Multimedia Databases

The Image Medium

Prof. Dr. Michael Granitzer
Prof. Dr. Harald Kosch

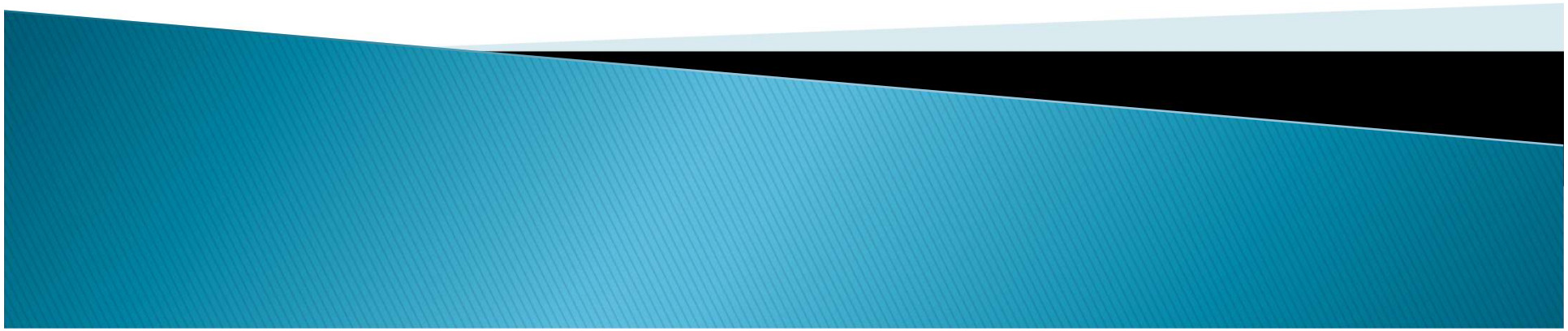


Table of Contents
Image

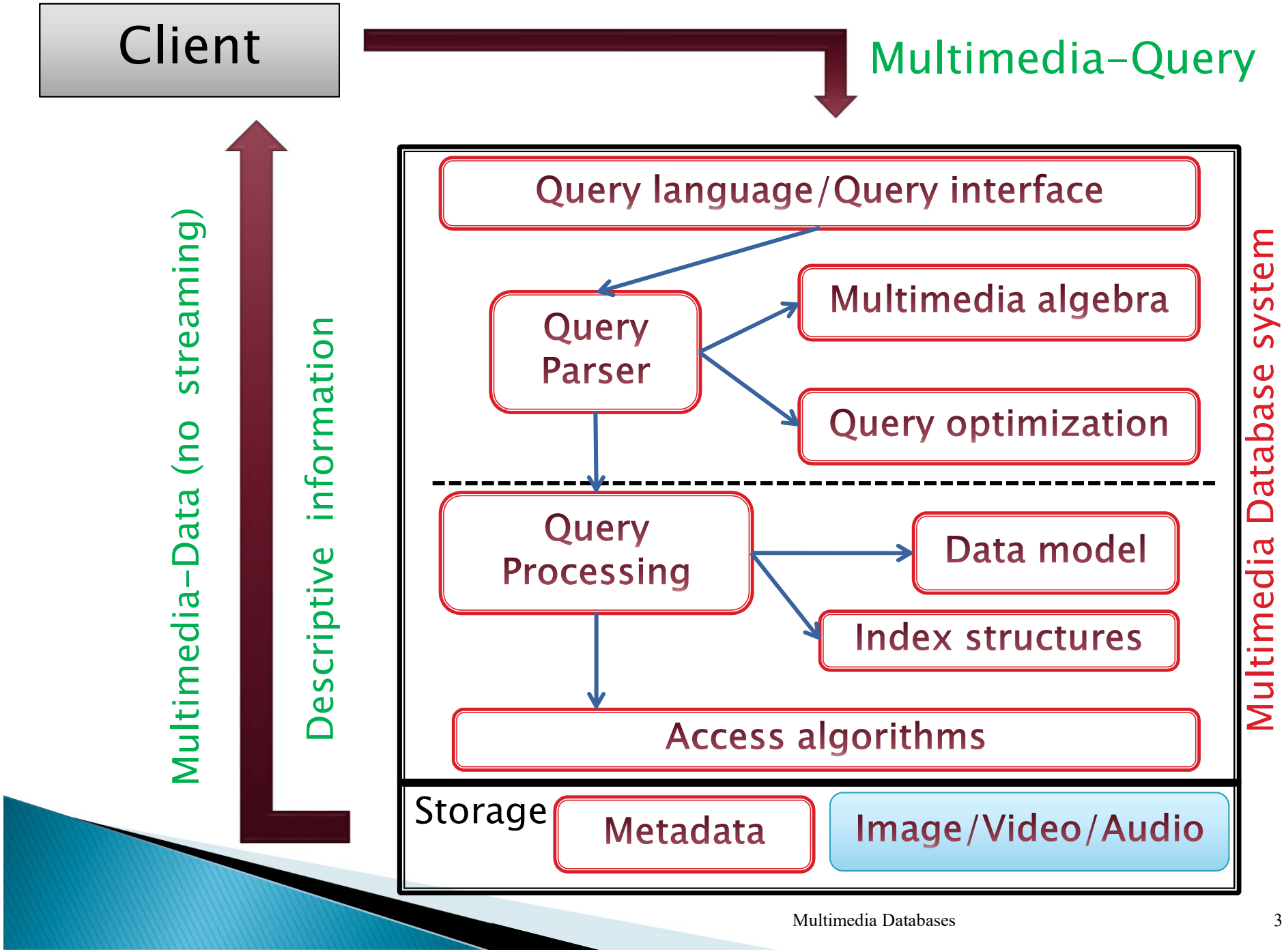
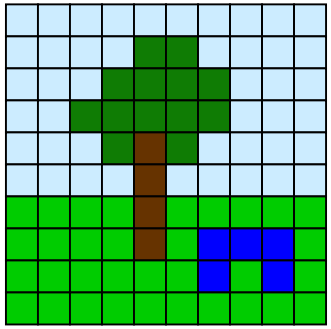

Client
Multimedia-Query
Multimedia-Data (no streaming)
Descriptive information
Query language/Query interface
Query Parser
Multimedia algebra
Query optimization
Query Processing
Data model
Index structures
Access algorithms
Storage
Metadata
Image/Video/Audio
Multimedia Database system

Table of Contents
Image

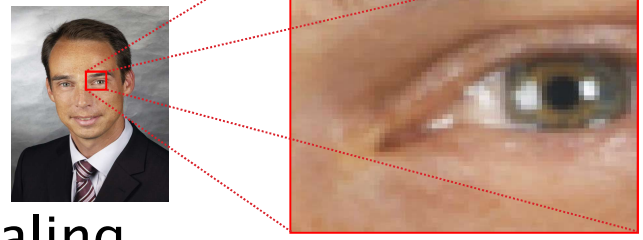
Basics
Digital Images

- „ Bitmapped Images“: Assignment of color values to single Image Elements



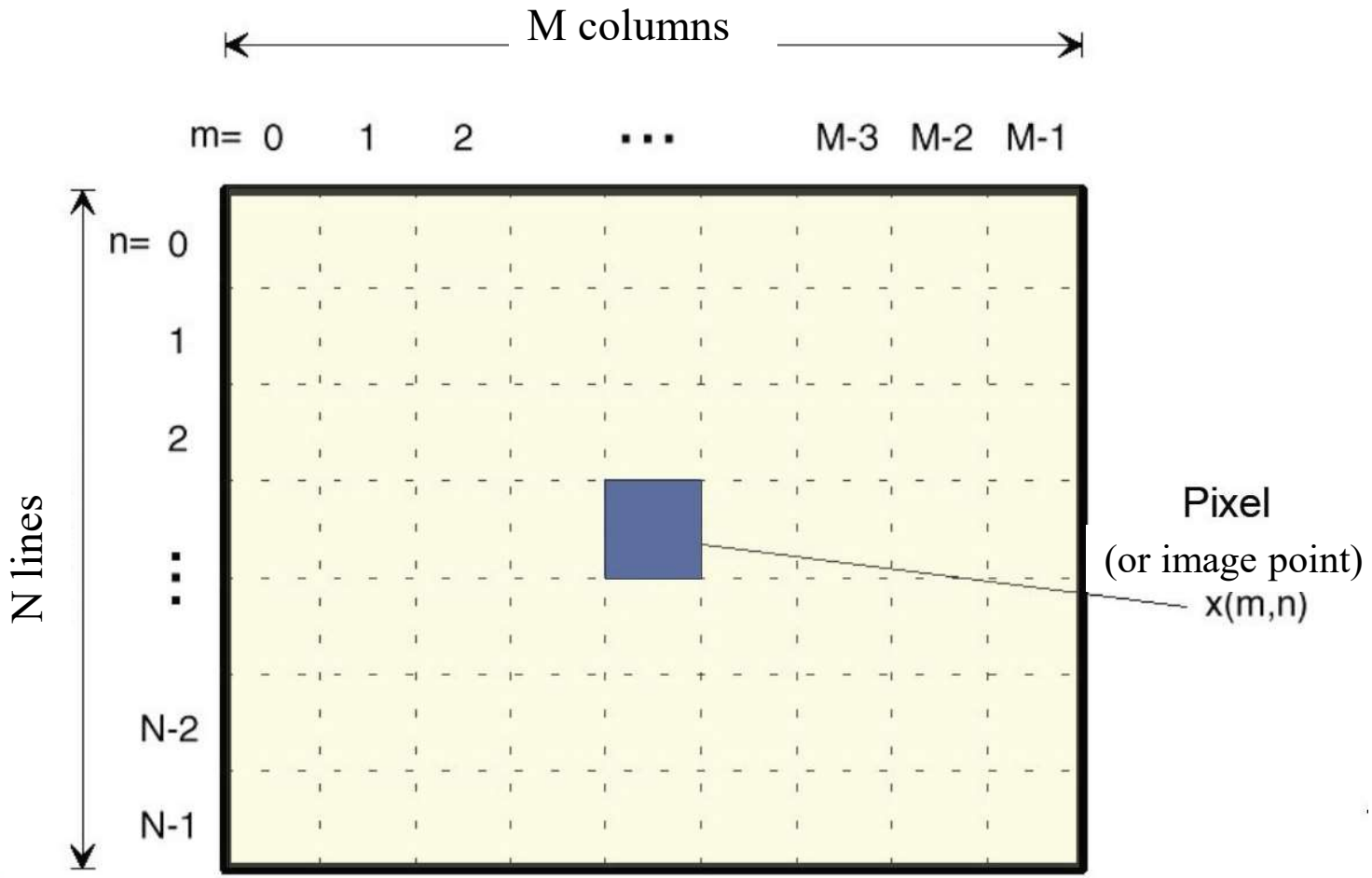
- Sources: scanner, digital camera, drawing programs, …

  Drawing programs use analogies: brush, spray can, …



- Disadvantages: need of memory space, scaling

Basics
Digitale Images II



Basics
Digital Image Editing and Image Processing

- Editing
  - Means the modification (Improvement, alteration or manipulation) of digital images with specialized software (for example Adobe Photoshop).

- Processing
  - Provides the mathematical algorithms used for editing and professionally modifying/analyzing digital images

Basics
Uses of Digital Impage Processing

- „Image improvement“
  - Image restauration, Retouche
- Image analysis, Image evaluation
  - Segmentation, analysis of texture
  - Extraction of object contours
- Image classification, Image recognition, Image sorting

Table of Contents
Image

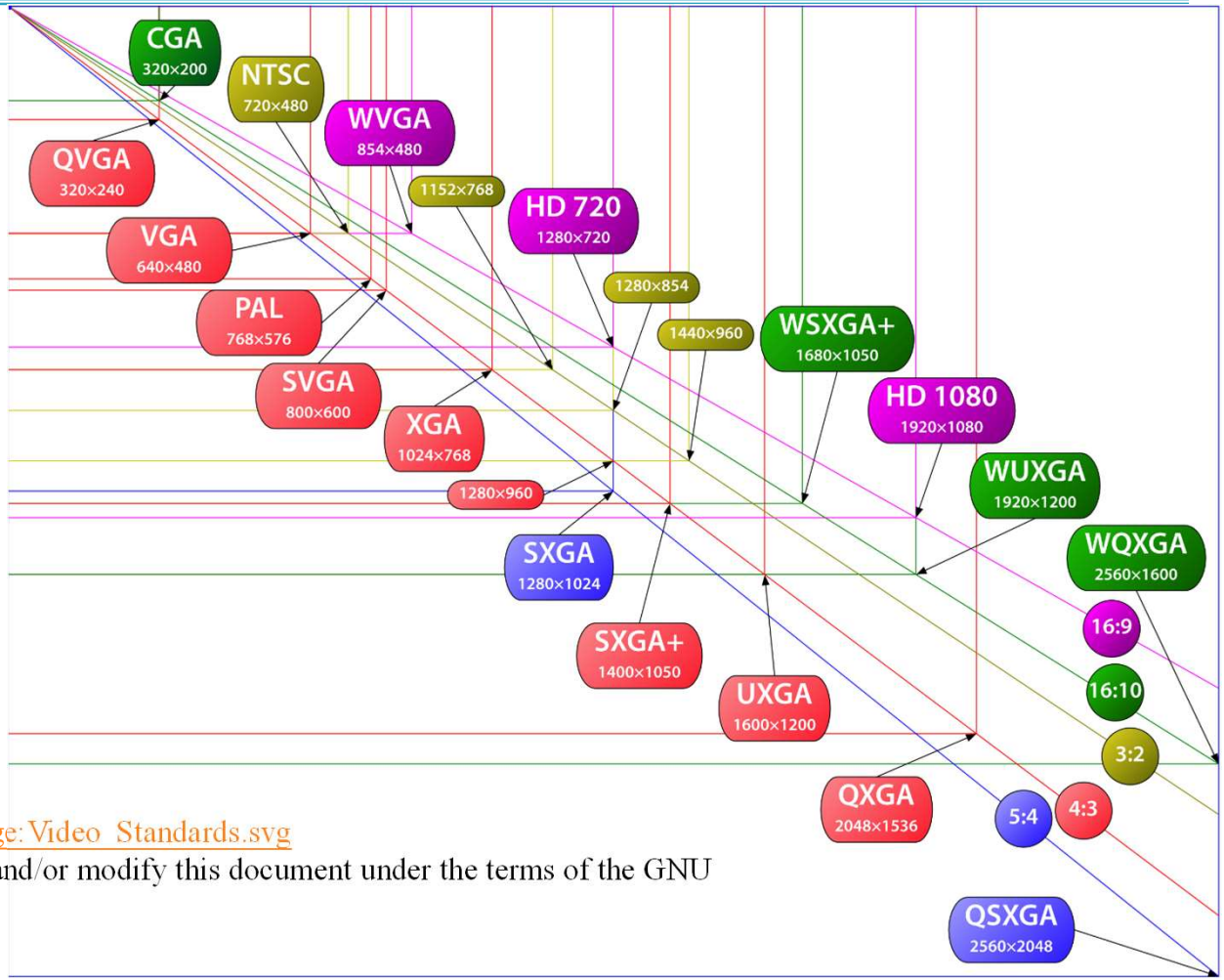
Device Resolution

- How accurately is the device/system approximating the image?

- Printer, Scanner:
  - **Relative value in „dots per unit length“, often „dots per inch“ (dpi)**
  - Desktop Printer 600dpi, Typesetter 1270dpi, Scanner 300-3600dpi, …

- Television, Monitors, Digital cameras:
  - **Absolute value in Pixel = Pixel Dimension (=Monitor Resolution)**
  - PAL TV 768 x 576 px, 17" LCD Monitor 1024 x 768 px, …
  - The resulting dpi depend on resolution and size of the device

Overview: Pixel Dimension



Source: http://de.wikipedia.org/wiki/Image:Video_Standards.svg
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License

Image Resolution

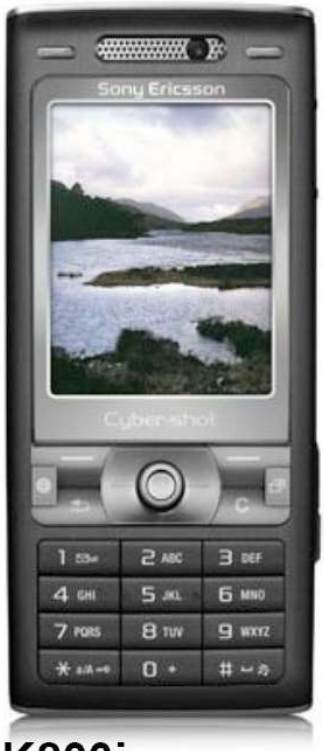
- The image is basically a 2D Array of pixels
- The array has a size in pixels but no physical dimension
- It follows:
  - The physical dimension and the pixel dimension (e.g., 1280x1024) let one calculate the Image Resolution in **pixels per inch (ppi)**

$$\textbf{\textit{Image Resolution}} = \frac{\textbf{\textit{Pixel dimension}}}{\textbf{\textit{Physical Dimension}}}$$

- Ex: 19“ monitor with resolution of 7 Megapixels:

	Pixel	mm	Pixel pro mm	Inch	„ppi“
Width	1280	376	3,4	14,8	86,47
Height	1024	301	3,4	11,85	86,41

Example: Resolution I

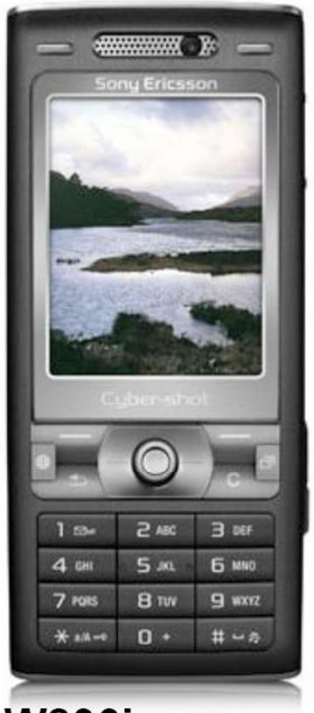


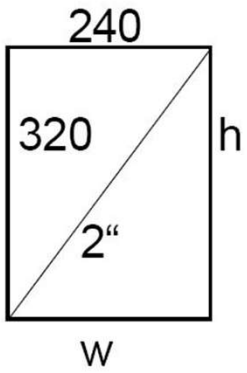
K800i
240x320 Pixel (200 ppi)
2“ Diagonale
262144 Farben

iPhone
320 x 480 Pixel 3.5“
(ca. 160 ppi)
16 Millionen Farben

© Prof. Dr. Kai Uwe Barthel, FHTW Berlin

Example: Resolution II





W800i
240x320 Pixel 2“ Diagonale
262144 Farben

Width w:

$$w^2 + h^2 = 2^2$$

$$w^2 + \left(\frac{4}{3}w\right)^2 = 4$$

$$w^2 \cdot \left(1 + \frac{16}{9}\right) = 4$$

$$w^2 \cdot \left(\frac{25}{9}\right) = 4$$

$$w^2 = \frac{9 \cdot 4}{25}$$

$$w = \frac{6}{5}\ inch$$

Resolution:

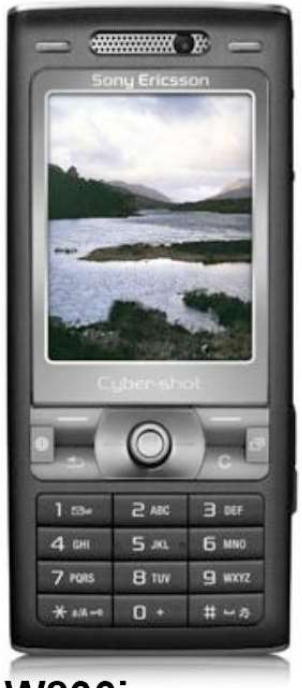
$$\frac{6}{5}\ inch \ \hat{=}\ 240\ Pixel$$

$$1\ inch \ \hat{=}\ 200\ Pixel$$

$$\Rightarrow 200\ ppi$$

© Prof. Dr. Kai Uwe Barthel, FHTW Berlin

Example: Resolution III



W800i
240x320 Pixel
262144 Farben

Bit depth:

$$2^b = 262144$$

$$\log_2\left(2^b\right) = \log_2\left(262144\right)$$

$$b = 18$$

$$\Rightarrow \; 18/3 \; = \; 6 \text{ Bits per channel}$$

Memory requirements:

$$240 \cdot 320 \cdot 18 \; bit$$

$$= 1382400 \text{ Bits} \; \text{->} \; 172800 \text{ Bytes}$$

© Prof. Dr. Kai Uwe Barthel, FHTW Berlin

Resolution change

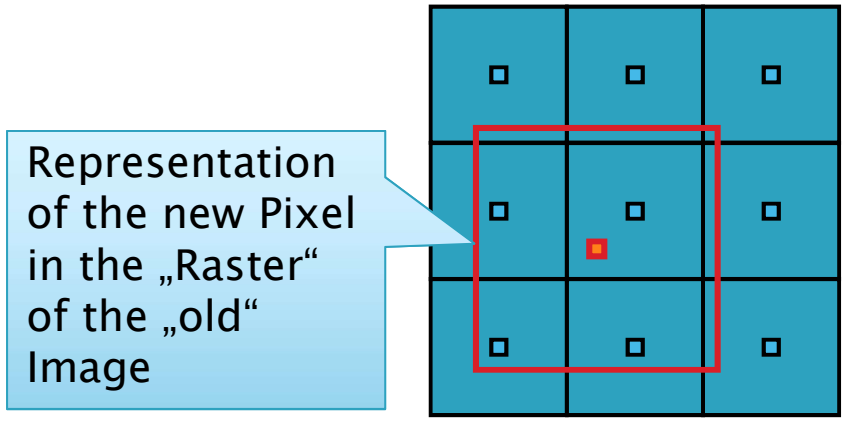
- If Image Resolution < Device Resolution: Interpolation required
  - Leads to loss of quality
  - Example → next slides

- If Image Resolution > Device Resolution: „Downsampling“ required
  - Subjective quality partially better than Device Resolution, because more information is available
  - The technique through which the resolution (finer sampling) of an image is set higher than required by the display device is called „Oversampling“
  - Oversampling is useful if good algorithms are used for Downsampling

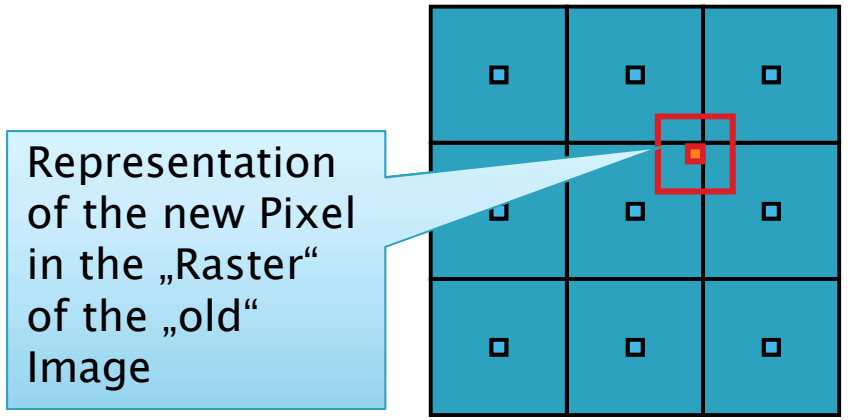
Example: Scaling

- We want to scale up an image to a factor s
- This amounts to compute P‘(x, y) from P(x/s, y/s)
- Problem: in general x/s and y/s are not integers
- Therefore: Interpolation required

Pixels become „bigger“



Pixels become „smaller“



Different Types of Interpolation

- **Nearest neighbour**
  - The new Pixel P‘(x,y) gets the color of the original Pixel which center is the closest to x/s, y/s
- **Bilinear Interpolation**
  - Use the color values of the „covered“ pixels weighted by the size of the respective intersection
- **Bicubic Interpolation**
  - Preform no linear Interpolation, use cubic Splines (similar to Bezier Curves) instead for interpolating

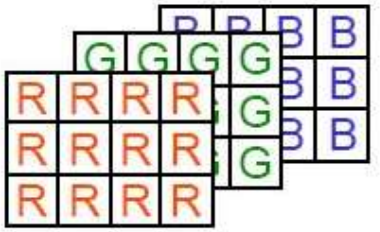
Height = 50 Pixel each (enlarged here)

Table of Contents
Image

Image Organization
Pixel arrangement

- Organization of the data in the file or storage system (after transmission)

Example: image of 3 lines of 4 pixels



Pixel Interleaved:

RGBRGBRGBRGBRGBRGBRGBRGBRGBRGBRGBRGBRGB

Line Interleaved:

RRRRGGGGBBBBRRRRGGGGBBBBRRRRGGGGBBBB

Plane Interleaved:

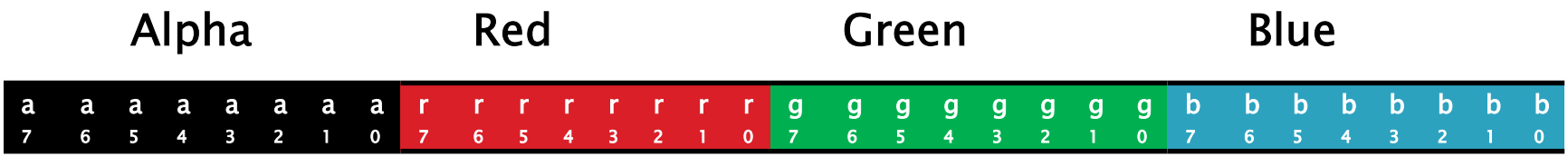
RRRRRRRRRRRRGGGGGGGGGGGGBBBBBBBBBBBB

© Prof. Dr. Kai Uwe Barthel, FHTW Berlin

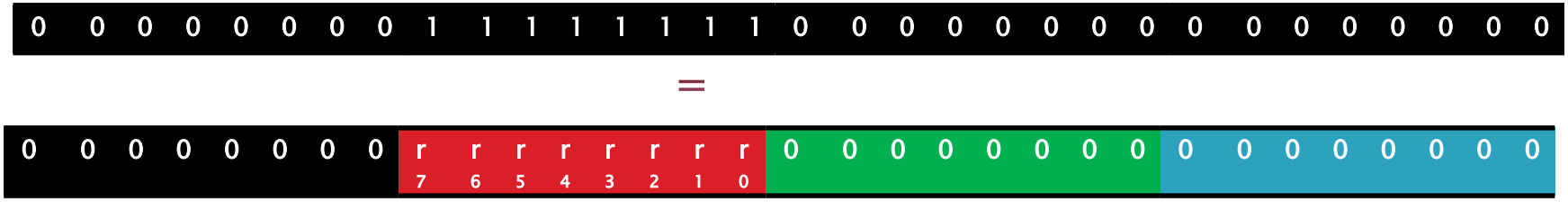
Image Organization

Pixel Representation as 32Bit Integer

Bit distribution:



Access through bit operation: &



Followed by a shift of 16 Bits towards right: >>16

```
// in Java
int c ...; // Pixel
int r = (c & 0x00ff0000 ) >> 16;
```

Table of Contents
Image

Color reduction / Color quantization

- Color Resolution
  - Trade-off between quantization error and storage costs
  - Dependent on the representation capabilities of a device

- Different Approaches
  - Direct assignement (e.g. 32bit in some Color Space)
  - Color Lookup Table
    - Dithering

Quantization of grey-levels images

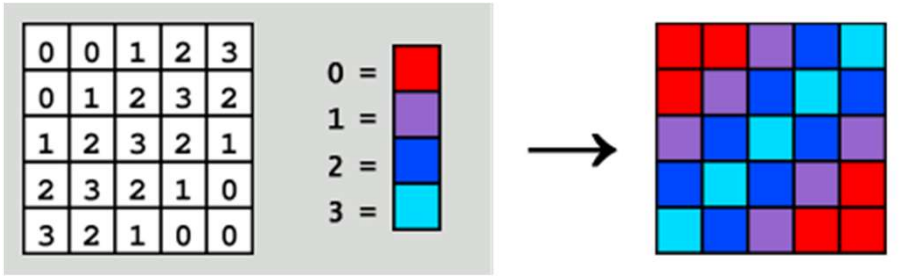
8 Bit **4 Bit** **2 Bit** **1 Bit**

Color depth – Number of Bits pro Pixel (bpp)

Bits	number of colors		Name
1	2	(black + white)	bitonal
8	256	black to white	grey level
8	256	256 colors	Palette
16	65536	black to white	16 Bit grey levels
16	65536	almost all colors	High Color
24	16,7 Mio	all colors	24 Bit TrueColor
32	16,7 Mio	all colors	24 Bit TrueColor + Alpha
48	281 Bio	even more colors	48 Bit TrueColor

Indexed colors / color tables

- 8-Bit color only provide 256 colors
- Instead of storing (r,g,b) for each pixel, only the Index of the pixel in a color table is kept „colour lookup table“ (CLUT)
  - The index has thus a small value (normally stored on one byte)
  - The CLUT (Palette) contains up to 256 (for 1-Byte Index) 24-bit values (often directly implemented in hardware (graphic card))
- The 24-Bit (r,g,b) value stored in the table is used to represent the color of a pixel



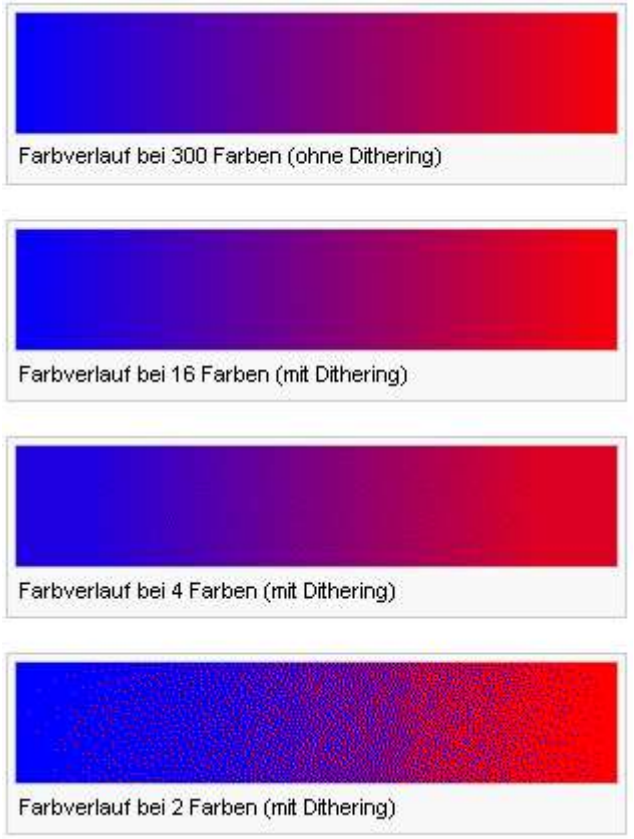
CLUT

Index = (R,G,B)

Image Source Wikipedia: https://en.wikipedia.org/wiki/Indexed_color

Color tables

- CLUT ideally contains the 256 most important colors of the image
- When an image gets reduced from 16.777.216 colors to the 256 most important ones, normally not all colors are found in the table
- Replacing the missing colors by the closest ones in the CLUT may cause a „posterization“ (Banding) effect
- Dithering: uses a pixel pattern and an optical shuffling
- Web-secure colors: 216 colors, known to be reproducable on all platforms by all browsers



Quelle: http://de.wikipedia.org/wiki/Dithering_(Bildbearbeitung)

Dithering

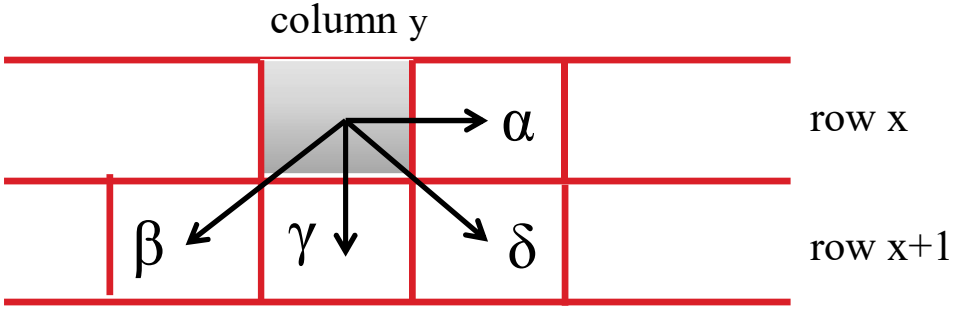
- Apparent increase of the number of perceivable colors (or levels) through spatial displacement

- Types of Dithering
  - Noise Dithering
  - Pattern Dithering
  - Error Diffusion (Floyd-Steinberg Algorithm)

Floyd-Steinberg Dithering I

- Published in 1976, widely used
- Distributes the quantization errors on neighbour pixels
  - The error is usually dispersed on the right and bottom pixels
  - Distribution: $\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$

$$\alpha + \beta + \gamma + \delta = 1.0$$



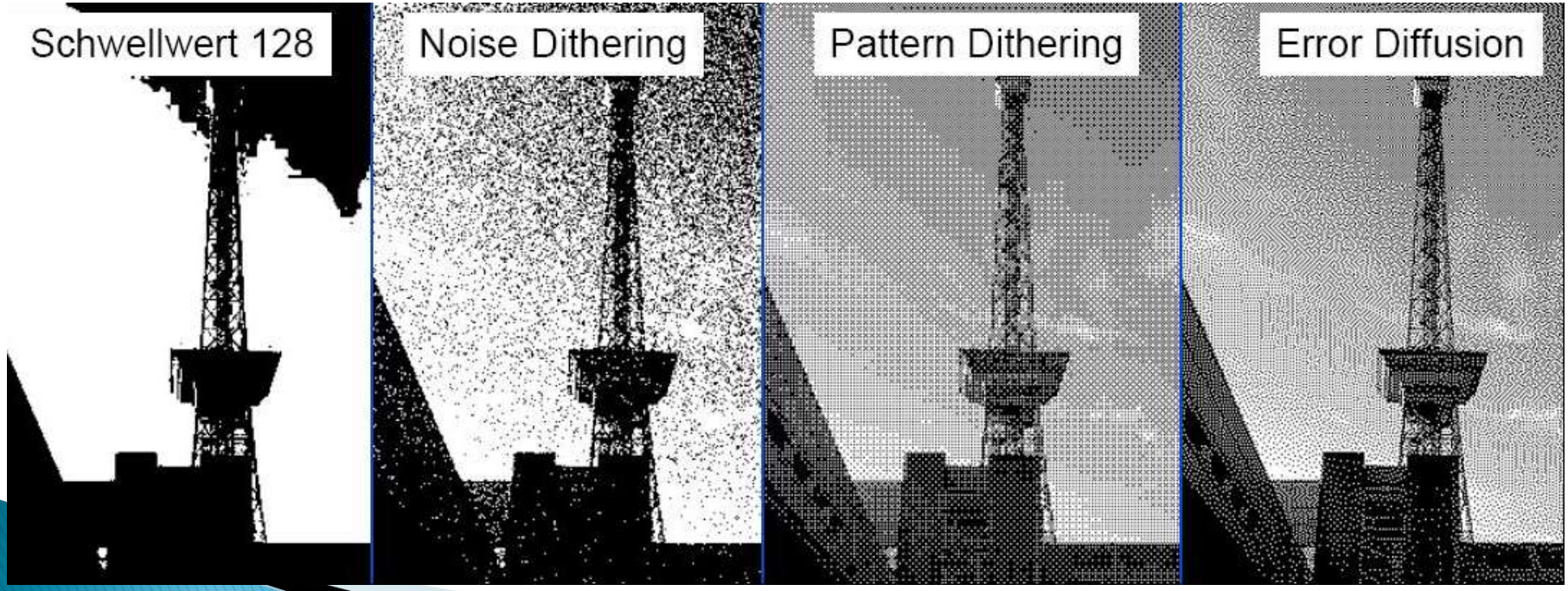
Floyd Steinberg Dithering II

```
for (x=0; x<width; x++) {
  for (y=0; y<height; y++) {

    int oldpixel = pixels[x][y];
    int newpixel = closestColor(oldpixel);
    int error = oldpixel - newpixel;
    pixels[x][y] = newpixel;

    pixels[x][y+1] += alpha*error;
    pixels[x+1][y-1] += beta*error;
    pixels[x+1][y] += gamma*error;
    pixels[x+1][y+1] += delta*error;
  }
}
```

Dithering
Example I



Dithering
Example II

Original

20 colors

20 colors with Pattern dithering

Table of Contents
Image

Image Formats

Choice of usual image formats

- JPG: JPEG File Format (1992)
- TIFF: Tagged Image File Format (~1985)
- BMP: MS Windows Bitmap
- RLE: Run-length encoded BMP
- GIF: Graphics Interchange Format (1987/9)
- TGA: Targa Image File (1984)
- PNG: Portable Network Graphics (1996)
- PBM, PGM, PPM: Portable Bit-/Grey-/Pix-Map (~1985)
- RAW: Raw data format (depends on producer)
- PSD: (Adobe) Photoshop Document
- JP2, JPX, JPM: JPEG2000 File formats (2002)
- ...

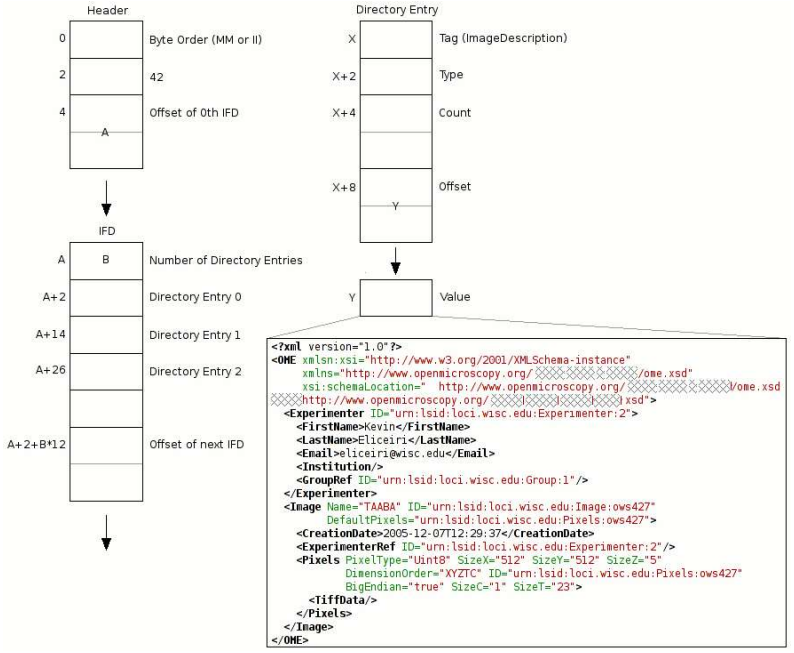
Properties of Image Formats

- Basic image parameters
- Metadata
- Functionalities / Interaction possibilities / Supported image types
- Type and principle of compression
- Support by browsers/software etc.
- Java Support
- Popularity / Interoperability

TIF-Format (Tagged Image File Format)

Developer: Aldus Corporation, Seattle, USA, 1986

- Very flexible, high usage, possibility of compressed storage
- Tiff-File: composed of three main units
  - File header:
    - First 8 Bytes.
    - 2 Bytes: Number representation (big or little endian),
    - 2 Bytes: TIFF-Version number
    - File-Offset-pointer to the first IFD-Block.
  - Image-File-Directory (e.g. one for each Resolution): 3 components:
    - Number of Tag-Entries
    - Tag Array (Array of 12 Bytes)
    - File-Offset-pointer to next IFD, or 0.
  - Tags:
    - Each Directory-Entry is called Tag; it consists of 12 Bytes. It contains registration data and actual image information.
      - 2 Bytes: Tag-ID (ca. 80 different ones, sorted in ascending order)
      - 2 Bytes: Type (SHORT, LONG, FLOAT, ASCII, BYTE)
      - 4 Bytes: Length of data area
      - 4 Bytes: values, if length of data area ≤ 4 (usually for metadata), or offset to data area



TIFF-data tags

- Main tags:

Description	Tag	Data types	Nb. values	Value
ImageWidth	100	short/long	1	...
ImageHeight	101	short/long	1	...
Bits per Pixel	102	short	1	...
Compression (1)	103	short	...	...
Color coding (2)	104	short	...	...
Origin (3)	112	short	...	...
Pointer to RGB color table	140	long	...	...
Information on used JPEG-coding	200 bis 209	short/long	...	...

(1) options: no compression, CCITT-Formats (b/w Images), LZW (Lempel-Ziv-Welch), JPEG, Huffman Coding, etc.

(2) options: WhiteIsZero, BlackIsZero, b/w, RGB, RGB color table, CMYK (Cyan-Magenta-Yellow-Black), etc.

(3) Image origin: up left, down right, down left, etc.

GIF-Format (Graphics Interchange Format)

- Developer: Compuserve Inc., Columbus, Ohio, USA, 1987
  - Several images may be stored in a single file
  - Goal: enable image exchange in a platform independent way
  - Characterized by „lossless“, efficient compression
  - Spezification in 2 incompatible versions
- Parts of a GIF image:
  - Header:
    - Contains the GIF identification sequence and the version number of the used algorithm.
    - The end of the header indicates the beginning of the data area.
  - Application:
    - Enables encoding version and name information of the software used to produce or modify the image.

GIF-Format (Graphics Interchange Format)

- Trailer:
  - Marks the end of the GIF data streams.
- Control:
  - Controls the representation of the following image-block.
  - It is possible to specify animation delay as well as a transparent background color.
- Image:
  - Consists of an Image Header, an optional color table and of pixel data.
- Comment:
  - Textual comment about an image block.
- Plain Text:
  - Enables the ASCII-based coding of texts inside an image (font, color, size, position, stretching and direction or text).

Repeated for each image

GIF-Format (Graphics Interchange Format)

- An image consists of
  - Logical Screen Descriptor
    - For coding the size, position and type of color table of the image,
  - optional global or local color tables
    - The global color table is used for all the following images except if they have an own local color table
  - The pixel colors, noted as pointers to the color table
- The pixel values are compressed by LZW (Lempel, Ziv, Welch)
  - Identifies repeating bit patterns of variable length in the data area → LZW-Analysis
    - Fills a table thanks to which the repeating patterns are replaced by shorter bit sequences (id of the sequence in the table)
    - Frequent Bit patterns are replaced by shorter bit sequences
- GIF is only defined for Image Formats using an 8-bit-Color table

  ⇒Cannot fulfill specific requirements on image quality

GIF – Summary

- Remarks:
  - Developed for fast transmission of Images under (then) low transmission rates.
  - Image dimension restricted to 16 000 x 16 000 pixels
  - Number of colors restricted to 256
  - The maximal color depth is thus 8 bits.
- Image coding:
  - LZW
    compression rate: usually reaches 50 % (but 0 % is also possible).
- Summary:
  - Very popular, e.g. for inline-transmission of images on the Web
  - Cannot deal with 24-Bit Images (True-Color)

Portable Network Graphics (PNG)

- Graphics format for raster graphics
- Proposed as free and less complex than TIFF replacement for GIF, which was encumbered by patents until 2004
- Lossless data compression
- The compression may be improved using PNG prefilters:

Number	Name	Description
0	None	No prefiltering. Original pixels used
1	Sub	The differences to the next left neighbour are used
2	Up	The differences to the next top neighbour are used
3	Average	The differences to the average of the next top and left neightbours are used
4	Paeth	A so-called Paeth-Predictor, which uses differences to the next top left and diagonal top left neighbours, is applied.

Portable Network Graphics

- Pre-filters improve the compressibility of the data
- After pre-filtering: lossless deflate-Algorithm
  - Also used in
    - ZIP-Archive format (developed for it)
    - The gz-format of the archiving program gzip
    - The image format TIFF
- Deflate is currently the only supported method.
- The standard leaves however room for extensions

- The pre-filters enable a lower data size compared to GIF-files

Which format for which usage?

Format	Storage req.	Application	Remark
JPEG	Low to average	P, D, W	Lossless compression
TIFF	hoch	M, I, P, D	Lossless compression possible
GIF	Average	M, I, W	Indexed colors, possible high loss of tones
PNG	Average (or low)	M, I, W, (P)	Lossless compression possible

Anwendungsbereiche: M = Monochrome images,
I = Illustrations,
P = Photos,
W = Images on WWW,
D = Data exchange with graphic companies

[But: to be taken with a pinch of salt!]

The End