br>als Resources das Objekt 3 0 R (ein Dictionary)<br>→ ④a, als Contents das Objekt 2 0 R (ein Stream)<br>→ ④b                                                                                                                                                                                                                                                                              |
|                       | ⑤ 4 0 obj<br><<<br>/Type /Font<br>/Subtype /Type1<br>/Name /F1<br>/BaseFont /Helvetica<br>>><br>endobj                                                                | %4 0 obj legt fest, dass die Schrift F1 (Helvetica)<br>verwendet wird.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Body                  | ④b 2 0 obj<br><<<br>/Length 53<br>>><br>stream<br>BT<br>/F1 24 Tf<br>1 0 0 1 260 600 Tm<br>(Hallo Druckvorstufe) Tj<br>ET<br>endstream<br>endobj                      | %Dieses Objekt ist der Content. Er ist ein Stream.<br>Er Beginn mit dem Dictionary, das zwingend die<br>Länge ausweist. BT beginnt, ET endet den Text. In<br>der nächsten Zeile werden die Schrift F1 und die<br>Größe 24 pt aufgerufen, Tf ist der Operator dafür<br>(in Postfix Notation). Es folgen die Koordinaten für<br>den Text (Textmatrix) mit dem Operator Tm. Dann<br>kommt der eigentliche Text als String zwischen<br>runden Klammern, der mit dem Operator Tj ausgegeben<br>wird. |
|                       | ② 5 0 obj<br><<<br>/Type /Pages<br>/Kids [ 1 0 R ]<br>/Count 1<br>>><br>endobj                                                                                        | %5 0 obj ist ein Verzweigungsknoten, erkennbar am<br>Wurzelknoten mit dem Namen /Kids. Es gibt nur ein<br>Kid, nämlich 1 0 R (die einzige Seite des<br>Dokuments) → ③                                                                                                                                                                                                                                                                                                                           |
|                       | ④a 3 0 obj<br><<<br>/ProcSet[/PDF/Text]<br>/Font <</F1 4 0 R >><br>>><br>endobj                                                                                       | %3 0 obj ist ein Dictionary, das ein Procedure Set<br>definiert; als Schrift (das ist der Key des<br>Dictionary) wird auf ein weiteres Dictionary<br>(doppelte spitze Klammern) verwiesen, das indirekte<br>Objekt 4 0 R → ⑤                                                                                                                                                                                                                                                                    |
| Cross-Reference Table | xref<br>0 7<br>0000000000 65535 f<br>0000000058 00000 n<br>0000000244 00000 n<br>0000000411 00000 n<br>0000000164 00000 n<br>0000000352 00000 n<br>0000000009 00000 n | %Beginn der Cross-Reference Table<br>%0 ist die Nummer des ersten Objekts, insgesamt 7<br>Objekte. Da es das Objekt "0" nicht gibt, steht am<br>Ende der Zeile ein "f".                                                                                                                                                                                                                                                                                                                         |
| Trailer               | trailer<br><<<br>/Size 7<br>/Root 6 0 R<br>>><br>startxref<br>472<br>%%EOF                                                                                            | %Beginn des Trailers<br>%Dictionary mit Angabe der Zahl der Einträge in der<br>der Cross-Reference Table und der ID des<br>Root-Objekts → ①<br><br>%Startposition der Cross-Reference-Table                                                                                                                                                                                                                                                                                                     |

Abb. 9.3.4.1 | Beispiel einer einfachen, manuell erstellten PDF-Datei.

### 9.3.5 Das Koordinatensystem von PDF

Wie bei PostScript handelt es sich um ein klassisches kartesisches Koordinatensystem mit dem Ursprung links unten. Die Maßeinheit ist Point (pt), mit  $1 \text{ pt} = 1/72 \text{ Inch}$ . Andere Einheiten können definiert werden, z. B.:

/UserUnit 2.0

%36 UserUnits (statt 72 pt) sind 1 Inch.

Ebenfalls wie bei PostScript unterscheidet man den

- ▶ User Space (unabhängig von der Auflösung des Ausgabegeräts) und den
- ▶ Device Space (Abb. 9.3.5.1).

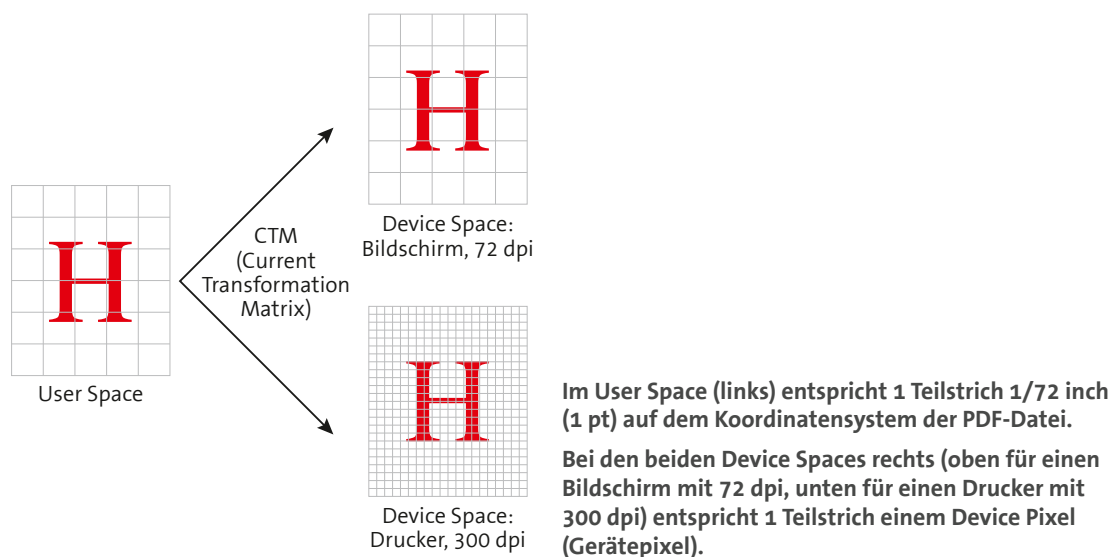


Abb. 9.3.5.1 | User Space und verschiedene Device Spaces in PDF.

### 9.3.6 Box-Angaben in PDF

Die MediaBox muss immer angegeben werden. Sie ist das Format des Ausgabemediums (Abb. 9.3.6.1).

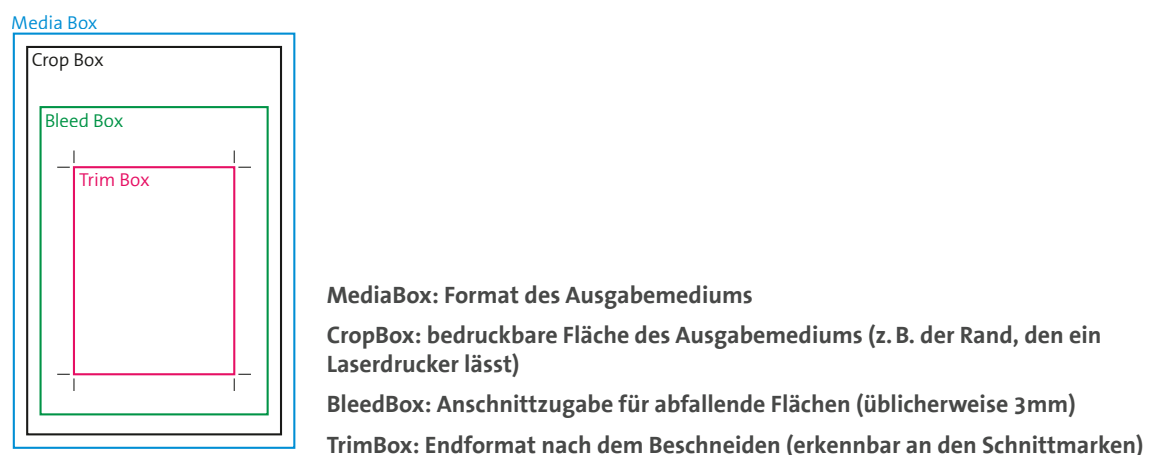


Abb. 9.3.6.1 | Box-Angaben in einer PDF-Datei.

### 9.3.7 Linearization von PDF-Dateien

Da nach dem Header der Trailer und dann die Cross-Reference Table gelesen werden, können PDF-Dateien erst dargestellt werden, wenn die gesamte Datei vorliegt.

Um beim Herunterladen aus dem Web schon vor dem kompletten Laden einer PDF-Datei mit der Darstellung beginnen zu können, werden PDFs linearisiert („für schnelle Web-Ansicht optimieren“, „Fast Web View“):

- ▶ Dabei werden die Objekte seitenweise gruppiert und sortiert.
- ▶ Ein *Linearization parameter dictionary* folgt unmittelbar auf den Header.
- ▶ Eine Teiltabelle mit den Cross-References und ein Teiltrailer werden am Anfang der Datei eingefügt, damit auf alle Objekte, die das Root-Objekt benötigt, sowie die Objekte der ersten Seite zugegriffen werden kann.

## 9.4 PDF-Versionen von 1.3 bis 1.7 (Ausblick: 2.0)

| Jahr     | PDF-Version | Acrobat-Version | Für Druckvorstufe relevante Merkmale                                                                                                                                                                                                       |
|----------|-------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1999     | 1.3         | ab 3.0          | Geräteunabhängige Farben werden unterstützt, bis zu 8 Farbgeber; keine Transparenz (Transparenzreduktion erforderlich), keine Ebenen.                                                                                                      |
| 2001     | 1.4         | ab 5.0          | + Transparenz, Tags; PDF-Erstellung über PostScript-Datei/ Distiller unterstützt Transparenz nicht.                                                                                                                                        |
| 2003     | 1.5         | ab 6.0          | + Einbetten von Bildern als JPEG2000, von Filmen als MPEG und von Audiodaten als MP3; verbesserte Kompression durch „object streams“, Ebenen (Layer), bis zu 31 Farbgeber.                                                                 |
| 2004     | 1.6         | ab 7.0          | + Unterstützung von OpenType.                                                                                                                                                                                                              |
| 2006     | 1.7         | ab 8.0          |                                                                                                                                                                                                                                            |
| 2008     |             |                 | ISO 32000-1 wird veröffentlicht. Beruht auf 1.7. PDF ist nicht mehr Eigentum der Firma Adobe.                                                                                                                                              |
| 2017 (?) | 2.0         |                 | Erwartet werden Erweiterungen des Farbmanagements (z. B. seitenweise Zuordnung von Farbprofilen), Optimierung von Transparenzen, Verbesserungen im Workflow usw. (Infos unter <a href="http://www.pdflib.com/de/">www.pdflib.com/de/</a> ) |

## 9.5 PDF-Standards (A, E, X, UA, VT)

Für verschiedene Zwecke wurden Standards für PDF-Dateien definiert, die schon bei der Erstellung berücksichtigt werden müssen.

- ▶ PDF/A  
Standard für Langzeitarchivierung („Archive“) von Dateien, ISO-Standard  
Aktuell: PDF/A-3 (von 2012, basiert auf PDF 1.7)
- ▶ PDF/E  
Standard für „Engineering“, u. a. mit der Fähigkeit einer interaktiven 3-D-Darstellung.
- ▶ PDF/X:  
Ein ganzes Set von Standards für die Druckvorstufe (Details weiter unten)
- ▶ PDF/UA  
Standard für barrierefreie PDFs („Universal Access“).  
Seit 2012 Bestandteil der ISO. Basiert auf PDF 1.7.
- ▶ PDF/VT  
Standard für variablen Datendruck und Transaktionsdruck,  
z. B. personalisierte Newsletter, Mailings usw.



- ▶ PDF/H  
(Noch nicht offizieller) Standard für den Austausch von Gesundheitsdaten („Healthcare“), z. B. Röntgenbildern und anderen Befunden. Verschlüsselung zum Schutz von Patientendaten.

## PDF/X-Standards für den Druck

Das „X“ steht für „blind eXchange“, d. h. verlässlichen Austausch.

Hinweis:

Wenn eine PDF-Datei einen der folgenden Standards erfüllt, heißt das nicht, dass sie fehlerfrei ist (z. B. ausreichende Auflösung von Bildern). Ersetzt nie ein Preflight (siehe unten)!

- ▶ PDF/X-1a:2001  
Dieser Standard wird von nahezu allen Druckereien beherrscht.
  - ▷ Neben der MediaBox muss bei Bedarf auch eine BleedBox definiert sein.
  - ▷ Es muss ein Flag gesetzt sein, ob Überfüllung (Trapping) bereits erfolgte (true/false).
  - ▷ Verwendete Schriften bzw. Glyphen sind eingebettet.
  - ▷ Alle Farben müssen in CMYK bzw. Spotfarben vorliegen (kein RGB, kein Lab).
  - ▷ Ausgabe-Intent muss definiert sein.
  - ▷ Es gibt nur ein globales ICC-Profil, d. h. keine ICC-Profile für einzelne Elemente.
  - ▷ Keine Transparenz erlaubt.
  - ▷ PDF 1.3
  - ▷ Keine Verschlüsselung.
  - ▷ Keine Acrobat-Kommentare.
  - ▷ Keine Multimedialinhalte.
  - ▷ Keine Transferkurven.
- ▶ PDF/X-1a:2003  
Wie PDF/X-1a:2001, aber aktualisiert auf der Basis von PDF 1.4
- ▶ PDF/X-3:2002  
Enthält Erweiterung von X-1a auf der Basis von PDF 1.3
  - ▷ Erlaubt neben CMYK und Spotfarben auch RGB- und Lab-Daten.
  - ▷ Individuelle Farbprofile können eingebunden werden.
- ▶ PDF/X-3:2003  
Wie PDF/X-3:2002, aber aktualisiert auf der Basis von PDF 1.4
- ▶ PDF/X-4  
Basiert auf PDF 1.6
  - ▷ Unterstützt CMYK, RGB, Graustufen und Sonderfarben.
  - ▷ Transparenz und Ebenen sind erlaubt.
  - ▷ 16-bit-Bilddaten, JPEG2000.
  - ▷ OpenType-Fonts.
- ▶ PDF/X-5  
Wie X-4, zusätzlich extern referenzierte Elemente und Mehrkanal-ICC-Profile.

## 9.6 Preflight

Ein Preflight hat das Ziel, Probleme vor der Ausgabe auf Papier, Film oder Platte aufzudecken. Zu den gängigen Problemfeldern zählen:

- ▶ Überdrucken von schwarzer Schrift
- ▶ Dünne Linien
- ▶ Farbaufbau, z. B. von schwarzer Schrift auf weißem Hintergrund.
- ▶ Auflösung von Bildern

Software für ein Preflight:

- ▶ PitStop  
PitStop Pro ist ein Plug-in für Adobe Acrobat. Es erlaubt Preflight, das Bearbeiten und die automatische Korrektur von PDF-Dateien.
- ▶ Auch mit Acrobat Pro ist ein Preflight möglich. Verschiedene Sets von Ausgabebedingungen sind vordefiniert, z. B. „Bogen-Offset“. Fehlerkorrektur nur in Einzelfällen möglich.

## <!> Zusammenfassung „PDF“

- PDF ist heute das Standardformat für den Datenaustausch in der Druckvorstufe.
- Eine PDF-Datei besteht aus 4 Abschnitten:
  - ▷ Header: Info über PDF-Version
  - ▷ Body: sämtliche Objekte
  - ▷ Cross-Reference Table (Querverweistabelle): Tabelle mit den Adressen der indirekten Objekte
  - ▷ Trailer: mit Angabe der Zahl der Einträge in der Cross-Reference Table und deren Startadresse sowie ID des Objekts, das den Catalog enthält (Root-Object).
- PDF kennt 8 Objekttypen:
  - ▷ *Boolean*: Boolesche Werte (true, false)
  - ▷ *Number*: reelle Zahlen, keine Scientific Notation
  - ▷ *Name*: Namen fangen immer mit / an, z. B.: /Helvetica
  - ▷ *String*: Das ist darzustellender Text, eine beliebig lange Folge von 8-Bit-Zeichen zwischen runden Klammern.
  - ▷ *Array*: eindimensionale Gruppierung von Objekten zwischen eckigen Klammern
  - ▷ *Dictionary (Verzeichnis)*: assoziative Tabelle aus Paaren von Zugriffsschlüssel (Key) und zugehörigem Wert (Value) <</Version 0.01>>
  - ▷ *Stream*: beliebige Folge von beliebig vielen 8-bit-Zeichen, die komprimiert oder codiert sein können, z. B. Bilddaten, Schriftdaten oder XML-Strukturen
  - ▷ *Null*: nur theoretische Bedeutung
- Direkte und indirekte Objekte
  - ▷ *Direkt Objekte* werden direkt erhalten, wenn ihre Zeilen in der Datei gelesen werden.
  - ▷ *Indirekte Objekte* werden an einer beliebigen Stelle beschrieben und aufgerufen (referenziert), wenn sie benötigt werden.
    - Benennung: ON GN obj  
ON = eindeutige Objektnummer (ID), GN = Versionsnummer (generation number)
    - Aufruf: ON GN R
    - Die Adressen aller indirekten Objekte sind in der Cross-Reference Table aufgelistet.
    - Das Objekt mit dem Catalog (Root Object) muss als indirektes Objekt beschrieben sein.
- PDF-Dateien können aus PostScript-Dateien mithilfe von Programmen wie Distiller oder Ghostscript erstellt werden. Da PostScript keine Transparenz unterstützt, enthalten die so erzeugten PDFs keine Transparenz. Transparenz wird vor der Erstellung der PostScript-Datei verrechnet (Transparenzreduktion).
- Über verschiedene Software-Bibliotheken können PDF-Dateien z. B. aus Layoutprogrammen direkt (d. h. ohne Umweg über PostScript-Dateien) erstellt werden. Diese PDFs unterstützen alle Features von PDF, die in PostScript noch nicht implementiert sind, u. a. Transparenz.

## 10 Rastertechnik

### Übersicht

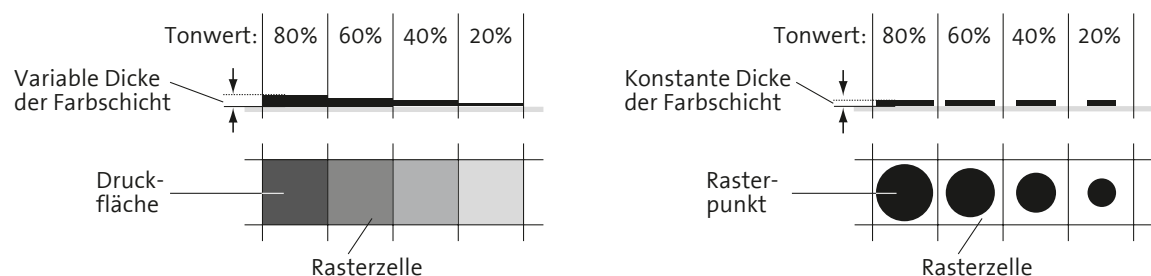
- 10.1 Warum muss gerastert werden?
  - 10.2 Analoge Raster
  - 10.3 Vom digitalen Bild zum digitalen Raster
  - 10.4 Amplitudenmodulierte (AM) Raster
  - 10.5 Frequenzmodulierte (FM) Raster
  - 10.6 Hybridraster (XM)
- 

### 10.1 Warum muss gerastert werden?

Druckverfahren lassen sich u. a. nach ihrer Fähigkeit einteilen, Halbtöne durch verschieden dicken Farbauftrag wiederzugeben. Man spricht in diesem Fall von „echten“ Halbtönen. Beispiel für ein solches Druckverfahren ist der Thermosublimationsdruck.

Im Gegensatz dazu kann der Offset-Druck die Dicke des Farbauftrags nicht variieren: Entweder wird an einer Stelle gedruckt (= Farbauftrag) oder nicht gedruckt (= kein Farbauftrag). Man spricht hier auch von binärem Verhalten („Farbe oder Nicht-Farbe“).

Immer dann, wenn ein Druckverfahren binär arbeitet oder die erreichbare Abstufung des Farbauftrags nicht fein genug ist, muss zur Darstellung von Halbtönen (zusätzlich) gerastert werden. Der jeweilige Halbton entsteht im Auge des Betrachters (oder im Sensor eines Messgeräts), indem sich das Papierweiß außerhalb der Rasterpunkte mit der Farbe der Rasterpunkte mischt. Abbildung 10.1.1 zeigt prinzipiell den Druck mit variablem und konstantem Farbauftrag.



**Abb. 10.1.1 | Druck mit variabler (links) und konstanter (rechts) Dicke der Farbschicht.**  
Bei konstanter Dicke der Farbschicht lassen sich Halbtöne nur durch Variieren der Größe der Rasterpunkte erzeugen.  
Quelle: Nach Kipphan

Hinweis:

Der oder das Raster? Gemäß Duden ist *der* Raster die Glasplatte, die zum Erzeugen analoger Raster verwendet wird (siehe unten), *das* Raster die Fläche eines Bildschirms, die sich aus Lichtpunkten zusammensetzt. In diesem Skript wird daher das Ergebnis des Rasterns als *das* Raster bezeichnet.

## 10.2 Analoge Raster

- Das Erzeugen von Halbtönen durch Raster hat eine lange Tradition. Abbildung 10.2.1 zeigt zwei Beispiele aus der Kunstgeschichte.



Linienraster auf Albrecht Dürers Kupferstich „Ritter, Tod und Teufel“ (1513; [Seite 69](#))



Der Pointillismus (19./frühes 20. Jahrhundert) erzeugt Farbwirkungen mit Tüpfeltechniken, die wie Vorläufer moderner Druckraster wirken. Paul Signac, „La calanque“ (1906)

Abb. 10.2.1 | Beispiele für das Erzeugen von Halbtönen durch Raster aus der Kunstgeschichte.

Die technische Erzeugung von analogen Rastern kennt Distanzraster (Reprokamera) und Kontaktraster (Film auf Film).

- Distanzraster: Rastern mit der Reprokamera

Abbildung 10.2.2 zeigt schematisch den Aufbau einer Reprokamera.

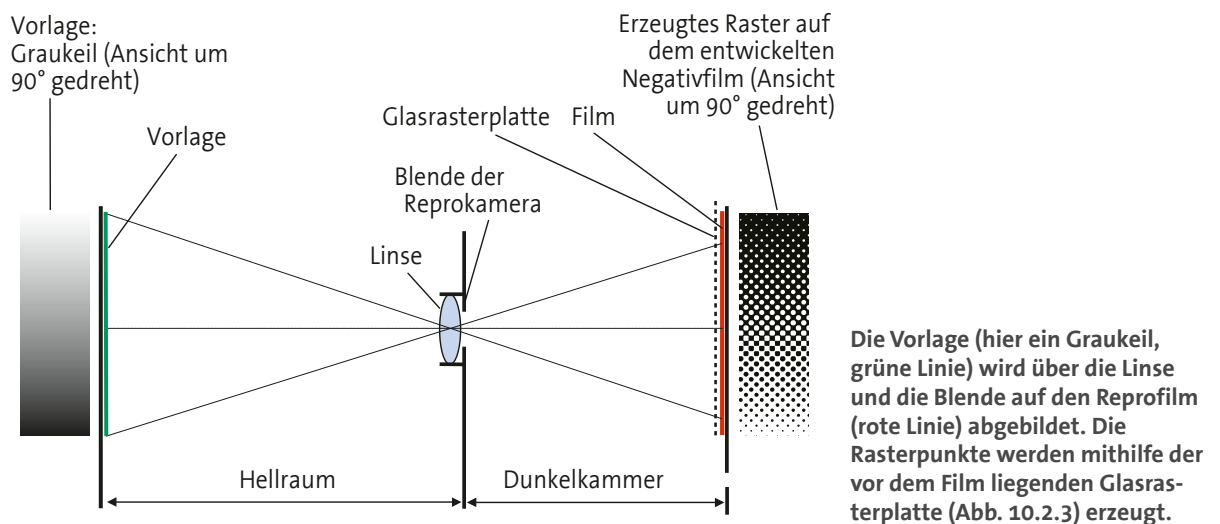


Abb. 10.2.2 | Anordnung und Aufbau einer Reprokamera zur Rasterung einer Vorlage.  
Quelle: Nach Kipphan

Abbildungen 10.2.3 zeigt die Funktion der Glasrasterplatte. Diese erzeugt Kern- und Halbschatten, sodass die Beleuchtungsstärke, die auf den Film trifft, in spitzen Maxima und Minima verläuft, deren Abstände dem Lochabstand auf der Platte (= Rasterweite) entsprechen. Schwellenwert und Gradation des Filmmaterials sind so ausgelegt, dass der Film unterhalb des Schwellenwerts überhaupt nicht geschwärzt wird, darüber maximal. Der Abstand der Rasterpunkte wird durch die Glasrasterplatte vorgegeben, der Durchmesser hängt von der Helligkeit der Vorlage ab.

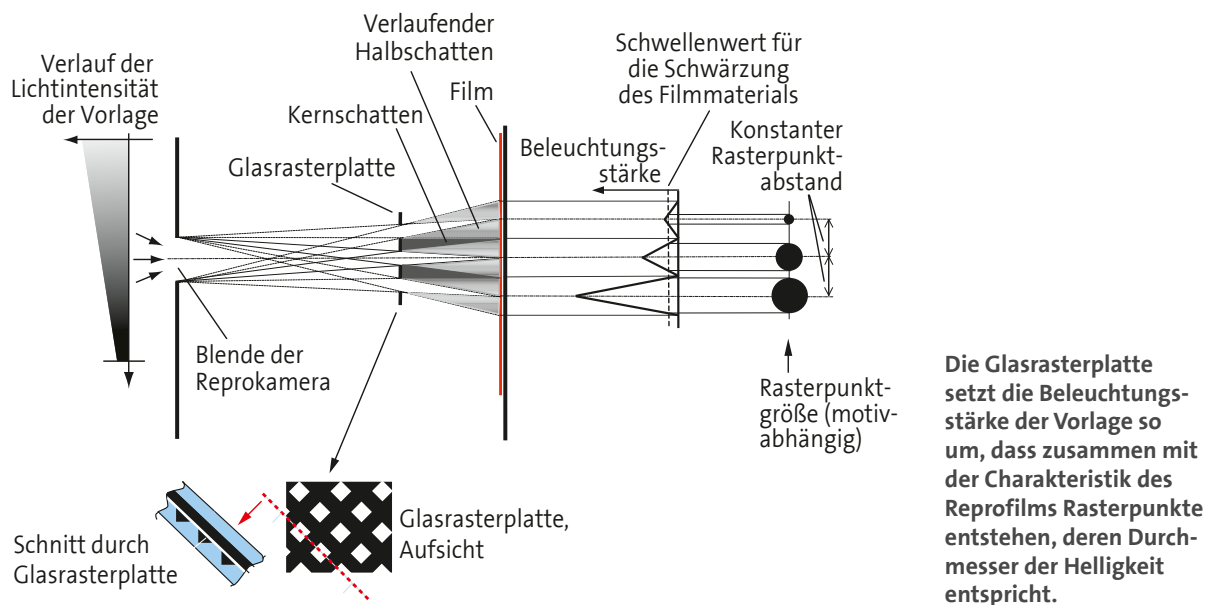


Abb. 10.2.3 | Wirkung der Glasrasterplatte in einer Reprokamera.  
Quelle: Nach Kipphan

#### ■ Kontaktraster

Ein Kontaktraster wird erzeugt, indem zwischen Vorlage und Reprofilm ein Kontaktrasterfilm gelegt wird. Auch hier führt das besondere Schwärzungsverhalten (Schwellenwert, Gradation) des Reprofilms dazu, dass der Film periodisch (= Rasterweite) geschwärzt wird und die Ausdehnung der Schwärzung von der Helligkeit der Vorlage bestimmt wird (Abb. 10.2.4).

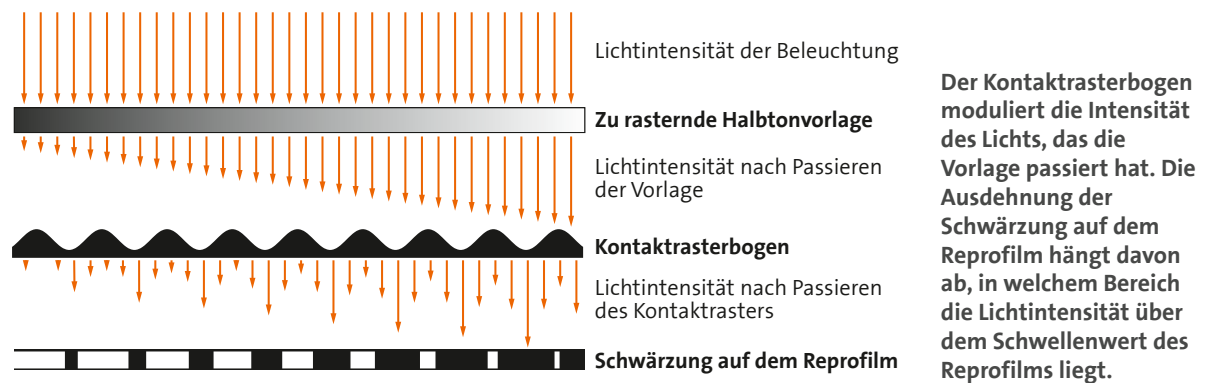


Abb. 10.2.4 | Rasterung mithilfe eines Kontaktrasterbogens, der direkt zwischen Vorlage und Reprofilm liegt.

## 10.3 Vom digitalen Bild zum digitalen Raster

Ausgangspunkt ist in dem folgenden Beispiel eine digitale Druckvorlage in Form einer PDF-Datei, in der ein digitales Foto in einem RGB-Farbraum eingebunden ist.

Es folgen zwei Schritte:

- ▶ RIP (Raster Image Processor, übernimmt Farbseparation und Rasterung),
- ▶ Filmbelichtung bzw. Druckplattenherstellung oder Ansteuerung einer digitalen Druckeinheit.

Die Farbseparation ordnet jedem Pixel des RGB-Bildes die Werte für die 4 Druckfarben Cyan, Magenta, Gelb und Schwarz (C, M, Y, K) zu. (Farbseparation ist nicht Thema dieses Skripts). Schematisch zeigt das Abbildung 10.3.1.



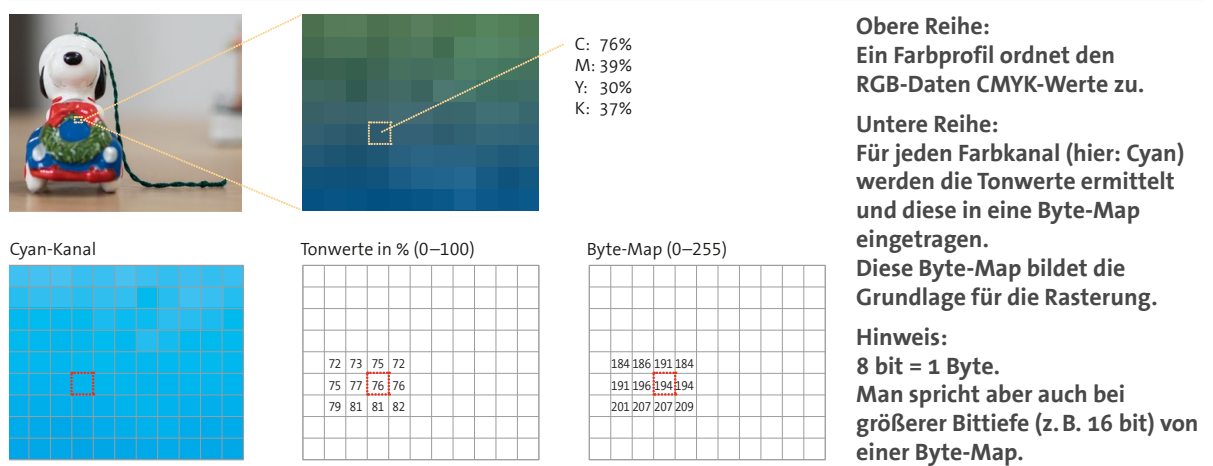


Abb. 10.3.1 | Farbseparation für ein RGB-Bild.

Abbildung 10.3.2 zeigt 2 moderne Druckraster in Ausschnittvergrößerungen: Raster mit Amplitudenmodulation (AM) und ein frequenzmoduliertes (FM) Raster.

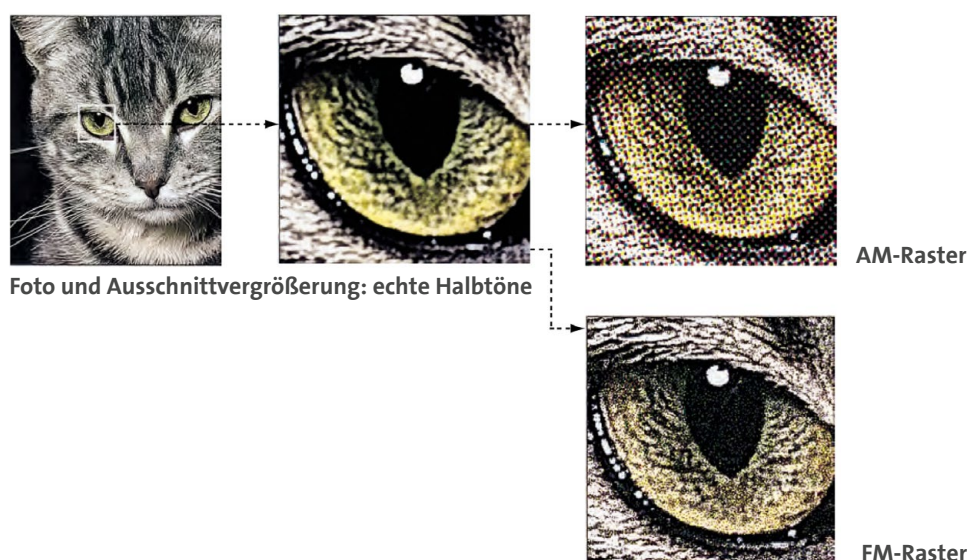


Abb. 10.3.2 | Beispiele für ein amplitudenmoduliertes (AM) und ein frequenzmoduliertes (FM) Raster.  
Quelle: Nach Kipphan

Beim AM-Raster haben die Rasterpunkte einen konstanten Abstand, aber ihre Größe variiert. Abbildung 10.3.3 zeigt die Umsetzung eines Grauverlaufs mithilfe verschiedener Rasterpunktformen sowie zum Vergleich ein FM-Raster (hier: Diffusion).

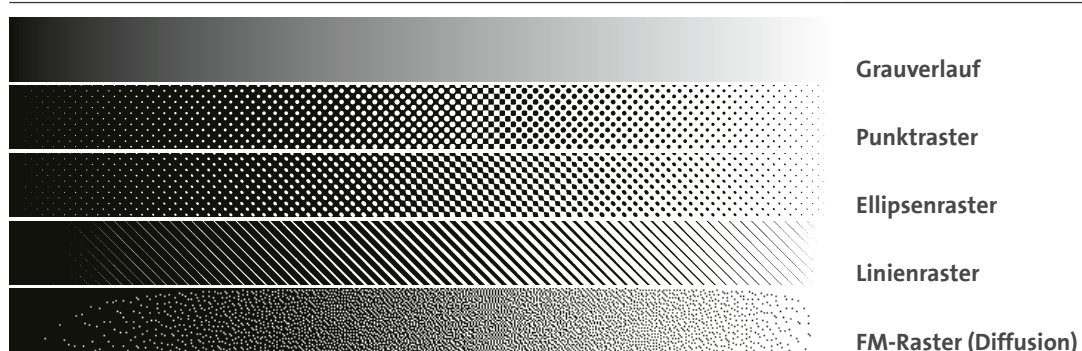


Abb. 10.3.3 | Umsetzung eines Grauverlaufs (oben) mit verschiedenen Rasterarten.



# Grundbegriffe

## ■ Raster (das)

In druckende und nicht druckende Bildelemente (Rasterpunkte oder -linien) zerlegte Fläche. Größe und/oder Anzahl der Bildelemente pro Fläche bestimmen den Tonwert.

## ■ Rasterpunkt

Grundelement eines Rasters, z. B. kreis-, ellipsen-, rautenförmig oder quadratisch.

### ▷ Hauptrichtung

Kreise, Rauten und Quadrate haben keine Hauptrichtung, denn eine Drehung um 90° ergibt dieselbe Form.

Ellipsen haben eine Hauptrichtung, denn erst die Drehung um 180° ergibt wieder dieselbe Form.

Der Vorteil von Rasterpunkten mit Hauptrichtung:

Sie erlauben doppelt so hohe Winkelabstände zwischen den drei dunklen Farbauszügen Magenta, Cyan und Schwarz (🔴 Seite 130).

## ■ Rasterfrequenz (Rasterfeinheit)

Anzahl der Rasterpunkte (oder Linien) pro Längeneinheit in der Richtung, in der sich der höchste Wert ergibt (= Winkellage des Rasters). Einheit: lpi (Linien/inch) oder lpc (Linien/cm). Ein 60-er Raster (typischer Wert im Offsetdruck) bedeutet 60 Rasterpunkte auf einer Länge von einem Zentimeter (= 152 Linien pro inch).

## ■ Rasterweite

Kehrwert der Rasterfrequenz; Abstand der Mittelpunkte benachbarter Rasterpunkte

## ■ Rasterzelle

2-dimensionale Matrix aus Rasterelementen (= Recorderelemente = REL) zur Erzeugung von Halbtönen. Die Größe der Rasterzelle entspricht der Rasterweite. Rasterelemente sind die kleinsten adressierbaren Ausgabepixel. Sie bestimmen die Ausgabeauflösung.

## ■ Druckpunkt (Spot)

Laserspot eines Belichters; entspricht einem Rasterelement, hat aber oft einen etwas größeren Durchmesser, sodass sich benachbarte Laserspots überlappen können.

Abbildung 10.3.4 erläutert einige der genannten Begriffe.

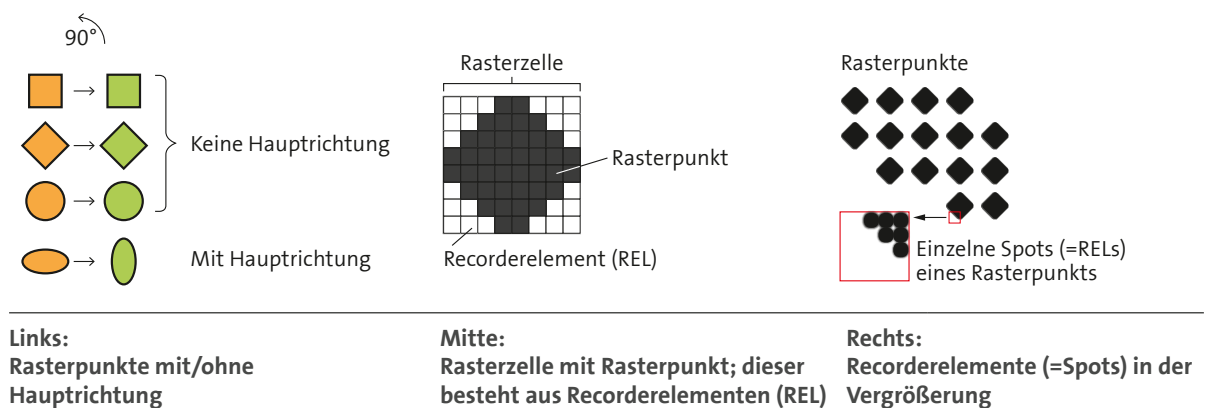


Abb. 10.3.4 | Grundbegriffe zu Raster.

## ■ Rastertonwert

Tonwert, der durch ein Raster erzeugt wird

## ■ Tonwert

ein Maß dafür, wie gedeckt eine Farbfläche einem Beobachter (bzw. für ein Densitometer) erscheint. Der Tonwert ist folglich kein *geometrisches* Maß für die Flächendeckung, sondern gibt die *optisch wirksame* Flächendeckung wieder.

Die Dichte wird relativ zur Dichte einer voll gedeckten Fläche (Volltonwert) angegeben (Murray-Davies-Formel, 🔴 Seite 49, kein Prüfungsstoff!).

### ■ Tonwertzuwachs (Dot Gain)

Grundsätzlich werden zwei Mechanismen unterschieden:

- ▷ Geometrische Veränderung (meistens Vergrößerung) der Rasterpunkte, etwa bei der Belichtung von Film oder Druckplatten oder bei der Farbübertragung auf das Papier.
- ▷ Lichtfang (Light Gathering, ► Abb. 4.2.3.2, Seite 49):  
Licht, das in unmittelbarer Nähe zu einem Rasterpunkt auf das Papier trifft, kann teilweise eindringen und wird wegen Streuung unter anderem Winkel zurückgeworfen und dabei vom Rasterpunkt „gefangen“.  
Lichtfang tritt folglich nicht bei transparenten Druckstoffen (Licht wird nicht remittiert) oder Druckstoffen auf, in die das Licht nicht eindringen kann (z. B. Blech).

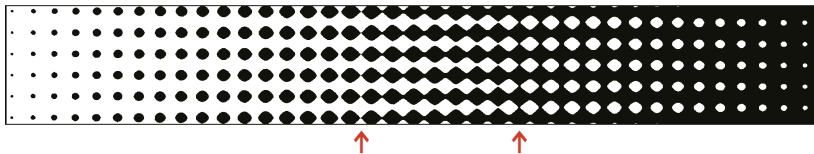
### ■ Punktschluss

Tonwertbereich, in dem sich die Rasterpunkte gerade eben berühren.

Dies ist mit einem deutlichen Tonwertsprung verbunden.

Abbildung 10.3.5 zeigt den Punktschluss bei einem Raster mit elliptischen Punkten.

Hier gibt es bei steigendem Tonwert zwei Punktschlüsse: zuerst in Längsrichtung (bei einem Tonwert von etwa 44 %), dann in Querrichtung der Ellipsen (bei etwa 61 %).



Der linke Pfeil zeigt den Punktschluss in Längsrichtung der Ellipsen, der rechte Pfeil den zweiten Punktschluss in Querrichtung.

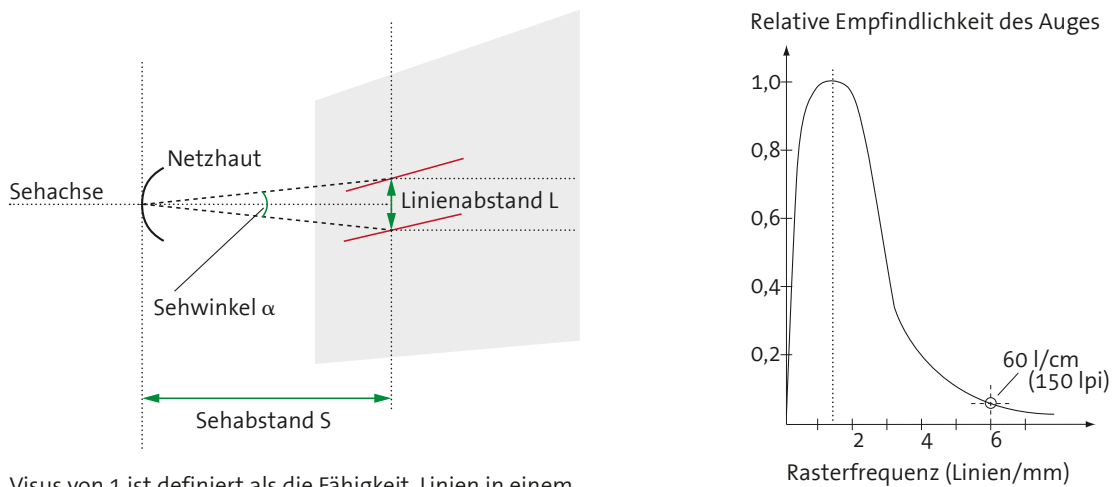
Abb. 10.3.5 | Punktschluss bei einem Raster mit elliptischen Punkten.

## Sichtbarkeit von Rasterstrukturen

Druckraster können heute so fein sein, dass sie kaum noch mit bloßem Auge erkennbar sind. Die Sichtbarkeit von Rasterstrukturen hängt ab von:

- ▶ Betrachtungsabstand
- ▶ Feinheit der Rasterstrukturen
- ▶ Linienunterscheidungsvermögen des Auges (Visus)
  - ▷ Visus ist ein Maß für die Fähigkeit des Auges, Linien bzw. Punkte getrennt wahrzunehmen.
  - ▷ Ein Visus von 1 bedeutet, dass das Auge Linien im Winkelabstand von einer Bogenminute (1') als getrennt wahrnehmen kann.
  - ▷ Wird der Visus ggf. mit Brille/Kontaktlinsen ermittelt, wird er mit cc gekennzeichnet (cum correctione).
  - ▷ Im Bevölkerungsdurchschnitt beträgt der Visus etwa 1,4 bis 1,5.
  - ▷ Die weit verbreiteten Prozentangaben für den Visus sind unsinnig.

<Ø> Abbildung 10.3.6 zeigt, wie die Sichtbarkeit von Rasterstrukturen von Visus, Betrachtungsabstand und Rasterfrequenz abhängt.



Visus von 1 ist definiert als die Fähigkeit, Linien in einem Sehwinkel von 1 Bogenminute (1') zu unterscheiden.

$$\tan \frac{\alpha}{2} = \frac{L/2}{S}$$

Für  $\alpha = 1'$  ( $1/60$  Grad), einen Visus von 1,0 und einen Sehabstand von 30 cm errechnet sich ein gerade noch wahrnehmbarer Linienabstand L von 0,087 mm.

Bei einem Visus von 1,5 beträgt  $L \approx 0,06$  mm

Bei einem Raster mit 60 l/cm beträgt der Abstand der Mittelpunkte der Rasterpunkte etwa 0,17 mm. Der Abstand zwischen den Rändern benachbarter Rasterpunkte ist aber z.B. bei einem Tonwert von 50% kleiner als die Hälfte des Abstandes der Mittelpunkte. Daher liegt ein Raster mit 60 l/cm bei 30 cm Betrachtungsabstand nahe an der Grenze des Auflösungsvermögens des Auges.

Abb. 10.3.6 | Punktschluss bei einem Raster mit elliptischen Punkten.

</Ø>

## Rasterzelle, Rasterpunkt, Druckpunkt

Die Größe einer Rasterzelle entspricht bei einem AM-Raster der Rasterweite. Beispiel: Ein Raster mit 60 Linien pro cm hat einer Rasterzellengröße von etwa  $1/60$  cm = 0,167 mm.

Eine Rasterzelle setzt sich aus einer Matrix von Druckpunkten zusammen. Die Druckpunkte sind die vom RIP adressierbaren Pixel (= Spots) des Belichters (Film oder Platte, Abb. 10.3.4 (Seite 122)).

Abbildung 10.3.7 zeigt, wie eine Rasterzelle durch unterschiedliche Adressierung der Recordelemente einen Tonwert von 50% erzeugen kann.

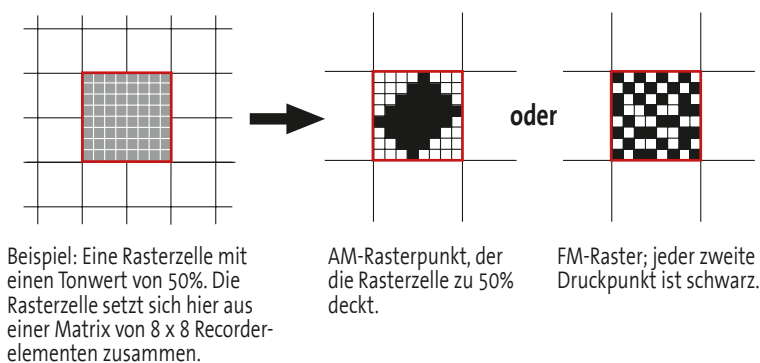


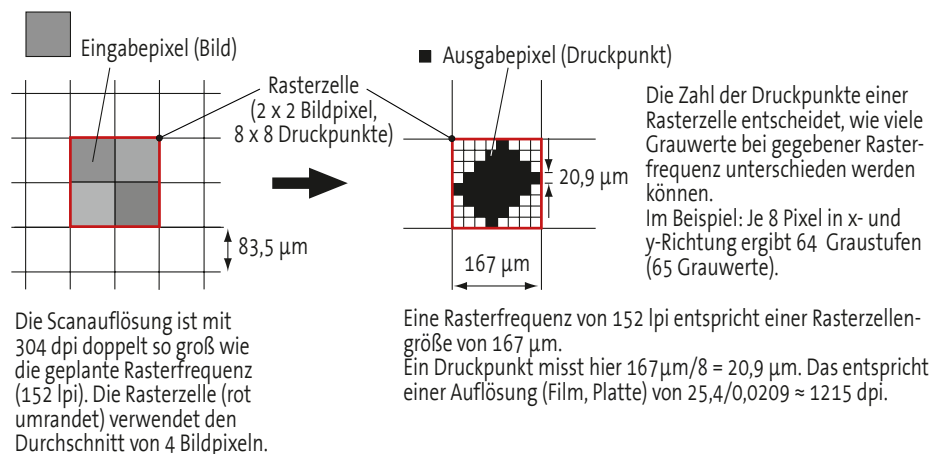
Abb. 10.3.7 | Rasterzelle aus 8 x 8 Recordelementen. Eine Flächendeckung von 50% kann durch unterschiedliche Ansteuerung der Druckpunkte erreicht werden.  
Quelle: Nach Kipphan

## Rasterfrequenz, Druckpunktgröße, Zahl der Grauwerte/-stufen

Aus dem Quantisierungstheorem (► Seite 45) folgt in Umkehrung:

Die Ausgabefrequenz (= Rasterfrequenz) sollte nur halb so hoch sein wie die Eingabefrequenz (Auflösung des Bildes).

Um ein Bild mit einer Rasterfrequenz von 60 Linien/cm (152 lpi) zu drucken, sollte es eine Auflösung von mindestens  $2 \times 152 = 304$  dpi haben (Qualitätsfaktor von 2). Beim Rastern bilden somit 4 Pixel der Bildvorlage (2 in jeder Dimension) eine Rasterzelle (Abb. 10.3.8).



**Abb. 10.3.8** | 2 x 2 Bildpixel bilden gemäß der Umkehrung des Quantisierungstheorems eine Rasterzelle, die in diesem Beispiel aus 8 x 8 Recorderelementen besteht.  
Quelle: Nach Kipphan

Die Zahl der Druckpunkte pro Rasterzelle bestimmt, wie viele unterschiedliche Flächen-deckungsgrade (= *Grauwerte*) mit einem Belichter/Drucker darstellbar sind. In der Abbildung 10.3.8 hat die Matrix der Rasterpunkte 8 x 8 Druckpunkte, d. h. es lassen sich 65 *Grauwerte* darstellen.

Wichtig:

Mit  $n$  Druckpunkten, die nur 2 Zustände kennen (binär, d. h. drucken oder nicht drucken), lassen sich  $n$  *Graustufen* darstellen. Die Zahl der *Grauwerte* ist immer um 1 größer als die Zahl der Graustufen. Dieser zusätzliche Grauwert ergibt sich, wenn keiner der im Beispiel 64 Druckpunkte angesteuert wird als 65. Grauwert (0%).

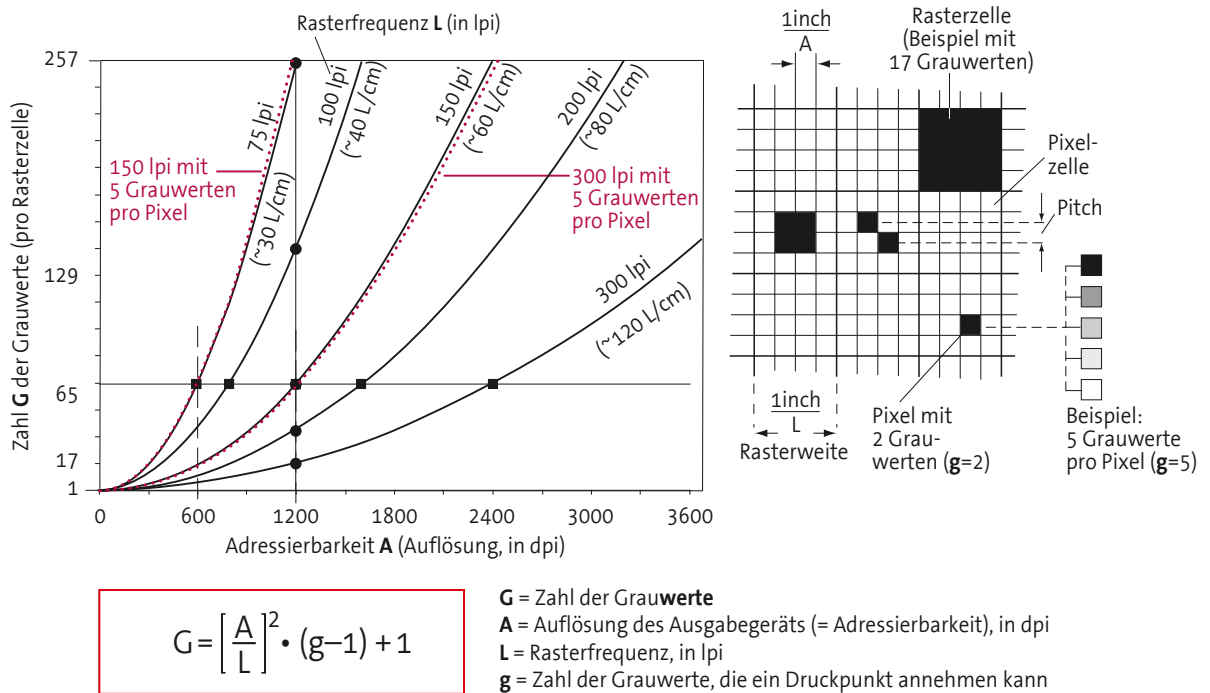
Bei quadratischen Rasterzellen und quadratischen binären Recorderelementen gilt für die Zahl der *Graustufen* folgende Beziehung:

$$\text{Zahl der Graustufen} = \left( \frac{\text{Ausgabeauflösung}}{\text{Rasterfrequenz}} \right)^2$$

*Ausgabeauflösung* ist die Zahl der Recorderelemente pro inch bzw. cm, die der Belichter/Drucker ansteuern kann.

Um 256 Graustufen (entspricht einem Bild mit einer Farbtiefe von 8 bit pro Kanal) darstellen zu können, muss das Verhältnis von Ausgabeauflösung zu Rasterfrequenz  $\sqrt{256} = 16$  betragen. Für ein Raster mit 60 Linien pro cm (= 152 dpi) ergibt das eine Ausgabeauflösung von etwa  $152 \cdot 16 = 2400$  dpi.

Im obigen Beispiel wurde die Zahl der darstellbaren Graustufen unter der Annahme ermittelt, dass ein einzelner Druckpunkt nur zwei Werte annehmen kann: drucken oder nicht drucken. Dies gilt z.B. für Laserdrucker oder Belichter. Wenn ein einzelner Druckpunkt mehr als 2 Zustände kennt, wie das z. B. im tiefenvariablen Tiefdruck der Fall ist, errechnet sich die Zahl der darstellbaren Grauwerte gemäß Abbildung 10.3.9.

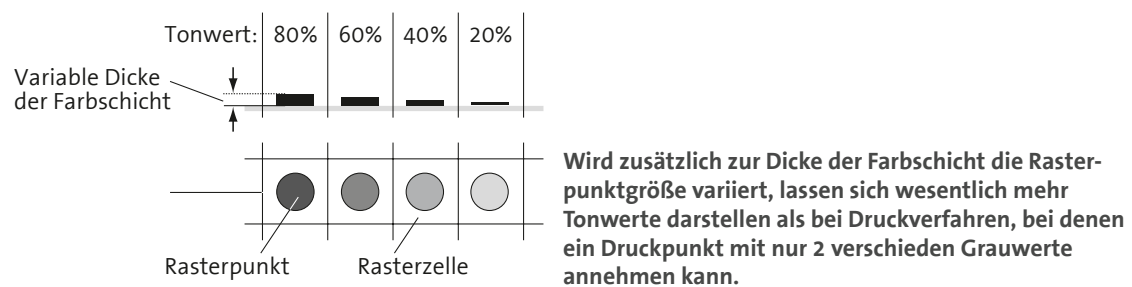


**Abb. 10.3.9 | Formel zur Ermittlung der darstellbaren Grauwerte für Drucksysteme, deren Druckpunkte mehr als 2 Zustände annehmen können. Die schwarzen Kurven gelten für  $g=2$ , die roten für  $g=5$ .**  
 Quelle: Nach Kipphan

## Dichtemodulation (variable Farbschichtdicke)

In einigen Non-Impact-Printing (NIP)-Verfahren und beim tiefenvariablen Tiefdruck ist eine Variation der Farbschichtdicke, also eine Modulation der optischen Dichte eines Druckpunkts möglich (Abb. 10.3.10). Tonwerte können bei diesen Druckverfahren sowohl durch Ändern der Flächendeckung als auch durch unterschiedliche Farbschichtdicke realisiert werden:

- ▶ Größerer Farbraum, optimierte Farbwiedergabe (weil mehr Abstufungsmöglichkeiten)
- ▶ Beim tiefenvariablen Tiefdruck muss nicht (kann aber) gerastert werden.



**Abb. 10.3.10 | Druckverfahren mit variabler Farbschichtdicke können verschiedene Tonwerte ohne Variieren der Rasterpunktgröße umsetzen.**  
 Quelle: Nach Kipphan

## Vergleich der Halbtonwiedergabe bei AM- und FM-Raster

Abbildung 10.3.11 zeigt die Umsetzung eines Halbtonbildes für ein Druckverfahren, das Halbtöne nur durch Variieren der Rasterpunktgröße (und damit der Flächendeckung) erzeugen kann, z. B. den Offsetdruck.

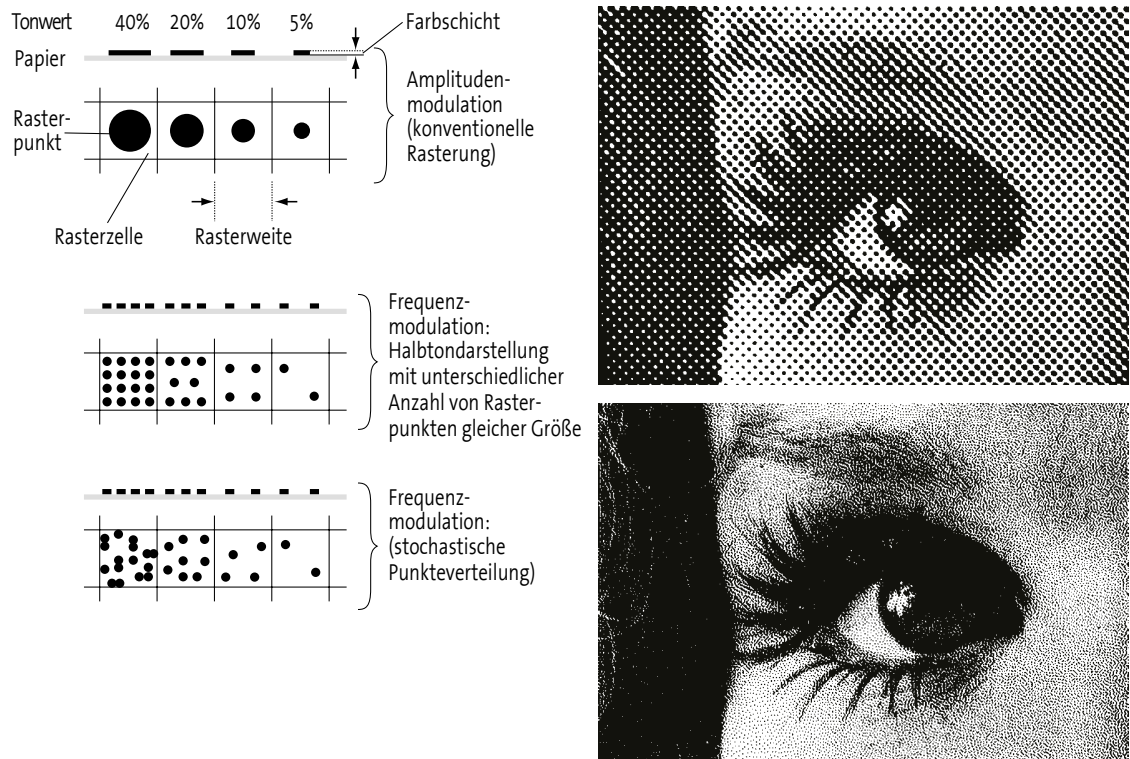


Abb. 10.3.11 | Halbtonwiedergabe mithilfe von AM- und FM-Raster.  
Quelle: Nach Kipphan

### 10.4 Amplitudenmodulierte (AM) Raster

#### ■ Amplitudenmodulation (AM)

Die Ausdehnung der Rasterpunkte wird variiert, die Anordnung folgt einem festen Muster (konstanter Abstand = konstante Rasterfrequenz). Der Abstand der Rasterpunkte entspricht der Rasterweite.

■ AM-Raster werden synonym auch als *autotypische Raster* bezeichnet.

#### ■ Rasterwinkel

Bei groben AM-Rastern können die Rasterpunkte als stark störend wahrgenommen werden. Am wenigsten stört im Allgemeinen ein Rasterwinkel von  $45^\circ$ . Einfarbige SW-Bilder werden daher meistens mit  $45^\circ$  gerastert (Abb. 10.4.1).

#### ■ Rasterwinkel im 4-Farb-Druck

Im 4-Farb-Druck müssen die einzelnen Farben in bestimmten Winkeln zueinander gedruckt werden. Der Grund: Werden gleiche Winkel gewählt, kann sich aufgrund minimaler Passerdifferenzen eine deutliche Farbdrift von Bogen zu Bogen einstellen (Abb. 10.4.2).

Hinweis:

Für die folgenden beiden Abbildungen wurde ein sehr grobes Raster gewählt, um Moiré-Effekte bei der Betrachtung am Bildschirm zu minimieren.



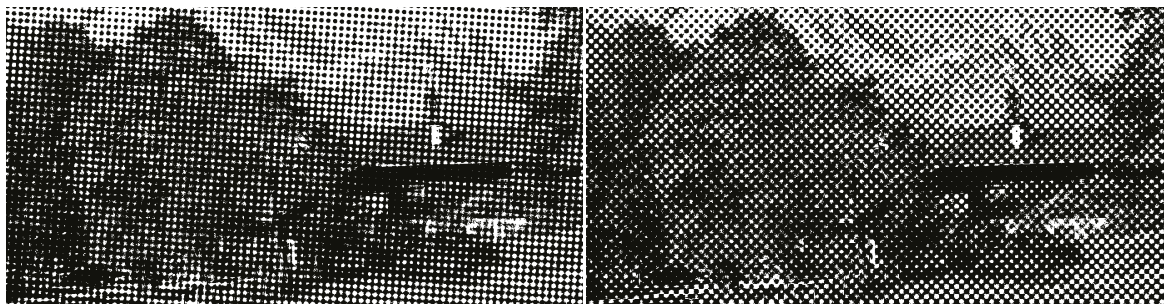


Abb. 10.4.1 | Unterschiedlicher Rasterwinkel bei einem S/W-Bild: links 5°, rechts 45°.

## Unterschiedliche Rasterwinkel für die einzelnen Farben

Wenn alle Farben im selben Winkel gedruckt werden, wird das Druckergebnis sehr empfindlich für Passerdifferenzen. Abbildung 10.4.2 zeigt das an einem simulierten Beispiel.

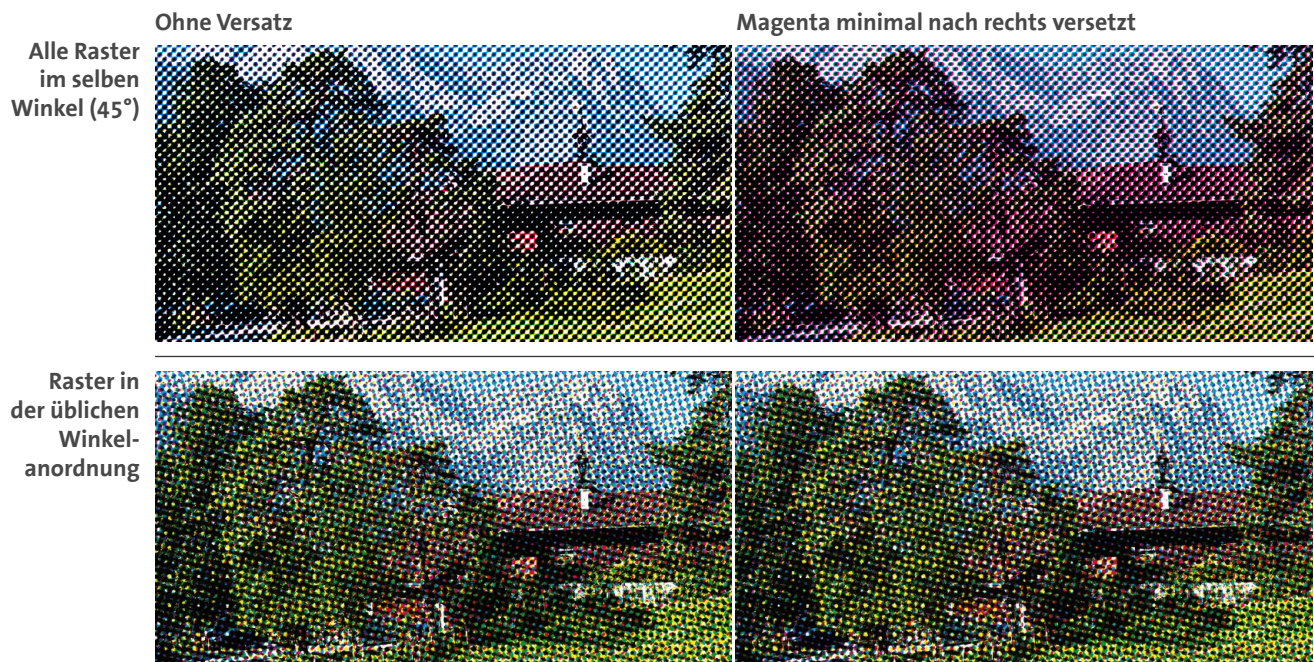


Abb. 10.4.2 | Werden alle Farben mit dem selben Winkel gedruckt (obere Reihe), bewirkt eine minimale Passerdifferenz (rechte Spalte, hier für Magenta) eine deutliche Farbveränderung. Bei der üblichen Rasterwinkelung (untere Reihe) ist die Farbveränderung durch den Versatz kaum wahrnehmbar.

## Moiré

Wenn sich zwei Schwingungen überlagern, wird das als Interferenz bezeichnet. Moiré ist das sichtbare Ergebnis solcher Überlagerungen, z. B. von Rastern mit unterschiedlicher Frequenz (Abb. 10.4.3) oder/und unterschiedlichem Winkel (Abb. 10.4.4).

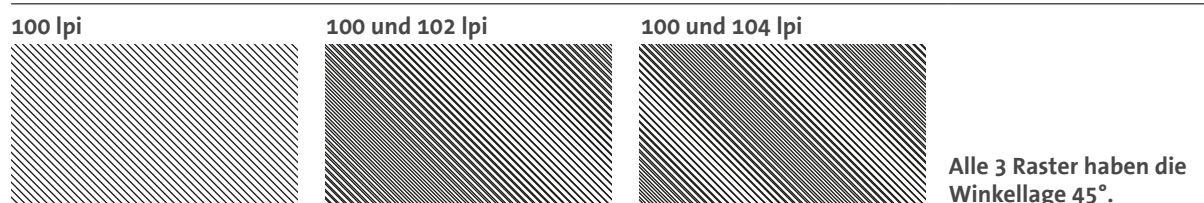


Abb. 10.4.3 | Moiré durch Überlagern eines Linienrasters (100 lpi) mit 102 lpi und 104 lpi.

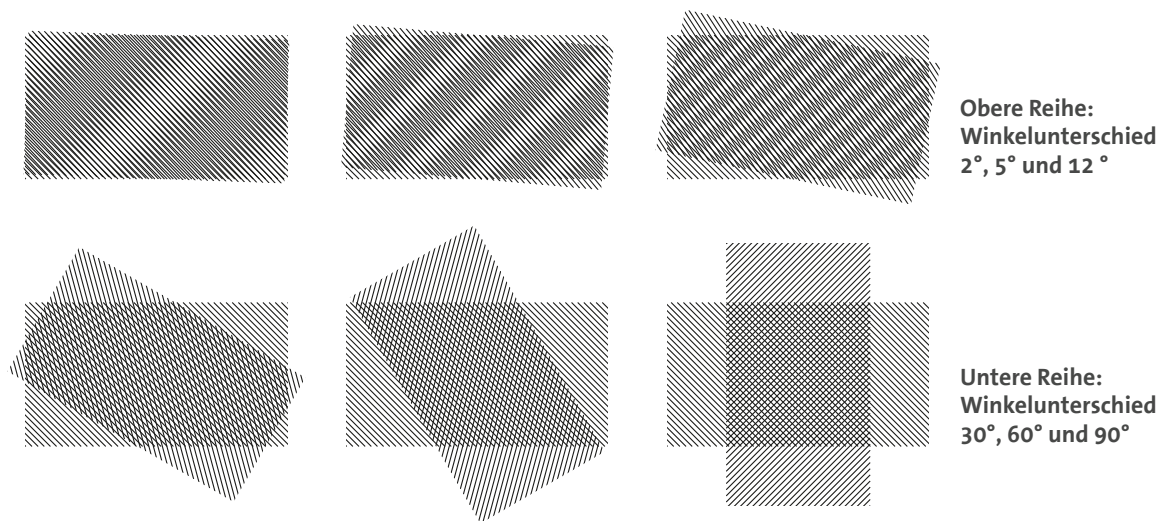


Abb. 10.4.4 | Moiré durch Überlagern eines Linienrasters (100 lpi) mit Rastern gleicher Frequenz, aber unterschiedlicher Winkellage.

Interferenzmuster sind geometrisch berechenbar, z.B. für Linienraster mit den Rasterfrequenzen  $K_1$  und  $K_2$  und dem Winkel  $\alpha$ :

$\langle \emptyset \rangle$  Moiréperiode  $e = \frac{K_1 \cdot K_2}{\sqrt{(K_1)^2 + (K_2)^2 - 2 \cdot K_1 \cdot K_2 \cdot \cos \alpha}}$   $\langle / \emptyset \rangle$

Auch bei der Überlagerung von Punktrastern mit unterschiedlicher Frequenz und/oder Winkellage entsteht Moiré (Abb. 10.4.5 und 4.10.6)

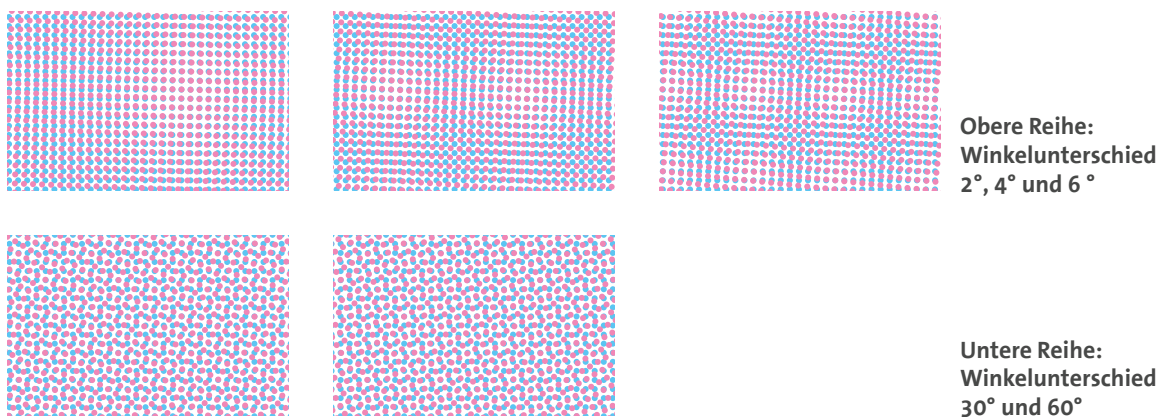


Abb. 10.4.5 | Moiré durch Überlagern von Punktrastern mit unterschiedlichem Winkel.

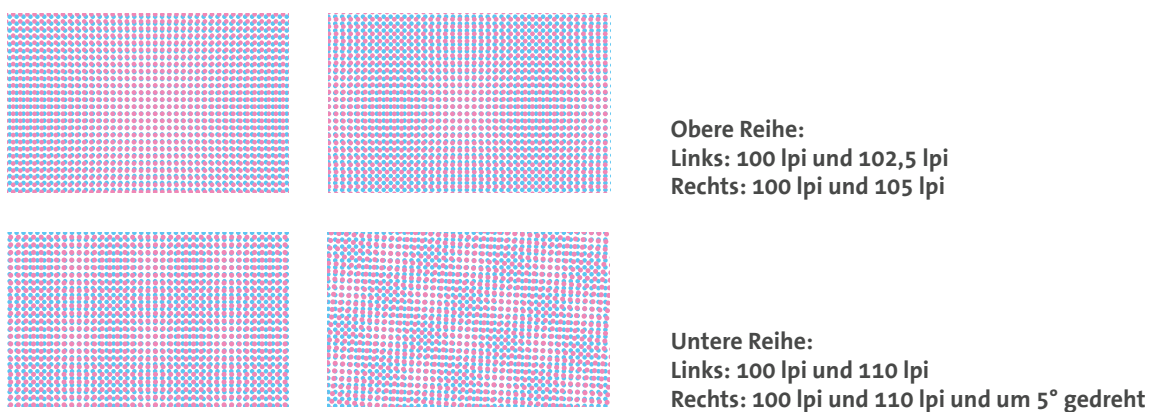


Abb. 10.4.6 | Moiré durch Überlagern von Punktrastern mit unterschiedlicher Rasterweite und darüber hinaus unterschiedlichem Winkel.



Neben dem Moiré, das durch Überlagerung unterschiedlicher Rasterfrequenzen und Rasterwinkel beim Mehrfarbendruck entstehen kann, gibt es weitere Formen von Moiré, die damit nicht verwechselt werden dürfen:

- ▶ Überlagerung durchsichtiger Strukturen des Motivs (Abb. 10.4.7) oder Überlagerung gleichmäßiger AM-Rasterstrukturen mit Strukturen des Bildes (Lamellen, Stoffstrukturen, Gitter usw.).
- ▶ Scannen bereits gerasterter Vorlagen mit zu niedriger Auflösung: Die Scanauflösung muss (mindestens) das Doppelte der Druckrasterfrequenz betragen (Quantisierungstheorem, [Seite 45](#)). Die durch Scannen mit zu niedriger Auflösung entstehenden Aliasingeffekte ([Abb. 4.1.3.1, Seite 46](#)) können durch kein Filter entfernt werden.
- ▶ Erneutes Rastern bereits gerasterter (gescannter) Vorlagen. Abhilfe ist möglich durch leichtes Weichzeichnen vor dem Rastern oder den Einsatz des Hochpassfilters. Bei Einsatz der Reprokamera ([Seite 119](#)) kann diese Form des Moirés durch leichtes Unschärfstellen abgemildert werden.



**Hinweis:** Wird dieses Bild an einem Monitor betrachtet, können sich je nach Zoomfaktor zusätzlich Moirés durch Überlagern mit dem Bildschirmraster ergeben.

Abb. 10.4.7 | Moiré durch Überlagerung von Strukturen des Motivs (hier: Papageienfedern).  
Quelle: [http://commons.wikimedia.org/wiki/File:Moire\\_on\\_parrot\\_feathers.jpg](http://commons.wikimedia.org/wiki/File:Moire_on_parrot_feathers.jpg)

## Vermindern von Moiré durch optimierte Rasterwinkel

Wie oben gezeigt, müssen die 4 Farben Cyan, Magenta, Gelb und Schwarz verschiedene Rasterwinkel haben, um das Druckergebnis weniger empfindlich für Passerdifferenzen zu machen.

Hinweis:

Die bei den nachfolgend beschriebenen Rasterwinkeln sichtbaren Rosetten sind auch eine Form von Moiré.

Die Empfehlung nach DIN 16547 aus dem Jahr 1962 für den 4-Farbendruck verteilt die 4 Farben auf 90°:

- Schwarz wird mit 45° Grad gerastert,
- Magenta und Cyan um 30° Grad versetzt (können je nach Motiv getauscht werden) und
- Gelb mit 15° Grad Abstand dazwischen.

Diese DIN-Norm ist historisch. Runde bzw. quadratische Rasterpunkte (d. h. ohne Hauptachse) zwangen damals dazu, die 4 Farben auf 90° zu verteilen. Elliptische Rasterpunkte (d. h. mit Hauptachse) erlauben heute eine Verteilung auf 180°.

Die aktuelle ISO 12647-2 schreibt daher einen Abstand von je 60° zwischen Cyan, Magenta und Schwarz sowie von 15° für Gelb vor. Die dominante Farbe soll den Winkel 45° oder 135° bekommen (Abb. 10.4.8, Seite 131).

Hinweis:

Je nach dem ob Hauttöne (aus Magenta und Gelb) oder Grüntöne (aus Cyan und Gelb) im Motiv überwiegen, können die Rasterwinkel von Magenta und Cyan getauscht werden, um den Winkelabstand der beiden jeweils dominierenden Farbanteile möglichst groß zu halten und dadurch die Störungen durch Überlagerung der Rasterpunkte zu verringern.

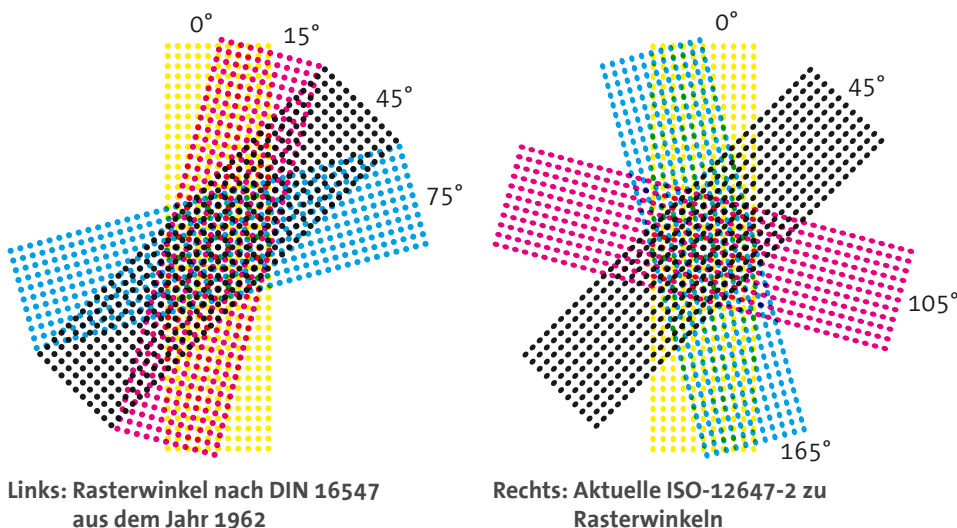


Abb. 10.4.8 | Normgemäße Rasterwinkel historisch (links) und aktuell (rechts).

FM-Raster benötigen prinzipiell keine Winklung.

## Rationale und irrationale digitale Raster

Bei digital erzeugten Rastern kommt der Tangens ins Spiel, weil die Drehung einer Rasterzelle nur in festen Schritten möglich ist, die von der Matrix der Recorderelemente vorgegeben sind (Abb. 10.4.9).

- Von einem *rationalen Raster* spricht man, wenn der Tangens des Rasterwinkels eine rationale Zahl ist (rationaler Tangens, daher als *RT-Raster* abgekürzt).
- Ein irrationales Raster liegt folglich vor, wenn der Tangens des Rasterwinkels eine irrationale Zahl ist (irrationales Rastern = irrational screening, IS). Irrationale Zahlen lassen sich nicht als Bruch aus ganzen Zahlen darstellen.

### Rationale Raster

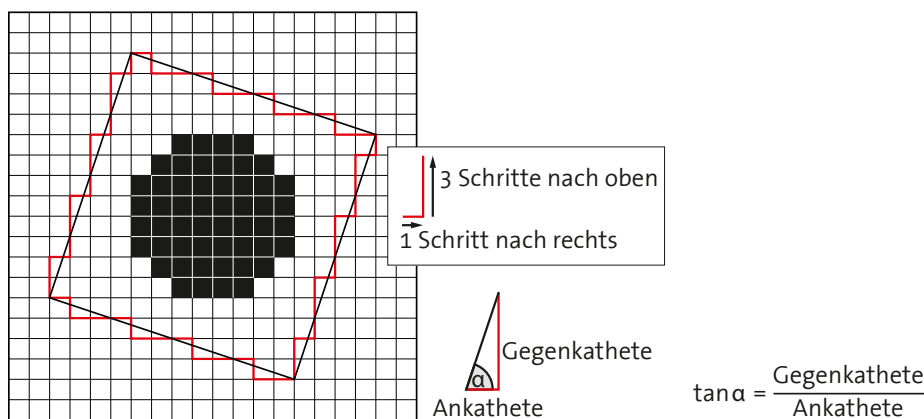


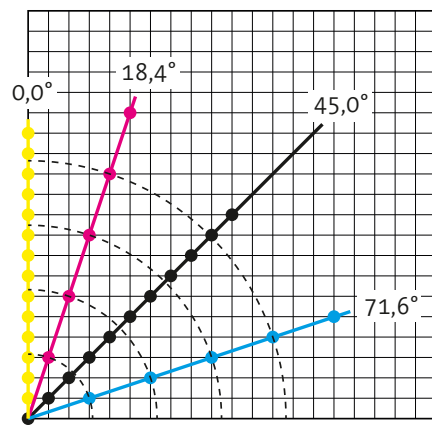
Abb. 10.4.9 | Die Drehung einer Rasterzelle in der rechtwinkligen Matrix ist nur in festen Schritten möglich.  
Quelle: Nach Kipphan

Abbildung 10.4.9 zeigt, warum nur 3 verschiedene Winkel für die Drehung der Rasterzelle möglich sind, die den Vorgaben von 15°, 45° und 75° (► Abb. 10.4.8, links) nahekommen:

- $\tan \alpha_1 = 3 : 1$ ;  $\rightarrow \alpha_1 \approx 71,6^\circ$
- $\tan \alpha_2 = 1 : 3$ ;  $\rightarrow \alpha_2 \approx 18,4^\circ$
- $\tan \alpha_3 = 1 : 1$ ;  $\rightarrow \alpha_3 = 45,0^\circ$

2 weitere mögliche Winkel mit rationalem Tangens ( $2 : 1 = 2$  sowie  $1 : 2 = 0,5$ ) weichen mit 63,4° bzw. 26,6° noch weiter von den Vorgaben ab.

Abbildung 10.4.10 zeigt, dass bei einem RT-Raster nicht nur die geforderten Rasterwinkel verfehlt werden, sondern auch die Rasterweiten unterschiedlich sind.



Die Viertelkreise durch die Mittelpunkte der Rasterpunkte für Cyan (71,6°) und Magenta (18,4°) schneiden die 45°-Linie (Schwarz) und die 0°-Linie (Gelb) nicht an den Stellen, an denen die für diese Farben jeweils möglichen Rasterpunkte liegen. Das bedeutet, dass die Rasterweiten der 4 Farben bei einem RT-Raster nicht gleich sein können.

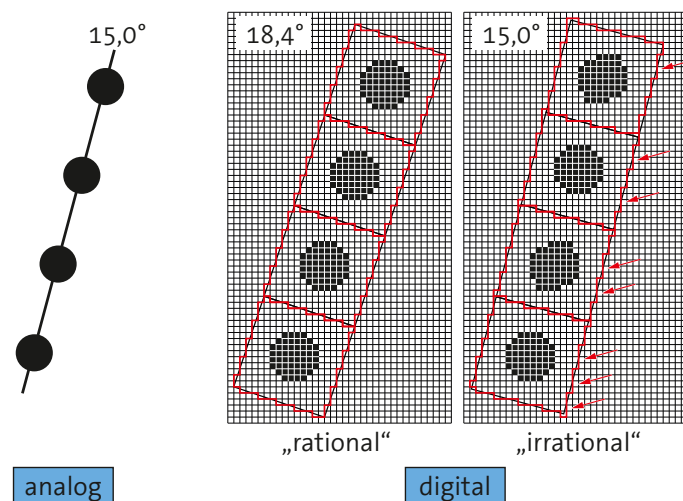
**Abb. 10.4.10** | RT-Raster erlauben weder die vorgegeben Rasterwinkel noch ermöglichen sie gleiche Rasterweiten für die 4 Farben.  
Quelle: Nach Kipphan

Gegenüber konventionell mit der Reprokamera erzeugten Rastern weisen digital erzeugte RT-Raster deutlich stärkere Moiré-Effekte auf. RT-Raster sind daher historisch und werden heute nicht mehr angewendet. Es handelt sich um das erste digitale Rasterverfahren, das in PostScript (► Seite 92) implementiert war. Die Nachteile der RT-Raster trugen (zu Recht) viel zu den anfänglichen Vorbehalten gegenüber digitalen Rastern bei.

### Irrationale Raster mithilfe von Superzellen

Durch Einführen sog. Superzellen wurde es möglich, die geforderten Rasterwinkel und Rasterweiten wesentlich genauer zu erreichen. Wie genau (z. B. bis auf 15,0013°), hängt von der Definition der Superzellen und dem daraus resultierenden Rechenaufwand ab.

Superzellen fassen mehrere Rasterzellmatrizen zu einer zusammen. Während bei einem RT-Raster mit einer einzelnen Rasterzelle nur Winkel möglich sind, die sich aus den ganzzahligen Schritten 3 rechts/1 hoch, 1 rechts/3 hoch und 1 rechts/1 hoch zusammensetzen (Abb. 10.4.9), können die Winkelvorgaben mit Superzellen besser angenähert werden, indem ab und zu z. B. die Schrittfolgen 4 rechts/1 hoch bzw. 1 rechts/4 hoch gewählt werden. Dies kann als Analogie zu den Schalttagen angesehen werden, mit deren Hilfe die Jahre an die tatsächliche Umlaufdauer der Erde angenähert werden. Die einzelnen Rasterzelle entspricht hier einem Jahr, die Superzelle dem Zeitraum von 4 Jahren. Abbildung 10.4.11 zeigt das Prinzip.



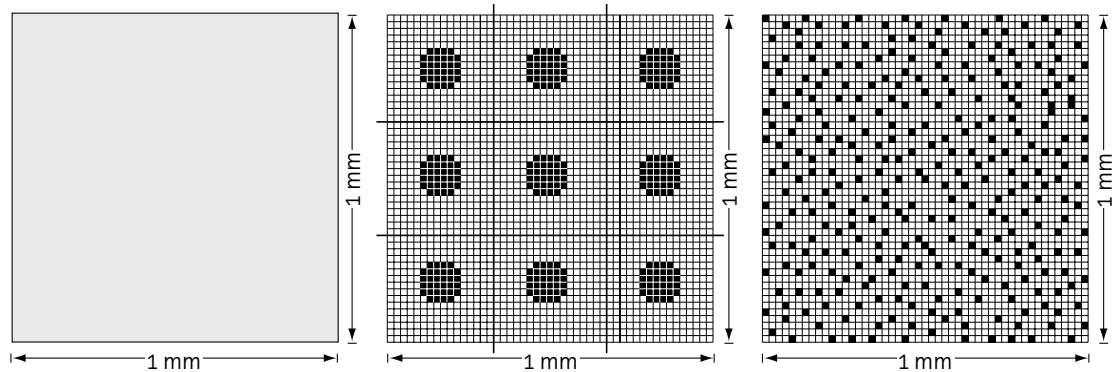
Beachte die Unterschiede der rot gezeichneten Umrisse der Rasterzellen zwischen „rational“ und „irrational“. Die roten Pfeile am rechten Rand der irrationalen Rasterzellen markieren Abschnitte mit einem Tangens von  $1/4$  (1 REL nach rechts, 4 RELs nach oben), der einem Winkel von  $14,04^\circ$  entspricht. Da einige Abschnitte einen Tangens von  $1/3$  haben ( $18,42^\circ$ ), ergibt sich für die Superzelle ein „durchschnittlicher“ Winkel von  $15,0^\circ$ .

Beachte die leichte Verzerrung einzelner Rasterpunkte bei der irrationalen Rasterung.

**Abb. 10.4.11** | Prinzip der Superzellen, das eine bessere Näherung der irrationalen Rasterwinkel erlaubt.  
Quelle: Nach Kipphan

## 10.5 Frequenzmodulierte (FM) Raster

Bei einem FM-Raster werden die einzelnen Druckpunkte einer Rasterzelle nicht zu einem Rasterpunkt gruppiert, sondern stehen einzeln über die Rasterzelle verteilt (Abb. 10.5.1).

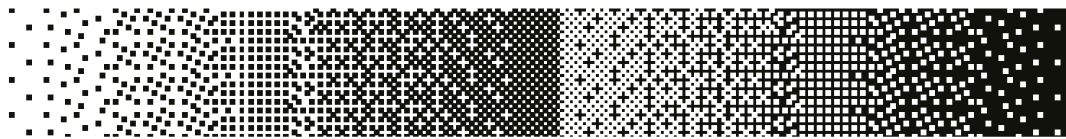


**Abb. 10.5.1** | Erzeugen eines Tonwerts von 12,5% (links) durch ein AM-Raster (Mitte) und ein FM-Raster (rechts).  
Quelle: Nach Kipphan

Für die Berechnung von FM-Rastern gibt es verschiedene Methoden: Dithering, Zufallsraster und Error-Diffusion-Algorithmen.

### Dithering

Für bestimmte Tonwerte werden geordnete Muster gleichgroßer Druckpunkte benutzt (Abb. 10.5.2). Die Zahl der Muster ist meistens eng begrenzt.



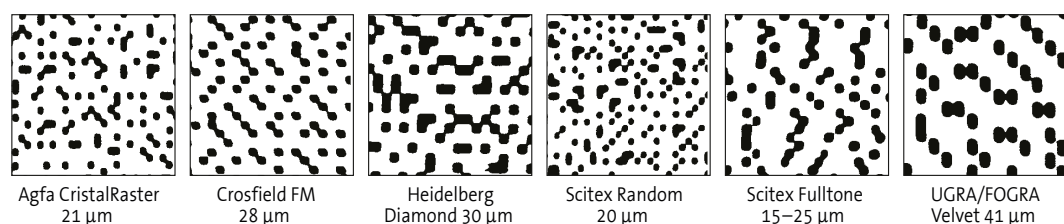
**Abb. 10.5.2** | Darstellung unterschiedlicher Grauwerte durch fest vorgegebene Dithering-Muster.  
Quelle: Nach Kipphan

### Zufallsraster

Zufallsraster werden u. a. im Offsetdruck eingesetzt. Für einen bestimmten Tonwertbereich der Druckvorlage wird per Zufall eine Punktanordnung erzeugt (Abb. 10.5.3). Die Algorithmen sind herstellerabhängig.

Erfolgt die Erzeugung der Rasterpunkte rein nach Zufall (FM-Raster der 1. Generation), kann die Dichte der Rasterpunkte sehr stark schwanken. Daher besteht bei gleichmäßigen Tonflächen die Tendenz zu unruhiger Farbwirkung.

Weitgehend vermeiden lässt sich das, wenn die zufälligen Schwankungen der Verteilung der Druckpunkte begrenzt werden (FM-Raster der 2. Generation). Die Verteilung der Druckpunkte ähnelt dann dem Muster von Silberkorn in der Fotografie (Abb. 10.5.3).



**Abb. 10.5.3** | Verschiedene FM-Raster der 2. Generation einzelner Hersteller. Die zufälligen Schwankungen der Dichteverteilung sind hier begrenzt.  
Quelle: Nach Kipphan



## Error-Diffusion-Algorithmen (EDA)

Für jeden Druckpunkt wird mithilfe einer Schwellenwertmatrix ermittelt, ob er weiß bleibt oder druckt. Im Allgemeinen gilt ein Schwellenwert von 50 %. Darunter wird auf 0 % abgerundet (druckt nicht), darüber auf 100% aufgerundet (druckt). Wenn sich die Rundungsfehler nicht über das ganze Bild gemittelt ausgleichen, können sich dadurch deutliche Helligkeitsverschiebungen ergeben.

Mithilfe verschiedener Error-Diffusion-Algorithmen (EDA) wird versucht, bei der Berechnung eines bestimmten Druckpunkts zu berücksichtigen, welche Rundungen bei den Druckpunkten der unmittelbaren Umgebung stattgefunden haben. EDAs werden oft in Tintenstrahl und Laserdruckern verwendet. Abbildung 10.5.4 zeigt das Prinzip der Floyd-Steinberg-Matrix. Dieser Algorithmus verändert immer nur Pixel, die noch nicht berechnet worden sind (nicht rekursiv) und lässt sich daher in einem einzigen Durchgang anwenden.

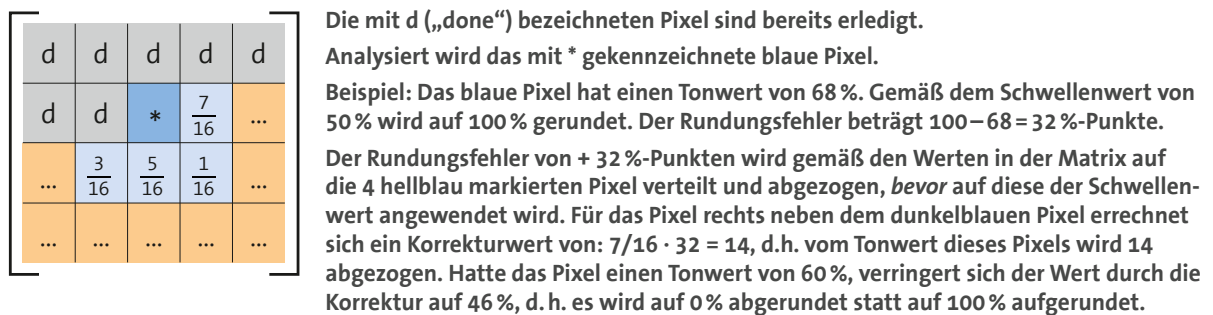


Abb. 10.5.4 | Floyd-Steinberg-Algorithmus zur Fehlerverteilung.

## 10.6 Hybridraster (XM-Raster, crossmodulierte Raster)

AM- und FM-Raster haben Stärken und Schwächen:

- ▶ AM-Raster
  - ▷ Vorteile:
    - Geringerer Tonwertzuwachs
    - Gleichmäßige Mitteltöne bei technischen Rasterflächen
    - Höhere Prozesssicherheit bezüglich Einhalten der ISO-Standards
  - ▷ Nachteile:
    - Moiré möglich (sowohl durch Übereinanderdruck der 4 Farben als auch durch Interferenz mit Strukturen des Motivs)
    - Rosetten erkennbar
    - Geringere Detailerkennbarkeit
- ▶ FM-Raster
  - ▷ Vorteile:
    - Kein Moiré, keine Rosetten
    - Nahezu fotorealistisches Druckergebnis, auch auf schlechteren Papiersorten
    - Höhere Detailerkennbarkeit
  - ▷ Nachteile:
    - Unruhige Wiedergabe von Rasterflächen mit gleichem Tonwert (sog. technische Raster)
    - Reproduzierbarkeit mit neu erstellten Platten schwieriger

Hybridraster (XM-Raster, crossmodulierte Raster) sind eine Kombination aus AM- und FM-Rastertechnik. Sie gleichen die jeweiligen Nachteile weitgehend aus. Das Prinzip dabei:

- ▶ In den Lichtern erfolgt eine FM-Rasterung.
- ▶ Zu den Mitteltönen hin wandelt sich die Rasterung in ein AM-Raster um.
- ▶ Zu den Tiefen hin wird zum FM-Raster zurückgewechselt.

Dadurch werden die Nachteile beider Rasterverfahren vermieden:

- ▶ Kaum Gefahr des Wegbrechens von Lichtern oder des Zulaufens von Tiefen (= Schwächen der AM-Raster).
- ▶ Technische Rasterflächen im Mitteltonbereich werden ohne Störungen gedruckt (= Schwäche vor allem der FM-Raster der 1. Generation)
- ▶ Thermische Druckplatten mit 20 µm großen Punkten sind verwendbar.

## <!> Zusammenfassung „Rastertechnik“

- Bei Druckverfahren, die den Farbauftrag nicht oder nur in geringem Umfang variieren können, muss zur Darstellung von Halbtönen gerastert werden.
- Analoge Raster lassen sich mithilfe einer Reprokamera oder als Kontaktraster erstellen. Analoge Raster sind amplitudenmoduliert (AM), d.h. der Abstand der Druckpunkte ist gleich, aber der Durchmesser schwankt in Abhängigkeit vom darzustellenden Tonwert.
- Für digitale Raster werden die Druckpunkte mithilfe einer Matrix berechnet. Digitale Raster können amplitudenmoduliert sein (AM-Raster, siehe oben) oder frequenzmoduliert (FM-Raster).  
Bei FM-Rastern gibt es keine regelmäßig angeordneten Rasterpunkte, sondern die einzelnen Druckpunkte einer Rasterzelle sind mehr oder weniger zufällig verteilt.
- AM-Raster können aus Rasterpunkten ohne oder mit Hauptachse bestehen.  
*Keine Hauptachse* liegt vor, wenn die Drehung um 90° dasselbe Objekt ergibt (Kreise, Rauten, Quadrate). Die Winkel für die 4 Druckfarben müssen sich dann auf 90° verteilen.  
*Mit Hauptachse* bedeutet, dass erst die Drehung um 180° dasselbe Objekt ergibt (Ellipsen). Rasterpunkte mit Hauptachse haben den Vorteil, dass die 4 Farben auf 180° verteilt werden können (weniger Moiré, ruhigere Struktur, siehe unten).
- AM-Raster erfordern, dass die 4 Farben unterschiedliche Rasterwinkel haben, weil sonst minimale Passerdifferenzen zu deutlichen Farbveränderungen führen.
- Um Moiré-Effekte minimal zu halten, müssen die 3 dunklen Farben Cyan, Magenta und Schwarz gemäß aktueller ISO-Norm einen Winkelabstand von 60° haben, Gelb als 4. Farbe liegt mit einem Abstand von 15° dazwischen. Außerdem müssen die Rasterfrequenzen für alle 4 Farben identisch sein.
- Als die ersten digitalen Raster eingeführt wurden, war nur eine sog. rationale Winkelung möglich (RT-Raster), weil eine einzelne Rasterzelle in der Matrix nur um feste Winkel von z. B. 18,4°, 45° und 71,6° (Winkel mit rationalem Tangens von 1/3, 1/2 bzw. 3/1) gedreht werden kann. Die damit möglichen Rasterweiten für die einzelnen Winkel unterscheiden sich.
- Diese Schwächen konnten erst mit Einführung der irrationalen Raster (irrational screening, IS) behoben werden. Prinzip ist das Zusammenfassen mehrerer Rasterzellen zu Superzellen, wodurch sich die Zielvorgaben für Rasterwinkel und Rasterweiten besser annähern lassen.
- FM-Raster haben keine Rasterweite und benötigen keine Winkelung.
  - ▷ Zufallsraster  
Die Verteilung der Druckpunkte innerhalb einer Rasterzelle erfolgt nach dem Zufall, allerdings wird die Schwankungsbreite der Dichte begrenzt (FM-Raster der 2. Generation), um glatte Flächen gleichmäßiger ausgeben zu können.
  - ▷ Dithering  
Einzelne Tonwerte werden mit fest vorgegebenen Punktmustern dargestellt.
  - ▷ Error-Diffusion-Algorithmen  
Sie haben das Ziel, den Rundungsfehler für ein einzelnes Pixel auf benachbarte Pixel zu verteilen.
- Hybridraster kombinieren AM- und FM-Raster:  
In den Lichtern und in den Tiefen folgen sie einem FM-Raster, in den Mitteltönen einem AM-Raster. Dadurch werden die Nachteile beider Rasterverfahren weitgehend verringert.





