

---

**Digitale Darstellung von Bildern**  
**Theorie (L)**

---

# 1 Rastergrafik

## 1.1 Eingabegeräte

Digitale Photoapparate, Videokameras und Scanner liefern uns Bilddaten, die wir bearbeiten und möglichst platzsparend (und ohne Informationsverlust) auf einem Datenträger speichern wollen.



Abbildung 1: Digitalkamera (S. Klein/Wikipedia, 5.7.2007)

### CCDs

Zentraler Bestandteil vieler dieser Geräte ist eine Matrix aus winzigen helligkeitsempfindlichen Sensoren. Mittlerweile (2013) sind verschiedene Technologien im Einsatz: *CCD* (Charged-coupled Device) oder *APS* (Active Pixel Sensor).

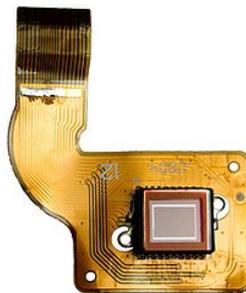


Abbildung 2: CCD-Element; C-M/Wikipedia, 23. Mai 2007

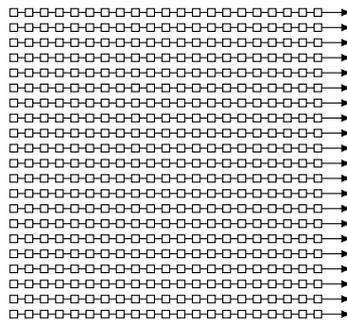


Abbildung 3: Schematische Darstellung eines Bildsensors ( $20 \times 20$  Pixel)

## Die Funktionsweise

Trifft ein Lichtstrahl auf eines der vielen Sensorelemente, so verursacht dies eine Ansammlung von Ladungen, die proportional zur Lichtintensität an dieser Stelle ist.

Diese Ladungen werden zunächst schrittweise an den Rand des Elements transportiert und von dort in einem schnellen Takt an einen Verstärker weitergereicht, der sie in eine Spannung umwandelt.

Durch die Digitalisierung dieser Spannungen erhält man ein Graustufenbild.

## Farbbilder

Um farbige Bilder zu erfassen gibt es verschiedene Verfahren. Eines davon spaltet einen einfallenden Lichtstrahl mittels eines kleinen Prismas in seine spektralen Komponenten (Rot, Grün, Blau) auf und führt sie dann drei verschiedenen CCD-Elementen zu. Ein anderes Verfahren verwendet eine Folie mit unterschiedlich absorbierenden Bereichen, so dass die darunter liegenden Pixelsensoren jeweils nur „ihr“ Frequenzspektrum messen.

In diesem Zusammenhang bezeichnet der Begriff *Farbtiefe* die Anzahl der darstellbaren Helligkeitswerte bzw. Farben pro Pixel.

Bei einem Flachbettscanner sind diese Sensoren in einer Linie angeordnet und zusammen mit einer Lampe auf einem Schlitten montiert. Dieser Schlitten wird unter einer Glasplatte durchgezogen, auf der das Original liegt. Die Lampe beleuchtet das Original und die CCD-Sensoren messen die Intensität des reflektierten Lichts. Durch einen Schrittmotor wird der Schlitten weitertransportiert. Auf diese Weise wird das zu scannende Objekt zeilenweise erfasst. Bei einem Einzugsscanner kann das Original auch über eine stationäre CCD-Einheit transportiert werden.

## Rastergrafiken

Die oben beschriebenen Verfahren produzieren Bilder, die aus rechteckig angeordneten Bildpunkten (Pixeln) zusammengesetzt sind. Daher spricht man von *Pixelgrafiken*, *Rastergrafiken* oder *Bitmaps*. Zur Speicherung solcher Grafiken stehen verschiedene Grafikformate zur Verfügung. Hier nur einige Beispiele:

## 1.2 Bildformate

### PNG

PNG (Portable Network Graphics): Rastergrafikformat mit verlustfreier Kompression. Es sind bestimmte Farbtiefen im Bereich von 1 Bit (schwarzweiss) bis 48 Bit (jeweils 16 Bit pro Farbkanal) möglich. Dieses Format eignet sich für die platzsparende Speicherung von Liniengrafiken (Logos, Pläne, Diagramme, ...)

### JPG oder JPEG (Joint Photographic Experts Group)

Rastergrafikformat mit verlustbehafteter oder verlustfreier Kompression. Der Standard sieht die Farbtiefen 8 Bit und 12 Bit vor. Üblicherweise wird dieses Format in der verlust-

behafteten Version mit einer Farbtiefe von 8 Bit verwendet. Dieses Format eignet sich für die platzsparende Darstellung von Fotografien.

### **TIFF (Tagged Image File Format)**

Rastergrafikformat zur verlustfreien Speicherung von Bilddaten. Es unterstützt bis zu 32 Bit Farbtiefe pro Farbkomponente. Es eignet sich für Bilder, die in hoher Qualität (z. B. für den Druck) dargestellt werden sollen.

### **BMP (Windows Bitmap):**

Rastergrafikformat zur verlustfreien Speicherung von Bilddaten. Die Daten werden mit einem einfachen Kompressionsverfahren komprimiert. Die Kompression ist jedoch nicht so leistungsfähig wie die des PNG-Formats. Der Einsatzbereich ist jedoch derselbe wie bei PNG.

## **1.3 Die PXM-Formate**

### **Übersicht**

Diese drei Bitmap-Dateiformate erlauben ein einfaches Speichern von Bildern:

- PBM: Portable BitMap (Monochrom)
- PGM: Portable GreyMap (Graustufen)
- PPM: Portable PixMap (Farbbilder)

Da diese Grafikformate unkompliziert sind, eignen sie sich gut für die Programmierung. Ein Grafikprogramm, das diese drei Bildformate lesen kann, ist „Gimp“. Damit können selber erzeugten Bilddateien in ein gebräuchlicheres Format wie PNG oder JPEG transformiert werden.

### **Bildaufbau**

1. „Magic Character“: PBM=P1, PGM=P2, PPM=P3
2. Breite des Bildes in Pixeln
3. Höhe des Bildes in Pixeln
4. nur PGM- und PPM-Format: Maximalwert für die Helligkeit (max. 255)
5. Liste der Bildpunkte
  - PBM: eine Zahl pro Bildpunkt (0=Weiss, 1=Schwarz)
  - PGM: eine Zahl pro Bildpunkt (0=Weiss, max=Schwarz)
  - PPM: drei Zahlen pro Bildpunkt (0=Weiss, max=R/G/B)

Die Daten sind jeweils durch mindestens ein Leerzeichen, einen Tabulator oder eine Zeilenschaltung zu trennen. Keine Zeile sollte mehr als 70 Zeichen haben.

## Beispiel 1: PBM-Format

```
P1 # Identifier fuer PBM-Dateien im ASCII-Format
5 7 # Breite: 5 Pixel, Hoehe: 7 Pixel
1 1 1 1 1 # 0=Weiss, 1=Schwarz
1 0 0 0 0
1 0 0 0 0
1 1 1 1 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
```



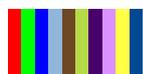
## Beispiel 2: PGM-Format

```
P2 # Identifier fuer PGM-Dateien im ASCII-Format
21 7 # Breite: 21 Pixel, Hoehe: 8 Pixel
9 # Graustufen-Maximum (0=Schwarz, ..., 9=Weiss)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 2 2 2 0 4 4 4 4 0 6 6 6 6 0 8 8 8 8 0
0 2 0 0 0 0 4 0 0 0 0 6 0 0 0 0 8 0 0 8 0
0 2 2 2 0 0 4 4 4 0 0 6 6 6 0 0 8 8 8 8 0
0 2 0 0 0 0 4 0 0 0 0 6 0 0 0 0 8 0 0 0 0
0 2 0 0 0 0 4 4 4 4 0 6 6 6 6 0 8 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```



## Beispiel 3: PPM-Format

```
P3 # Identifier fuer PPM-Dateien im ASCII-Format
10 5 # Breite: 10 Pixel, Hoehe: 5 Pixel
7 # Rot-Gruen-Blau-Maximum (maximale Intensitaet 0-7)
7 0 0 0 7 0 0 0 7 4 5 6 3 2 1 5 6 2 2 0 3 6 4 7 7 7 2 0 2 4
7 0 0 0 7 0 0 0 7 4 5 6 3 2 1 5 6 2 2 0 3 6 4 7 7 7 2 0 2 4
7 0 0 0 7 0 0 0 7 4 5 6 3 2 1 5 6 2 2 0 3 6 4 7 7 7 2 0 2 4
7 0 0 0 7 0 0 0 7 4 5 6 3 2 1 5 6 2 2 0 3 6 4 7 7 7 2 0 2 4
7 0 0 0 7 0 0 0 7 4 5 6 3 2 1 5 6 2 2 0 3 6 4 7 7 7 2 0 2 4
```



## Bemerkung

Die oben beschriebenen Bildformate speichern die Daten als Zeichenketten im ASCII-Format (7-Bit) und können daher auch von Menschen „gelesen“ und ohne spezielle Codierung in Textmails verschickt werden.

Es gibt jedoch noch die entsprechenden Binärversionen dieser drei Formate (P4, P5 und P6), welche die Daten binär und damit etwas platzsparender speichern. Wer genaueres darüber wissen will, kann sich beispielsweise unter

[https://de.wikipedia.org/wiki/Portable\\_Anymap](https://de.wikipedia.org/wiki/Portable_Anymap)  
informieren.

## 1.4 Ausgabegeräte

Die Ausgabegeräte von Computern sind in der Regel auch Rastergeräte. Ein (LCD)-Bildschirm ist aus Pixeln aufgebaut. Ein Laser- oder Tintenstrahldrucker baut das Bild aus einzelnen Bildpunkten auf.

# 2 Vektorgrafik

Auf der anderen Seite möchte man aus (geometrischen) Daten Diagramme, Pläne, Schriften und Grafiken erzeugen.

In diesem Fall ist eine geometrische Beschreibung eines Bildes mittels Punkten, Linien und Flächen vorteilhaft. Eine solche *Bildbeschreibungssprache* ermöglicht eine sehr kompakte Darstellung von Grafiken. Da man üblicherweise Vektoren zur Beschreibung von Punkten und damit auch von Linien und Flächen verwendet, nennt man diese Art von Grafik auch *Vektorgrafik*.

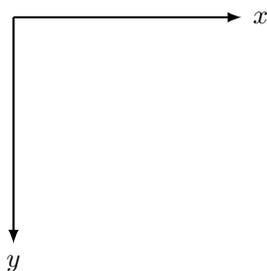
## 2.1 SVG (Scalable Vector Graphics)

### Übersicht

SVG ist ein auf XML basierendes offenes Grafikformat.

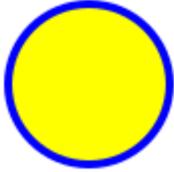
Die Bildbeschreibung erfolgt in einer für Menschen lesbarer Textform und wird üblicherweise von einem Interpreter im Browser ausgeführt und auch dort ausgegeben.

Im Gegensatz zum Koordinatensystem, das in der Schule verwendet wird, nehmen die  $y$ -Koordinaten von oben nach unten zu.



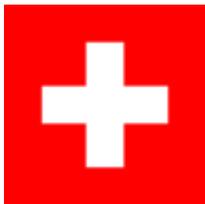
## Beispiel 1

```
1 <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
2   <circle cx="50" cy="50" r="40" stroke="blue"
3     stroke-width="4" fill="yellow" />
4 </svg>
```



## Beispiel 2

```
1 <svg xmlns="http://www.w3.org/2000/svg" width="100" height="100"
2   viewBox="0 0 32 32">
3   <path d="m0 0h32v32h-32z" fill="#ff0000"/>
4   <path d="m13 6h6v7h7v6h-7v7h-6v-7h-7v-6h7z" fill="#fff"/>
5 </svg>
```



## Beispiel 3

```
1 <svg xmlns="http://www.w3.org/2000/svg" width="200" height="120">
2   <text x="30" y="40" stroke="blue" stroke-width="2px"
3     fill="none" font-size="50px" transform="rotate(30)">
4     Hello</text>
5 </svg>
```



Im Beispiel wurde nur eine sehr kleine Teilmenge von SVG-Anweisungen gebraucht.

Auch wenn die obigen SVG-Bilder auf Pixel-Koordinaten definiert sind, können sie praktisch ohne Genauigkeitsverlust skaliert, d. h. vergrößert oder verkleinert werden. Erst für die Ausgabe auf einem Rastergerät müssen die Bildinformationen *gerastert* werden. Dafür kann aber eine ausreichend hohe Auflösung gewählt werden.

Die oben beschriebene Art der grafischen Darstellung eignet sich insbesondere für Logos oder für Computerschriften. Die Umrisse einzelnen Objekte bzw. Zeichen werden durch

Stützpunkte definiert, die anschliessend durch gerade oder runde Liniensegmente verbunden werden. Danach werden die Umrisse mit der Schriftfarbe gefüllt. Auf diese Weise haben auch vergrösserte Schriften eine glatte, d. h. „treppenfrie“ Qualität.

Neben SVG gibt es noch weitere Vektorgrafikformate:

### **PostScript**

Postscript (PS) ist eine Bildbeschreibungssprache, die in den 1980er Jahren von der Firma Adobe entwickelt wurde. Die Bildbeschreibung in Textform wird von einem Programm (dem Postscript-Interpreter) ausgewertet und auf einem Rastergerät (Bildschirm, Drucker) ausgegeben.

### **PDF (Portable Document Format)**

Ein Vektorgrafikformat von der Firma Adobe. Es handelt sich dabei um eine Weiterentwicklung von PostScript.

## **3 Vergleich von Raster- und Vektorgrafiken**

### **Vorteile der Rastergrafik**

- Die Dateiformate sind in der Regel programmunabhängig
- Reichtum an Farbabstufungen
- diverse Grafikformate für unterschiedliche Einsatzbereiche
- Bearbeitung jedes einzelnen Bildpunkts möglich
- Flächen können verschoben und kopiert werden
- Rasterung ist einfach, da die Bilddaten bereits gerastert vorliegen.
- Eignet sich für komplexe Bilder, Digitalfotos, gescannte Bilder, Icons

### **Vorteile der Vektorgrafik:**

- Skalierung mit geringem Qualitätsverlust (Achtung: bei einer Vergrösserung können Linien zu dünn werden oder die Schrift zeigt Knicke auf).
- gute Kompressionseigenschaften
- geringere Dateigrösse, da nicht jeder Bildpunkt gespeichert werden muss.
- Eigenschaften einzelner Kurven, Linien und Flächen bleiben erhalten und lassen sich dadurch nachträglich bearbeiten.
- Text kann in der Regel direkt bearbeitet werden.
- Eignet sich für Diagramme, Firmenlogos, geometrische Figuren und Schriften, technische Zeichnungen