Introduction to Media Informatics: Images

Dr. Mathias Lux
Alpen-Adria Universität Klagenfurt

Contents

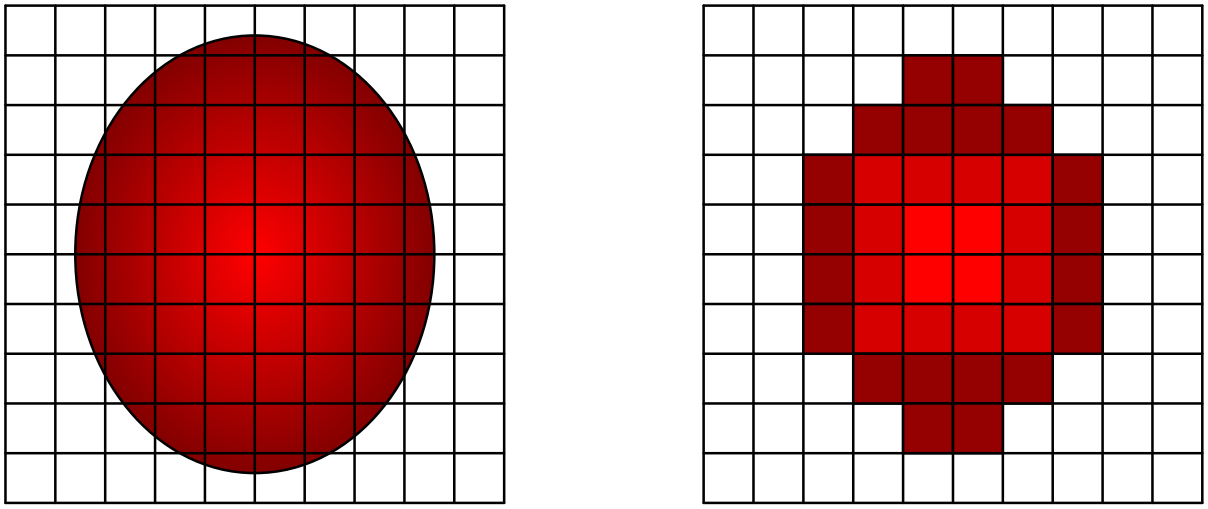
- Color Spaces
- Compression
- Formats
- Filtering

What are (digital) images?

- An Image is
  - Created by a set of photons
    - With different frequency
    - Moving from different sources
    - Along different vectors
  - A representation of sensor unit activation
    - Activated by the set of photons
- Storing an image
  - Based on the set of photons ???

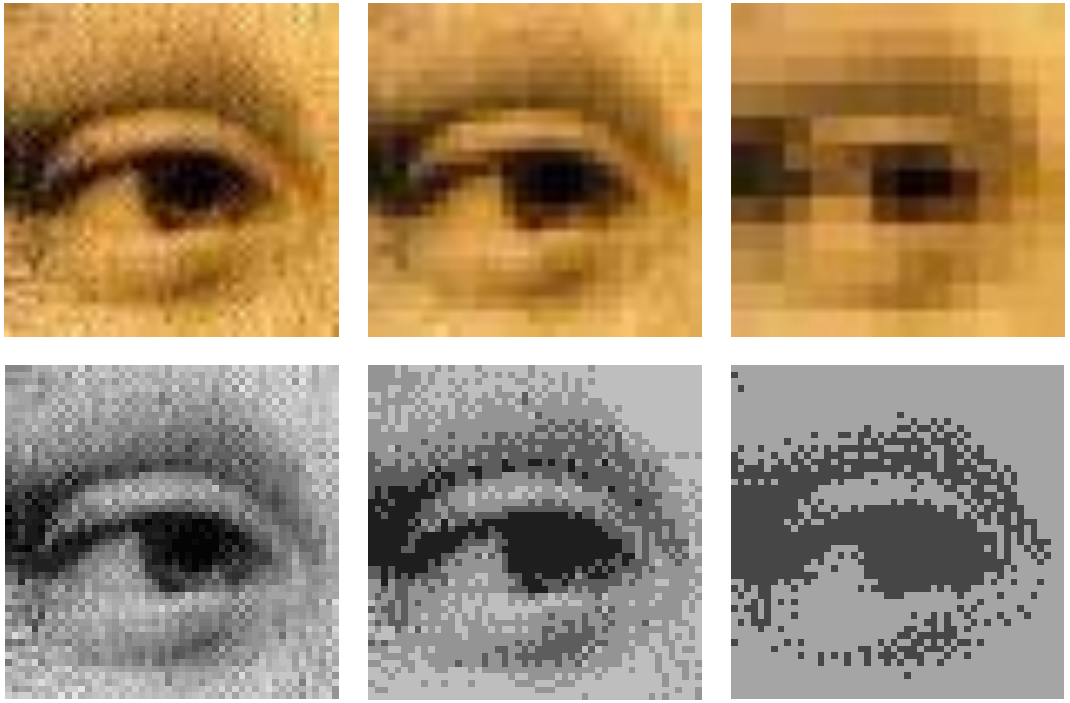
Sampling & Quantization

- Capturing continuous images on sensors
  - Sampling: Continuous to matrix
  - Quantization: Continuous color to value



Sampling & Quantization

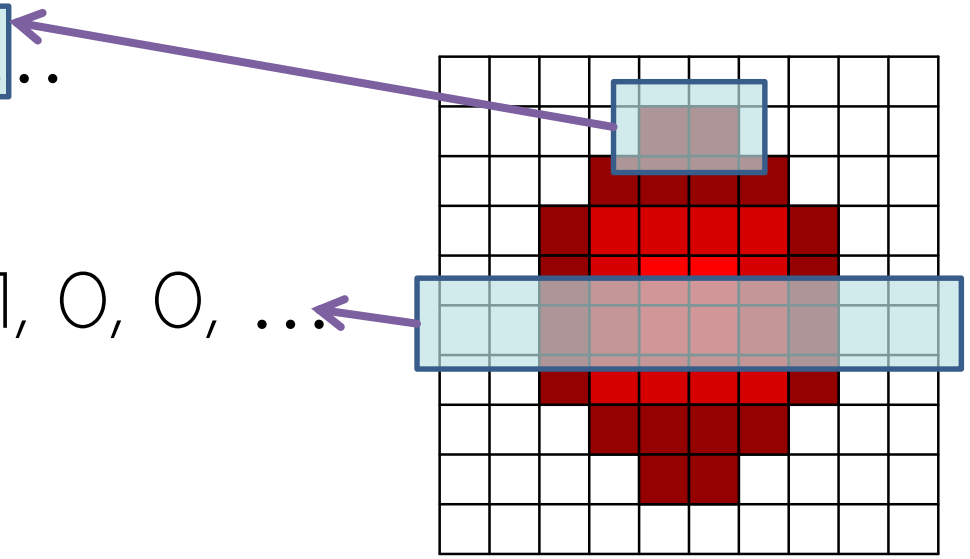
- Size of a captured image:
  - # of samples (width*heigth) * # of colors



Storing digital images

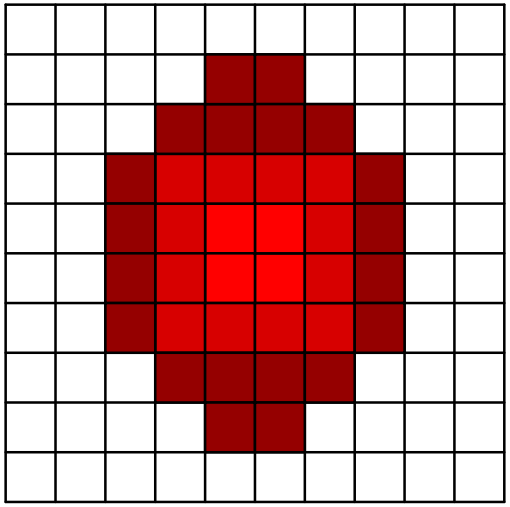
- Defining color per cell
- Left to right & top to bottom

- 0, 0, 0, … 0, 1, 1, 0 …
- … 0, 0, 1, 2, 3, 3, 2, 1, 0, 0, …



Storing digital images

- Header: Additional information
  - Width and height: 10x10
  - DPI, etc.
  - Quantization
  - Color space or color table
    - 0 -> white
    - 1 -> dark red
    - 2 -> medium red
    - 3 -> light red



Color Models

- How to „specify“ color?
- Color models are an abstraction
  - mapping color <-> numbers
- Color palettes are an enumeration

Color palettes

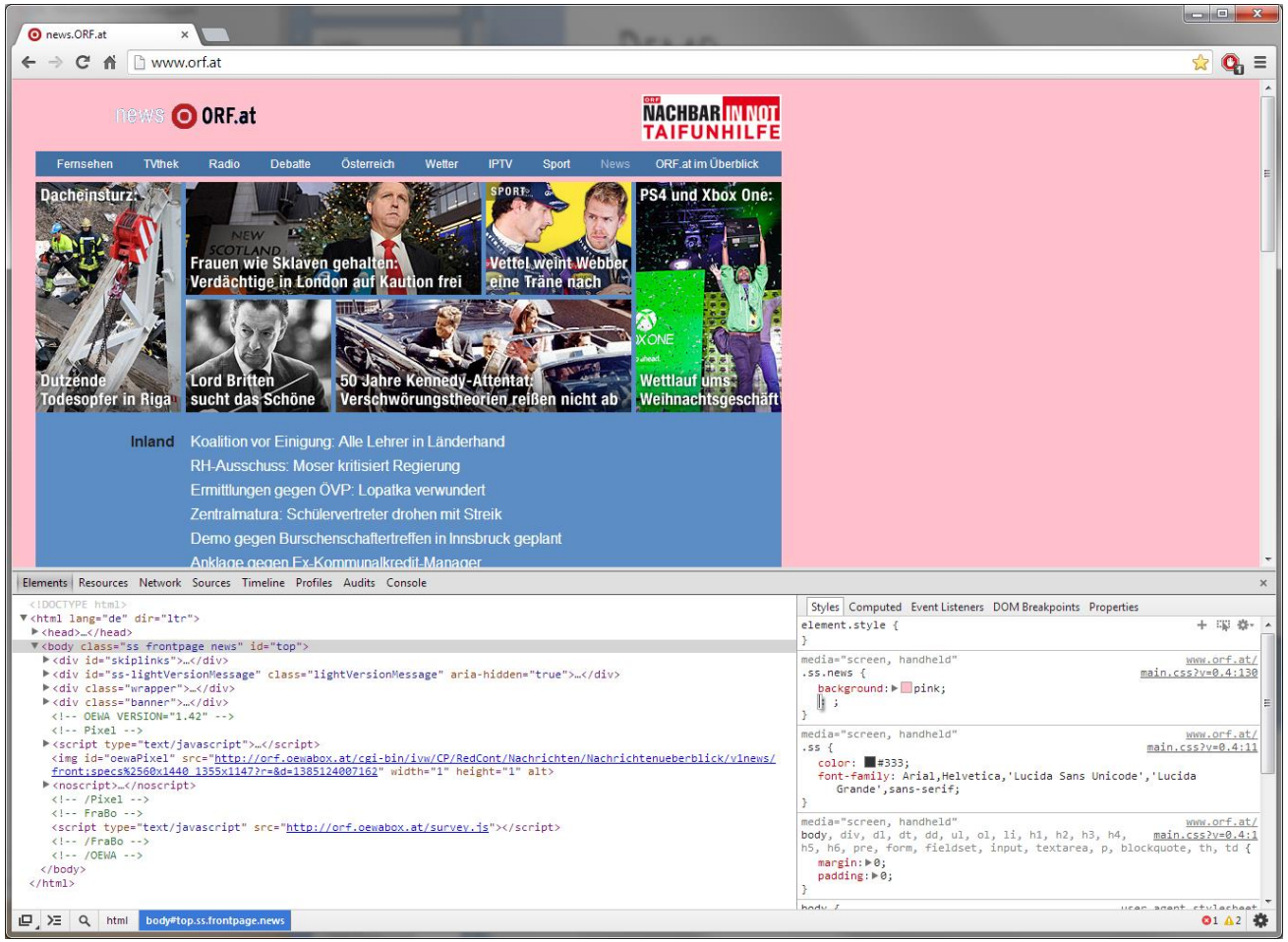
(HTML 4.01 spec.)

CSS1 / HTML3–4 / VGA color names

Name	Hex triplet	Red	Green	Blue	Hue	Satur	Light	Satur	Value	
White	#FFFFFF	100%	100%	100%	0°	0%	100%	0%	100%	
Silver	#C0C0C0	75%	75%	75%	0°	0%	75%	0%	75%	
Gray	#808080	50%	50%	50%	0°	0%	50%	0%	50%	
Black	#000000	0%	0%	0%	0°	0%	0%	0%	0%	
Red	#FF0000	100%	0%	0%	0°	100%	50%	100%	100%	
Maroon	#800000	50%	0%	0%	0°	100%	25%	100%	50%	
Yellow	#FFFF00	100%	100%	0%	60°	100%	50%	100%	100%	
Olive	#808000	50%	50%	0%	60°	100%	25%	100%	50%	
Lime	#00FF00	0%	100%	0%	120°	100%	50%	100%	100%	green
Green	#008000	0%	50%	0%	120°	100%	25%	100%	50%	
Aqua	#00FFFF	0%	100%	100%	180°	100%	50%	100%	100%	cyan
Teal	#008080	0%	50%	50%	180°	100%	25%	100%	50%	
Blue	#0000FF	0%	0%	100%	240°	100%	50%	100%	100%	
Navy	#000080	0%	0%	50%	240°	100%	25%	100%	50%	
Fuchsia	#FF00FF	100%	0%	100%	300°	100%	50%	100%	100%	magenta
Purple	#800080	50%	0%	50%	300°	100%	25%	100%	50%	

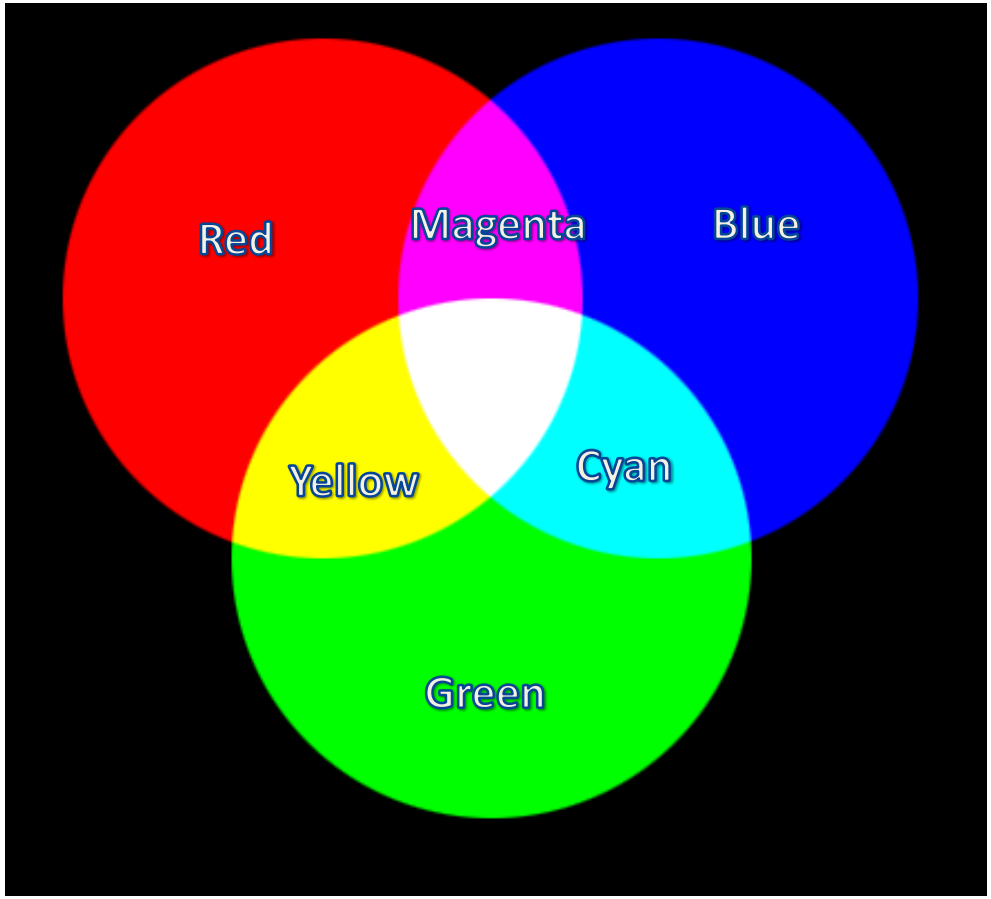
Src. Wikipedia

Demo


news.ORF.at
www.orf.at
news ORF.at
NACHBAR IN NOT
TAIFUNHILFE
Fernsehen
TVthek
Radio
Debatte
Österreich
Wetter
IPTV
Sport
News
ORF.at im Überblick
Dacheinsturz:
Dutzende Todesopfer in Riga
Frauen wie Sklaven gehalten: Verdächtige in London auf Kaution frei
Vettel weint Webber eine Träne nach
Lord Britten sucht das Schöne
50 Jahre Kennedy-Attentat: Verschwörungstheorien reißen nicht ab
PS4 und Xbox One:
Wettlauf ums Weihnachtsgeschäft
Inland
Koalition vor Einigung: Alle Lehrer in Länderhand
RH-Ausschuss: Moser kritisiert Regierung
Ermittlungen gegen ÖVP: Lopatka verwundert
Zentralmatura: Schülervertreter drohen mit Streik
Demo gegen Burschenschaftertreffen in Innsbruck geplant
Anklage gegen Ex-Kommunalkredit-Manager
Elements
Resources
Network
Sources
Timeline
Profiles
Audits
Console
Styles
Computed
Event Listeners
DOM Breakpoints
Properties
html
body#top.ss.frontpage.news

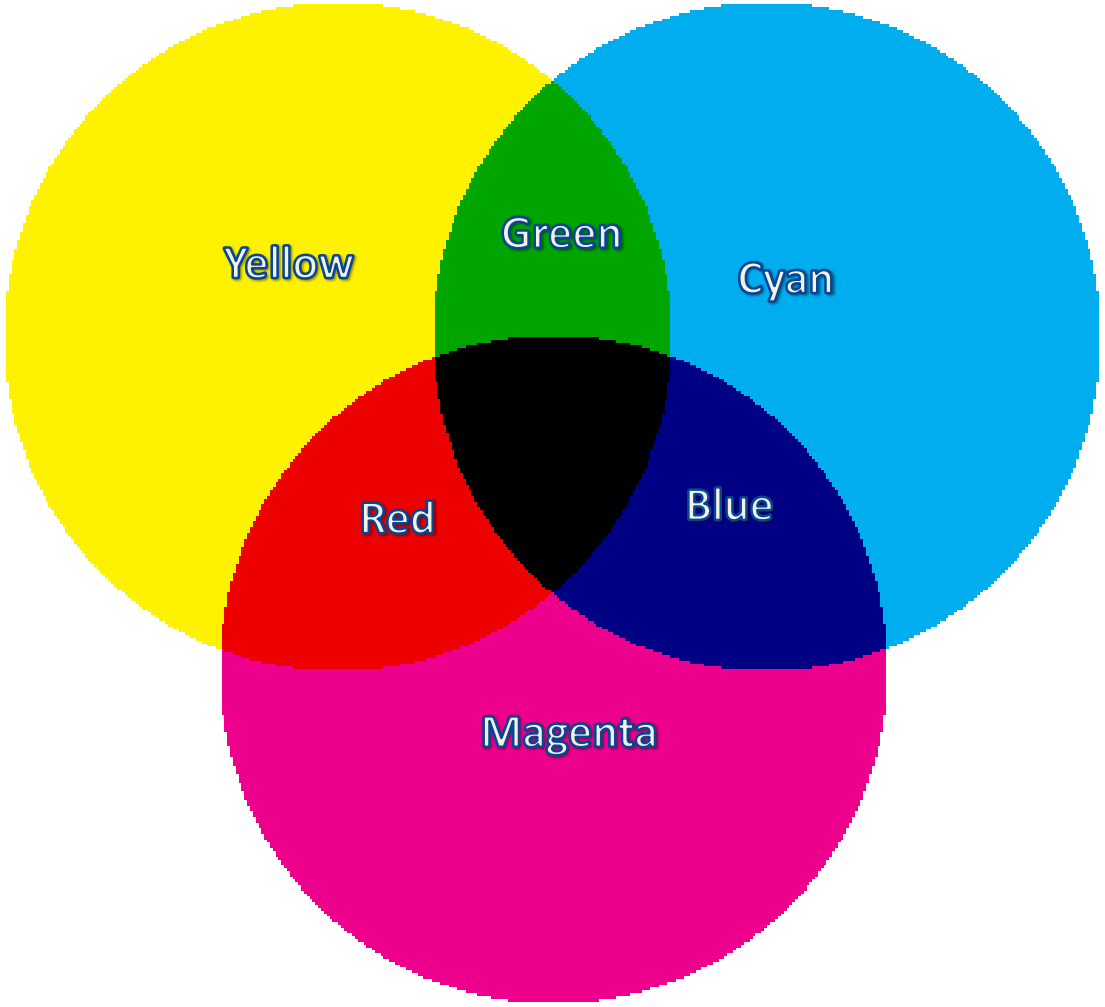
Color: Additive Scheme

(Mixtures of Light)



Color: Subtractive Scheme

(Mixtures of Pigments)

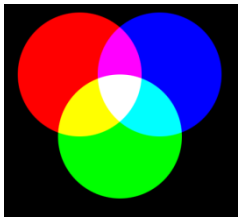


CIE 1931 COLOR SPACE

- First mathematically defined color space
- Tristimulus values: X, Y, Z
  - roughly red, green and blue
  - not observed as such
- Defined based on experiments in the 1920s

RGB Color Model

- Additive color model
- Based on red, green and blue
- Color is defined as mixture of intensities in all 3 channels
- Examples with channel intensities in [0,1]
  - (0, 0, 0) -> no intensities, darkest color
  - (1, 1, 1) -> full intensities, lightest color
  - (1, 0, 0) -> saturated red



RGB Color Model

- Intensities are typically quantized
  - E.g. 8 bit / channel (24 bit per pixel)
  - 16-bit Highcolor: 5 bits / channel (+1 for green)
- With 8 bits per channel
  - range from 0-255 (decimal) or 00-FF (hex)
- Web: 8 bits / channel, defined in hex
  - #000000 -> black
  - #FF0000 -> red

sRGB Color Space

- Standard defined by HP & Microsoft (1996)
- Used for monitors, printer & internet
- Uses 8 bits per channels
- Transformation to and from CIE XYZ defined by:

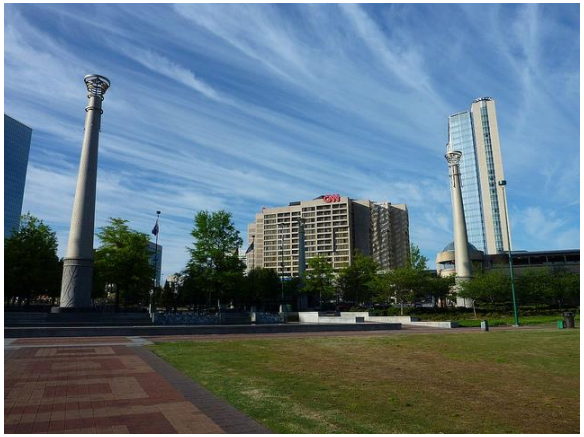
$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Color Space: HSV

- Transformation of RGB to
  - Hue: The actual color (0°-360°)
  - Saturation: The brilliance of the color
  - Value: The lightness of the color
- Useful for changing hue (color) withouth changing lighness (value) or brilliance (saturation)

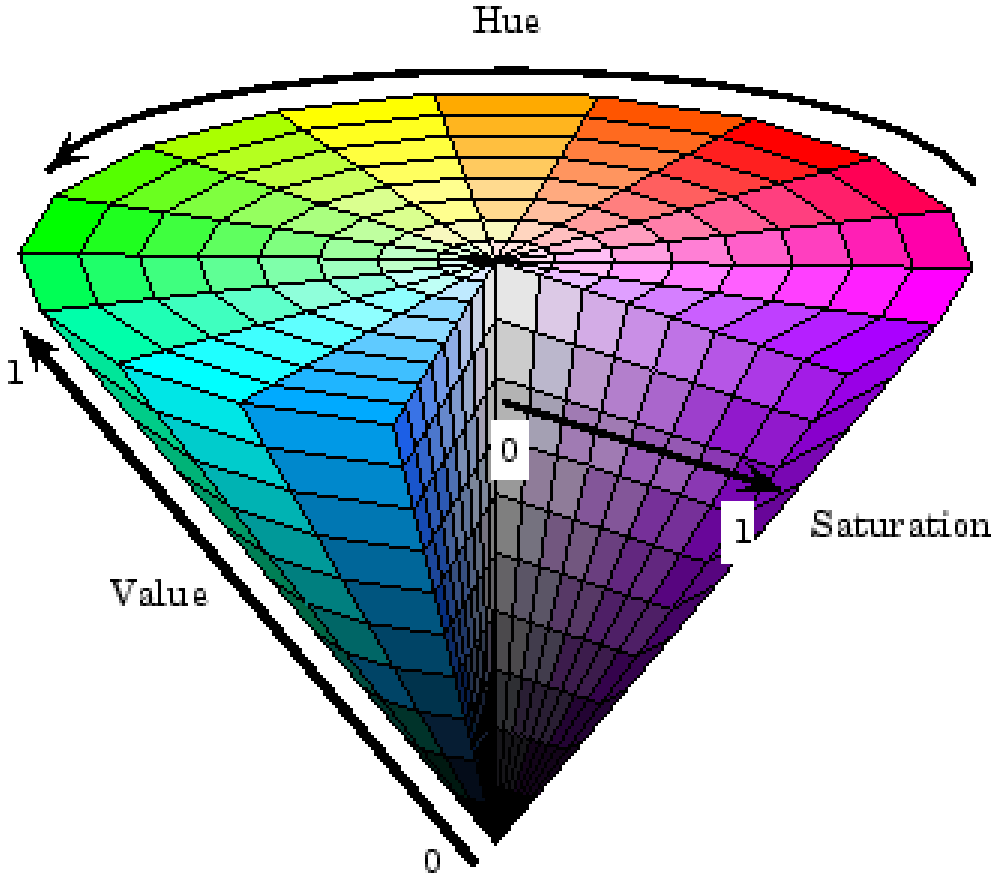
Color Space: HSV

Increase saturation



Set fixed hue

Color Space: HSV



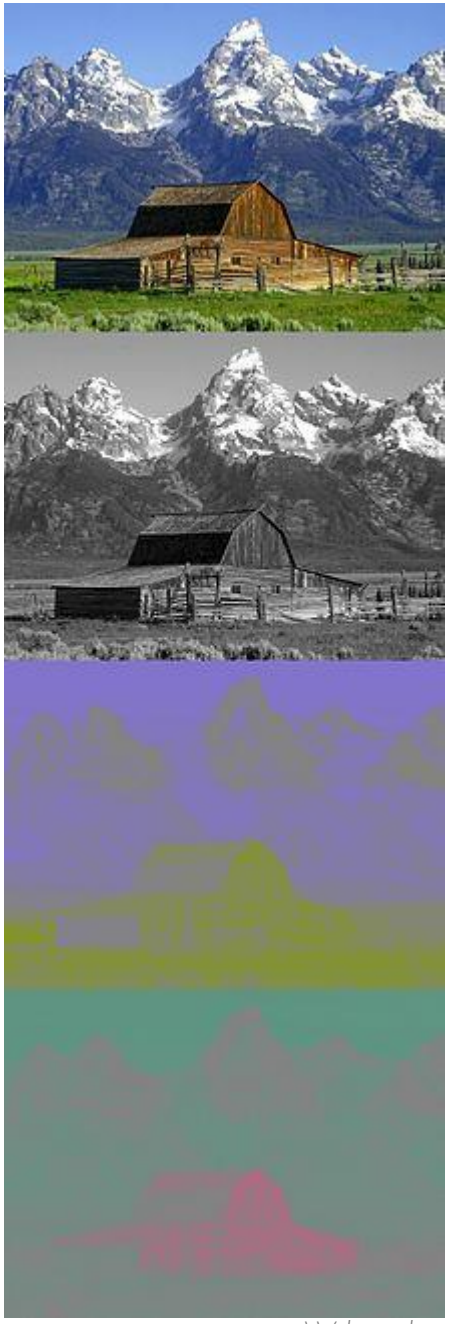
Src. Mathworks.com

Grayscale

- A pixel just has an intensity value
- Intensity channel := Y
- Conversion from RGB
  - Y = 0.2126*R + 0.7152*G + 0.0722*B

YCbCr

- Splits color into luma & chroma
  - Y: intensity
  - Cb: blue-difference
  - Cr: red-difference
- Used for chroma subsampling
  - Reducing "color" in images

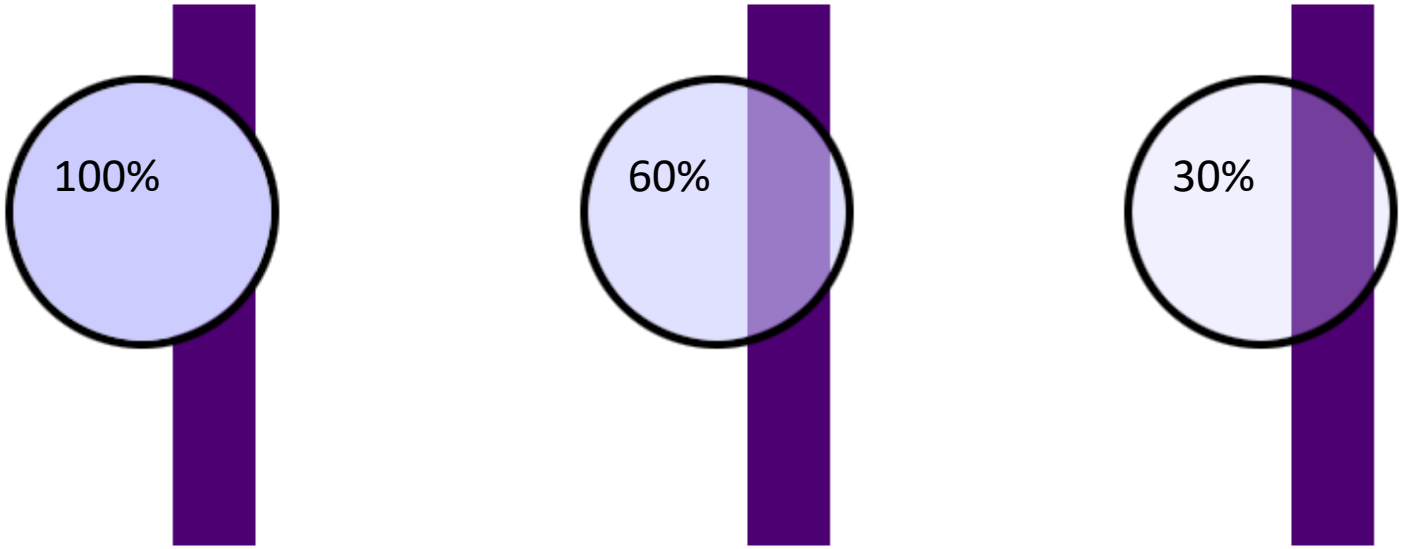


src. Wikipedia

Transparency & Alpha Channel

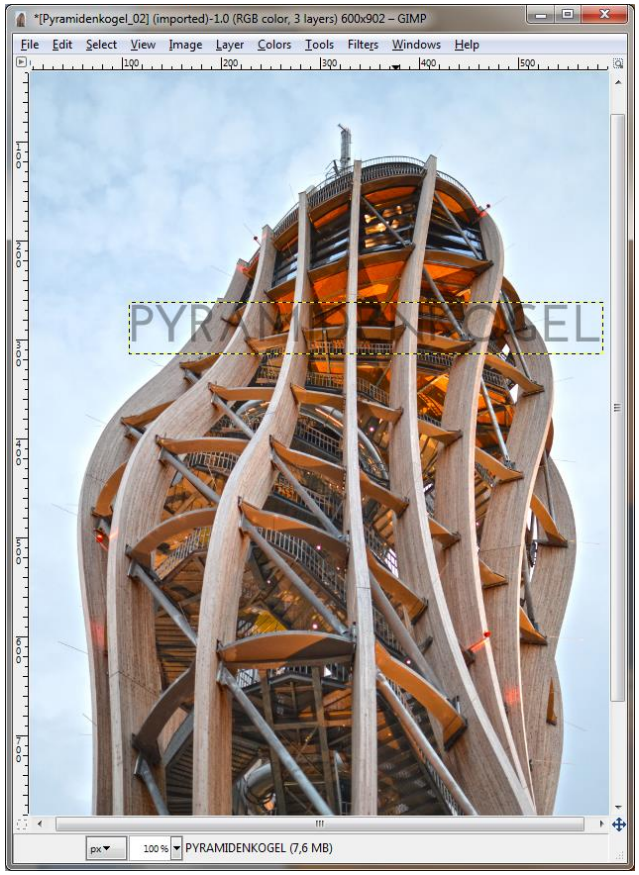
- How to define the opacity of a color / pixel?
- Alpha channel
  - 0% … fully transparent
  - 100% … fully opaque pixel
- Color spaces
  - RGBa – Red, Green, Blue & Alpha
  - ARGB – Alpha first, used in Flash, Silverlight

Transparency & Alpha Channel



Demo: GIMP

- Create a text layer on an image
- Change opacity of the layer



Size of digital images …

- Example: Canon EOS M
  - 333 € (Nov 22, 2013)
  - 18 Mega Pixels

- What's the size of an image?
  - 5,184 x 3,456 pixels
  - 24 bits per pixel for color = 3 bytes
  - Size: 5,184 x 3,456 x 3 bytes = ~ 504.7 MB

Compression: Lossless vs. Lossy

Compression is a tradeoff processing time vs. storage space

- Lossless
  - All information is retained
  - Image can be re-constructed without flaws
- Lossy
  - Unimportant information is dropped
  - Image reconstruction is similar to original image

Lossless vs. Lossy

463 KB -- 3.5 KB

Lossless compression

(examples)

- Run-length encoding
  - Crunching long sequences of same numbers
  - wwwwooowwoooowwww -> 4w3o2w4o4w
- Dictonary encoding (LZW, ZIP, ...)
  - Re-occuring sequences get short codes
- Huffmann
  - Variable length, entropy encoding
  - Numbers with high frequency get short representations

Lossy Compression: DCT

- DCT == „discrete cosine transform"
- Expresses data points as sum of cosine functions
- Defined by following equation:

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right] \qquad 1$$

$$C(u) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{array} \right\} \qquad 2$$

Discrete Cosine Transform

Actual encoding process easier than the equation looks:

1. For each 8x8 pixel block from an image
2. Multiply with DCT matrix
3. Quantize with quantization matrix
4. Encode in zig-zag manner

DCT: Example

- Extract 8x8 pixels block
- Subtract 128 from each entry

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

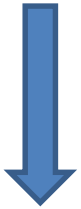
$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

DCT: Example

- Apply DCT with $D = T*M*T^{-1}$

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$



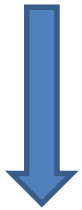
$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

DCT: Example

- Apply quantization matrix

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

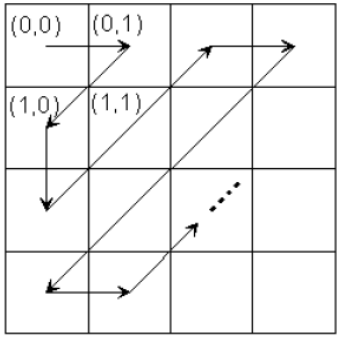


$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right)$$

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

DCT: Example

- Encode in zig-zag manner



$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- 10, 4, 3, -7, 9, 2, 5, 1, -5, -3, -2, ...

Note: There are a lot of zeros at the end!

DCT: Example

Inverse process:

- Multiply component-wise by the same quantization matrix & round
- Perform IDCT: $T^{-1}*M*T$
- Add 128 per component

DCT: Example

- DCT with Q_{50}
- Compressed version to the right

Image Containers & Formats

- Windows Bitmap File Format (BMP)
- Portable Network Graphics (PNG)
- Tagged Image File Format (TIFF)
- JPEG images and file format (JPEG)
- Scalable Vector Graphics (SVG)

Bitmap File Format

- Popular format for storing raster graphic
- Microsoft Windows and former OS/2 platforms
- Also known as Device Independent Bitmap (DIB) format
- Typical file extensions .bmp and .dib, MIME-Type image/x-ms-bmp

- Characteristics:
  - Stores pixel values typically uncompressed or in RLE
  - Color palette mode available
  - Supports RGB color model and greyscale bitmaps
  - Color depth of 1, 4, 8, 16, 24, or 32 bits per pixel
  - Simple format, easy handling
  - Typically used for icons and small pictures

Bitmap File Format

- File organization
  - Bitmap File Header (14 Bytes)
    - Magic Number (0x42 0x4D, ASCII Code for 'BM')
    - Total size of the bitmap file (4 Bytes)
    - Offset to bitmap data in file (4 Bytes)
  - Bitmap Information
    - Structure depends on file format version (e.g. V3, V4, V5)
    - Size ranging from 12 – 124 Bytes
    - Bitmap width and height in pixels (4 Bytes each)
    - Number of bits per pixel (color depth)
    - Optional compression method
    - Number of colors in palette (0 – 2^32)

Bitmap File Format

- File organization (cont'd)
  - Color Palette (optional)
    - Color descriptions using 24 Bit RGB values + 1 Byte Padding
    - Number of palette entries specified in Bitmap Information block
    - Colors referenced in Bitmap Data by index into Color Palette
  - Bitmap Data
    - Pixel values (either color or index into palette) stored row-by-row
    - Unless otherwise specified rows are stored in a bottom-up fashion, i.e. lower left corner to upper right corner
    - First byte of a row has to be word (32-bit) aligned
- Remarks
  - Actual storage depends on color-depth, compression etc.
  - Values are Little-Endian encoded (Intel x86)

Graphics Interchange Format (GIF)

- Developed by CompuServe in the late 80s
  - GIF87a – introduced 1987
  - GIF89a – extension in 1989 (animated GIFs)
- Storage of raster image
  - 8 Bit color palette
  - Each pixel consists of index into palette
- LZW-based compression of image data
  - Paper in 1984, Patent in 1985 in different countries,
    claims by Unisys in 1993,
    Patent expired in 2003 and 2004 (can be used freely)
- Supports
  - multiple images per file
  - transparent color
  - Progressive mode (interlaced image transmitted first)
- File extension .gif, MIME Type image/gif

Portable Network Graphics (PNG)

- Design goals
  - Portable storage of raster-images
  - Lossless compression
  - Patent-free replacement for GIF format
- Development History
  - v1.0 in 1996, RFC 2083 and W3C Recommendation
  - v1.1 in 1998 and v1.2 in 1999 – minor modifications
  - official ISO standard since 2004 - ISO/IEC 15948:2004

Portable Network Graphics (PNG)

- Characteristics
  - indexed-color, greyscale and true-color raster images
  - Supports only RGB color model (no YCbCr or CMYK)
  - Progressive display mode
  - Streamability (files can be read or written serially)
  - Transparency and alpha channel support
  - Checksums for file corruption checking
- Metadata
  - Ancillary information (e.g. Textual descriptions)
  - Gamma correction and color calibration information

PNG – File Structure

- PNG signature
  - File must start with fixed 8-Byte sequence
  - Hex: 0x89 0x50 0x4E 0x47 0x0D 0x0A 0x1A 0x0A
  - ASCII: \211 P N G \r \n \032 \n
- Sequence of chunks
  - First chunk: IHDR, last chunk: IEND
  - Each chunk consists of
    - Length (4 Byte): number of bytes in the chunk data (*n*)
    - Chunk Type (4 Byte): ASCII letters (a-z, A-Z)
    - Chunk Data (*n* Bytes): actual payload of chunk
    - CRC (4 Bytes): checksum over Type and Data field

PNG – Interlaced Mode

- For implementing progressive display of pictures/logos
- Use case
  - Slow internet connections
  - Low-res image is immediately visible
- Implementation – Adam7 interlace mode
  - Seven passes over the pixel of the image
  - In each pass a subset of pixels are encoded
  - Based on a 8x8 pattern replicated over the image data
  - Each pixel assigned to a specific pass

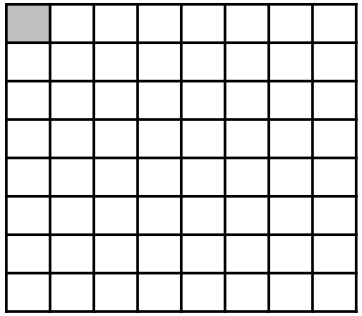
1	6	4	6	2	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7
3	6	4	6	3	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7

Progressive mode ...

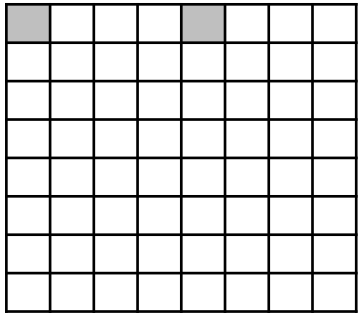
Src. http://msdn.microsoft.com/en-us/library/ee720036(VS.85).aspx

PNG – Interlaced Mode

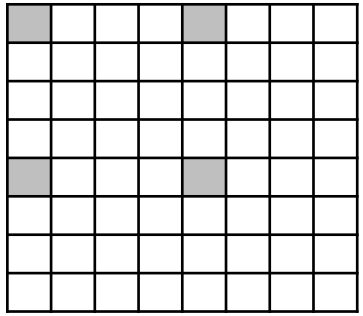
1st pass, 1/64



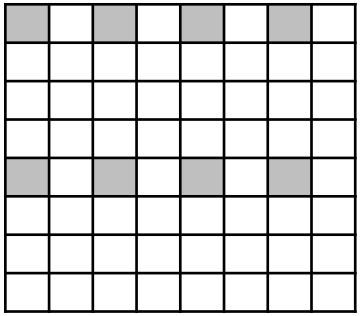
2nd pass, 1/32



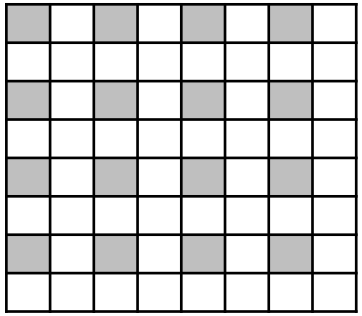
3rd pass, 1/16



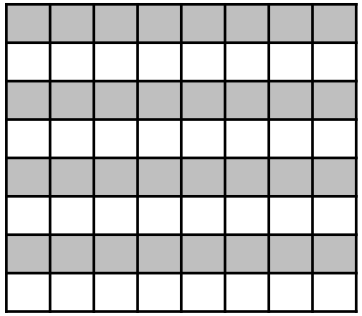
4th pass, 1/8



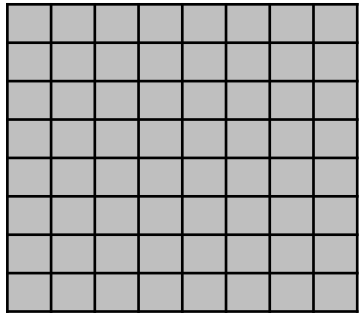
5th pass, 1/4



6th pass, 1/2



7th pass, full



TIFF – Tagged Image File Format

- Initially developed by Aldus (first version in 1986)
- Now part of Adobe Systems Inc.
- Latest revision 6.0 – June 1992,
  since than no further development
- Well-supported by many applications especially in the context of scanners and fax software
- Specification divided into two parts
  - Baseline TIFF
  - TIFF Extensions

TIFF – File Structure

- Each file consists of a fixed-size header and a set of (at least one) Image File Directory (IFD)

- Each file begins with a 8-byte image header
  - Byte 0-1: signals the byte order
    - 0x49 0x49 ("II")…Little-endian encoding
    - 0x4D 0x4D ("MM")…Big-endian encoding
  - Byte 2-3: magic number (0x2A or decimal 42)
  - Byte 4-7: offset (in bytes) of the first IFD

- Image File Directory
  - Represents an image in the file
  - One TIFF file can contain more than one image (e.g. Scanner)

TIFF – TIFF Extensions

- Support for multiple images per file
  - Storage of a scanned document in a single file
  - Storage of a received fax message
- Document Storage and Retrieval
  - Additional TIFF fields for annotations & metadata.
- Compression
  - Enhanced CCIT Bilevel Encodings
  - LZW
  - JPEG Compression support
- Color spaces
  - YCbCr (with chroma subsampling support)
  - CIE L*a*b*
  - RGB Image Colorimetry
  - CMYK (Cyan, Magenta, Yellow, Key) for printing

TIFF – Summary

- Supports a great variety of color spaces
  - RGB, CMYK, YCbCr, Grayscale, …
- Multiple, arbitrary sized images per file
  - Scanned documents, Fax servers
  - Thumbnails
- Compression modes
  - Uncompressed, Lossless & Lossy
- Wide range of color depths
  - 1 Bit B/W up to 48 Bit RGB (e.g., for scientific imaging)
  - Support of Alpha Channels
- Fragmentation - Strips or Tiles can be decoded independently
- File extension .tif or .tiff, MIME-Type image/tiff

JPEG

- Standardized as CCITT Rec.81 in 1992
  - Specification of encoder
  - Specification of decoder
  - Definition of an interchange format (Annex B)
- Applications
  - Good compression of photographs and paintings
  - Not very suitable for drawings, icons or text (sharp edges)
  - Not recommended as intermediate format during image editing
- Coding tools
  - DCT-based lossy compression
  - Prediction-based lossless compression
  - Huffman or arithmetic entropy coding

JPEG – File Format

- JPEG Interchange Format (JIF)
  - Annex B defines the JPEG Interchange Format (JIF)
  - Some shortcomings (definition of color-space etc.)
  - Almost never used in applications nowadays
- JPEG File Interchange Format (JFIF) in 1992
  - Defines YCbCr as default color space
  - Allows for defining sample aspect ratios
  - Clarifies the location of chroma subsamples
  - Thumbnail support
- Wide support in applications as well as devices like digital still cameras, mobile phones etc.

JPEG

- File organization based on marker
  - Can be identified in the file without decoding the image
  - In contrast to chunks (PNG) the marker do not contain length information about the compressed image data
- Variety of compression modes
  - Lossless and DCT-based
  - Sequential, progressive and hierarchical storage
- 8 or 12 Bit sample precision per component
- JFIF supports inclusion of thumbnail
- File extension .jpg or .jpeg, MIME-Type image/jpeg

Image Format Selection ...

Consider following use cases:

- Print: Color Model CMYK.
- Web: Storing a logo without compression artifacts.
- Email: Sending a photo to a friend.
- Image Processing: Store a preprocessed photo for more processing later on.

Spatial Filtering

- Methods for *enhancing* or *transforming* the image
- Typically a *kernel* or *filter* is used:
  - A matrix which is applied to the image
  - In a linear transformation

Spatial Filtering

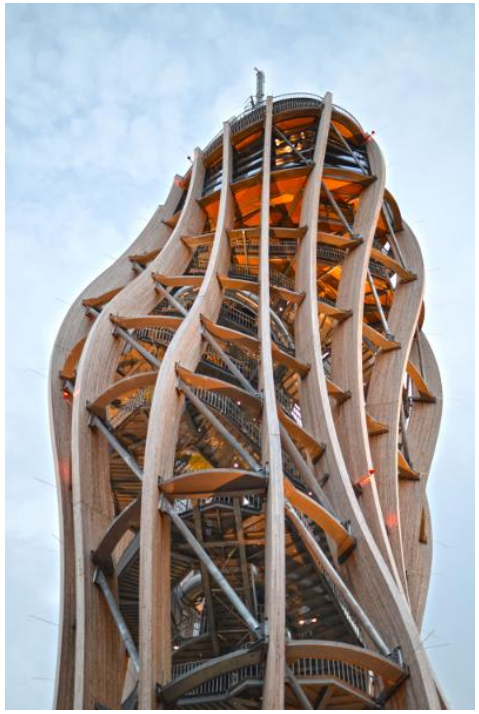
194	128	102	197	69
162	68	103	144	115
121	85	57	27	14
24	183	192	239	150
92	93	154	138	170

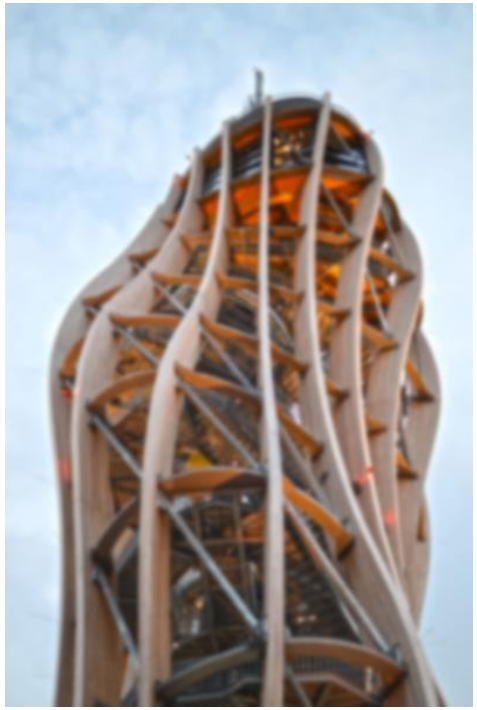
194	128	102	197	69
162	68	103	144	115
121	85	122	27	14
24	183	192	239	150
92	93	154	138	170

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Spatial Filtering

- This is a simple smoothing kernel
- Other operations
  - Sharpen
  - Gradient
  - …

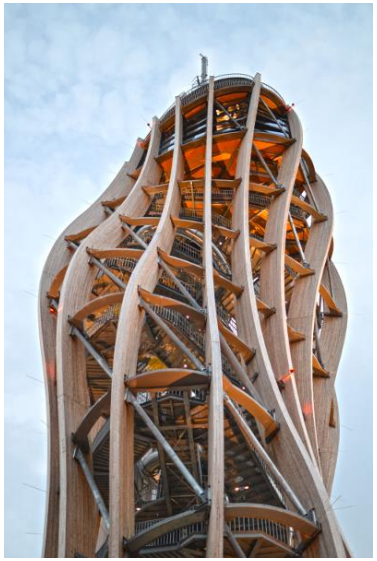


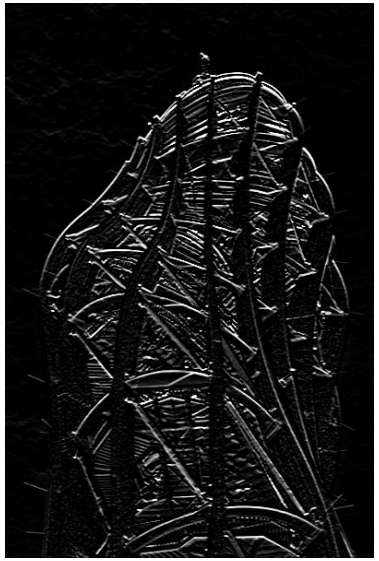


Sobel Filter

- 3x3 kernel

-1	-2	-1
0	0	0
1	2	1

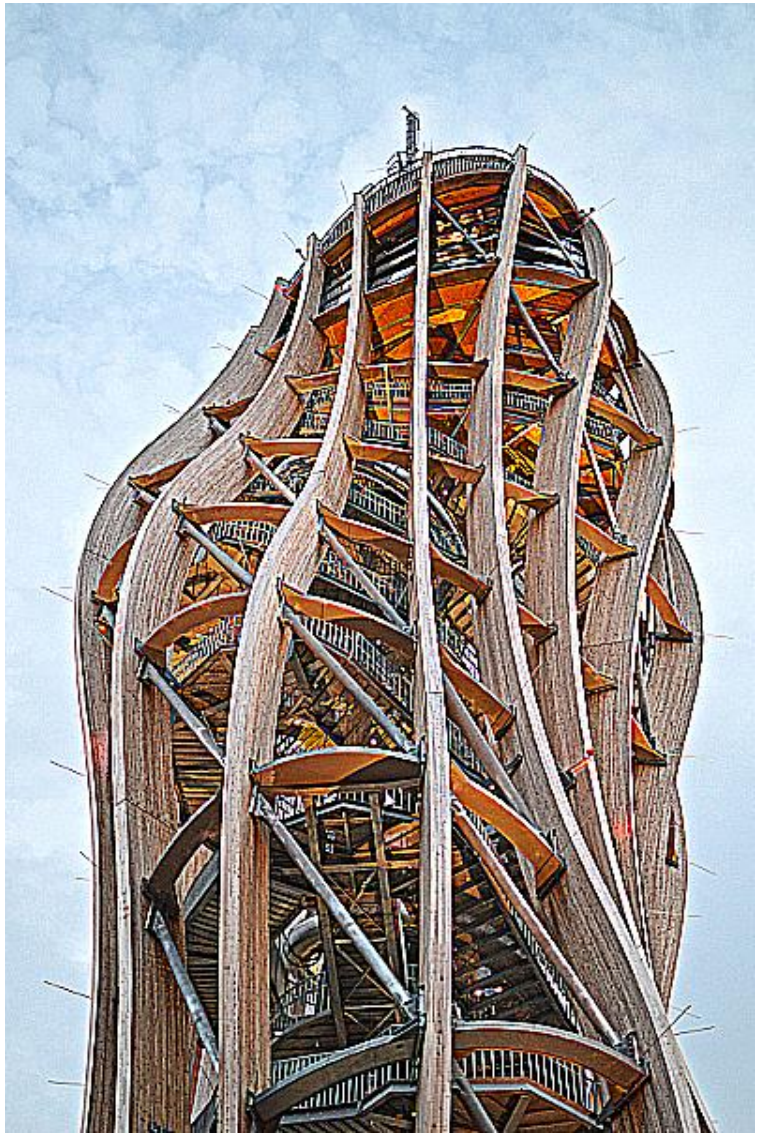




Sharpening

- 3x3 kernel

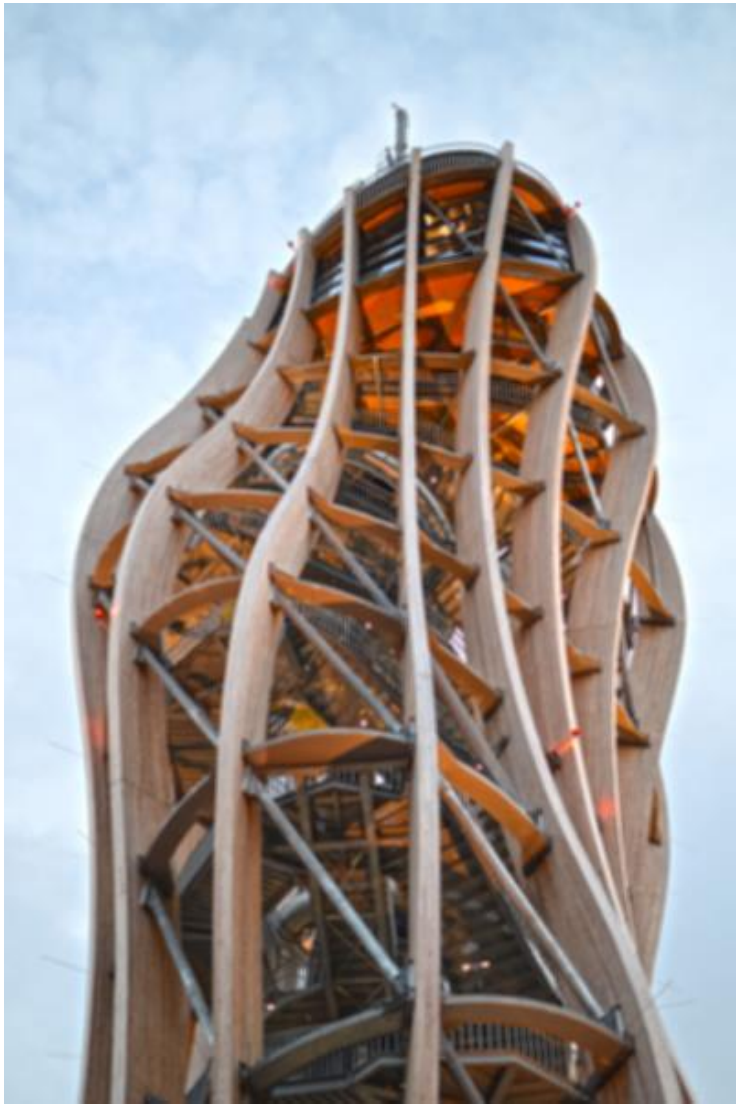
-1/9	-1/9	-1/9
-1/9	9	-1/9
-1/9	-1/9	-1/9



Blur Filter

- 3x3 kernel

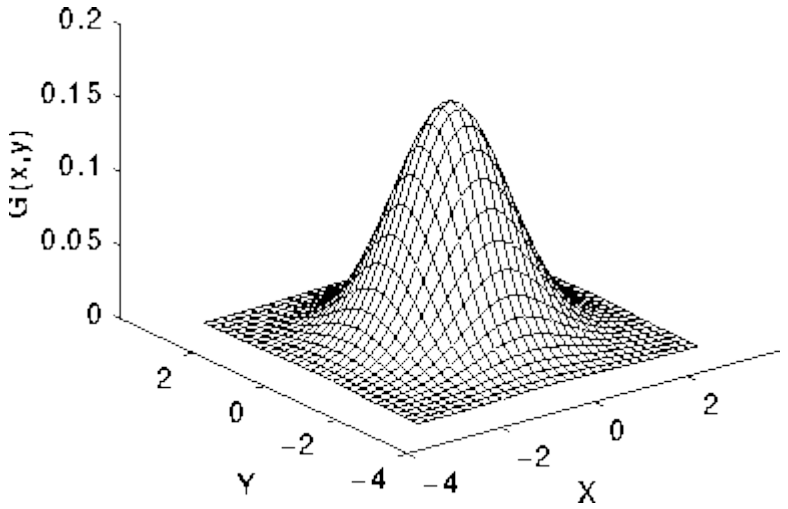
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Gaussian Blur Filter

- Kernel depends on Gaussian distribution

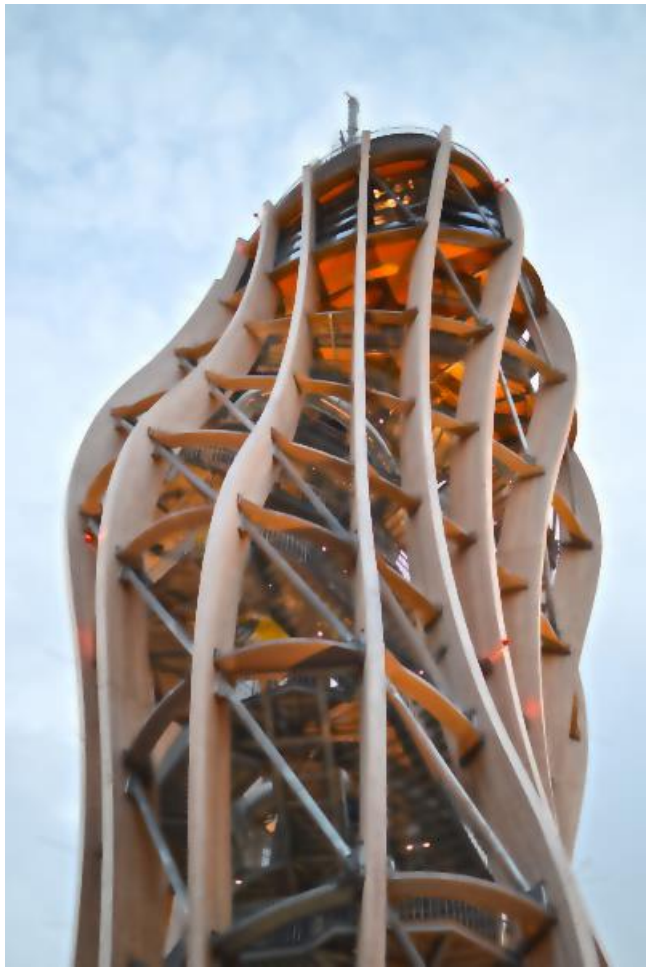
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Selective Blur (Egde Retaining)

- More complicated …

$$I^{\text{filtered}}(x) = \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$



Edge Detection

Edge Detection

- Apply two Kernels
  - with results L_x and L_y
- Then compute the gradient magnitude

$$|\nabla L| = \sqrt{L_x^2 + L_y^2}$$

- Edge direction: $\arctan(L_y/L_x)$

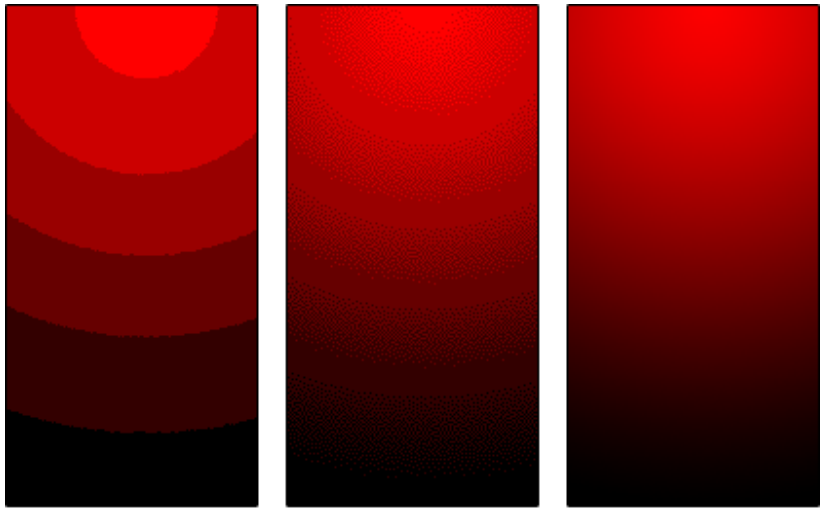
1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

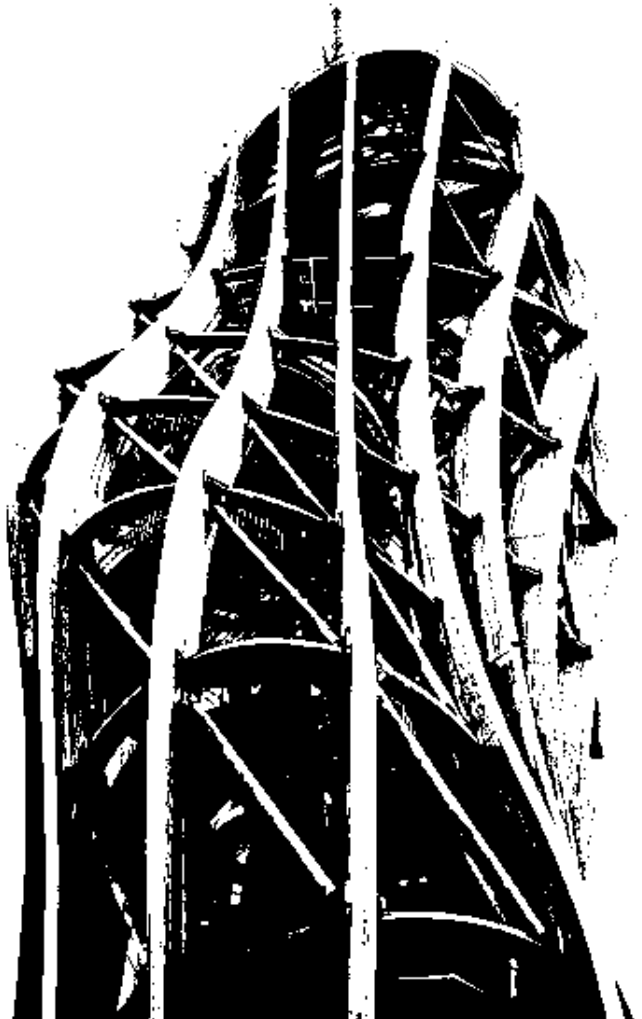
Edge Detection

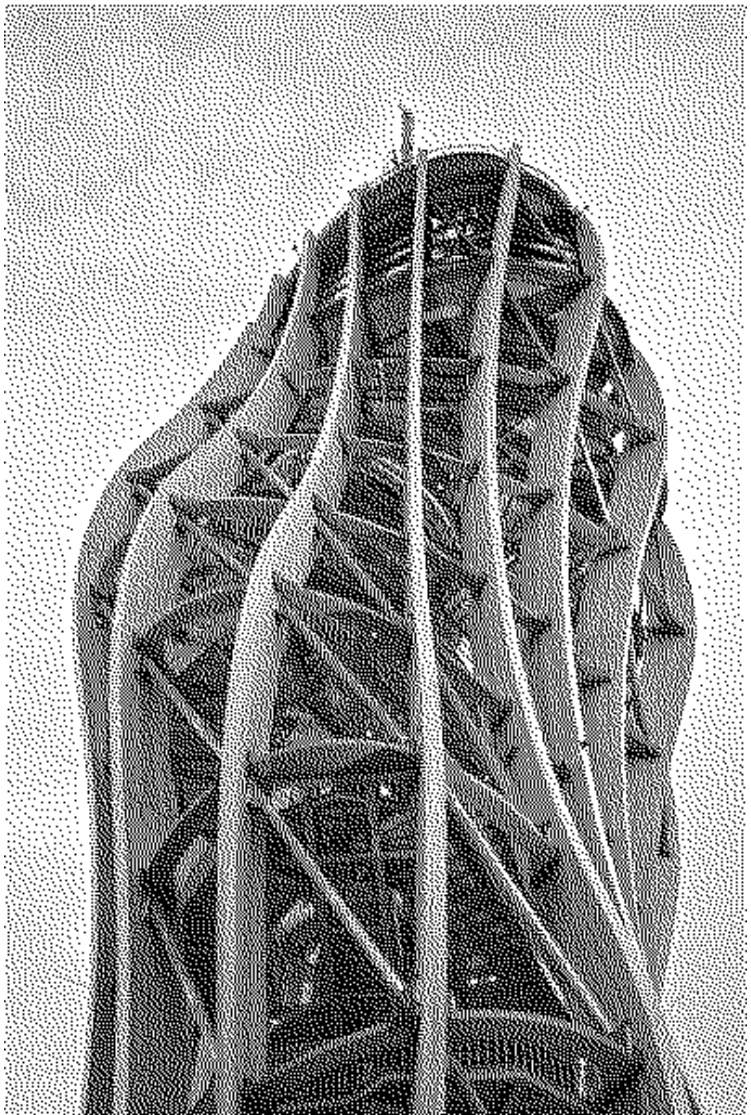
Dithering

- Intentionally introduced noise
  - to prevent color banding.
  - to have a more appealing visual representation with fewer colors.
- Used in newspapers
  - ie. to print grayscale images in black & white

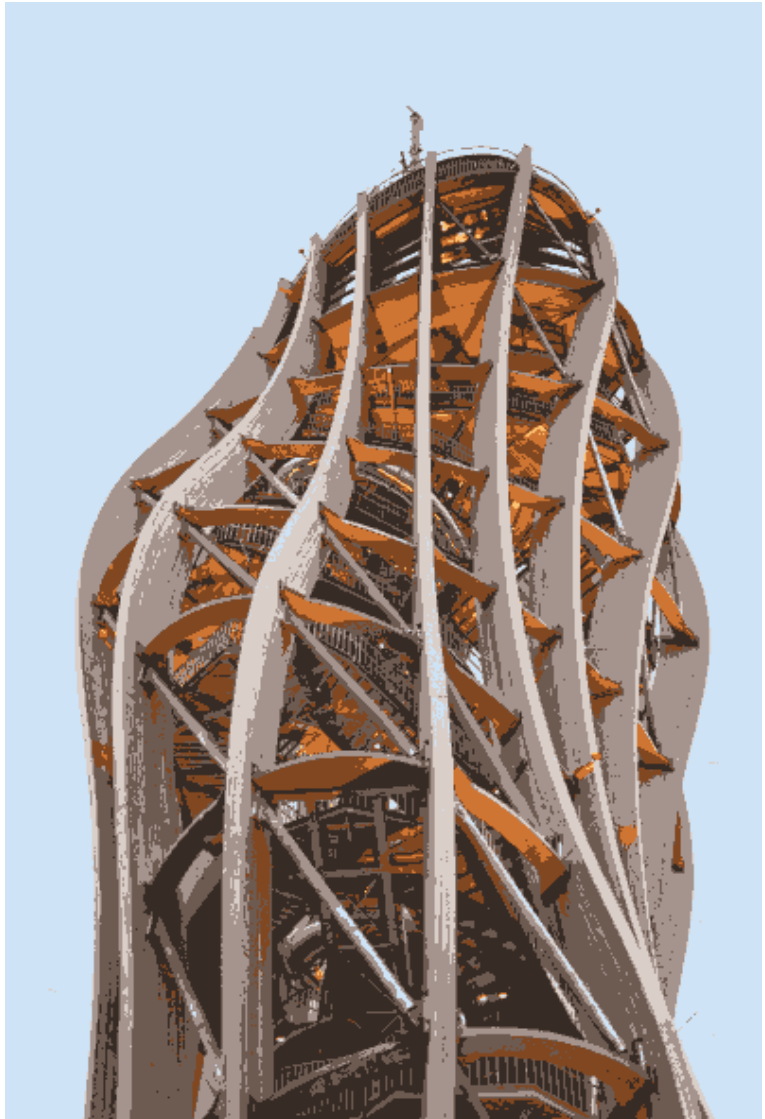


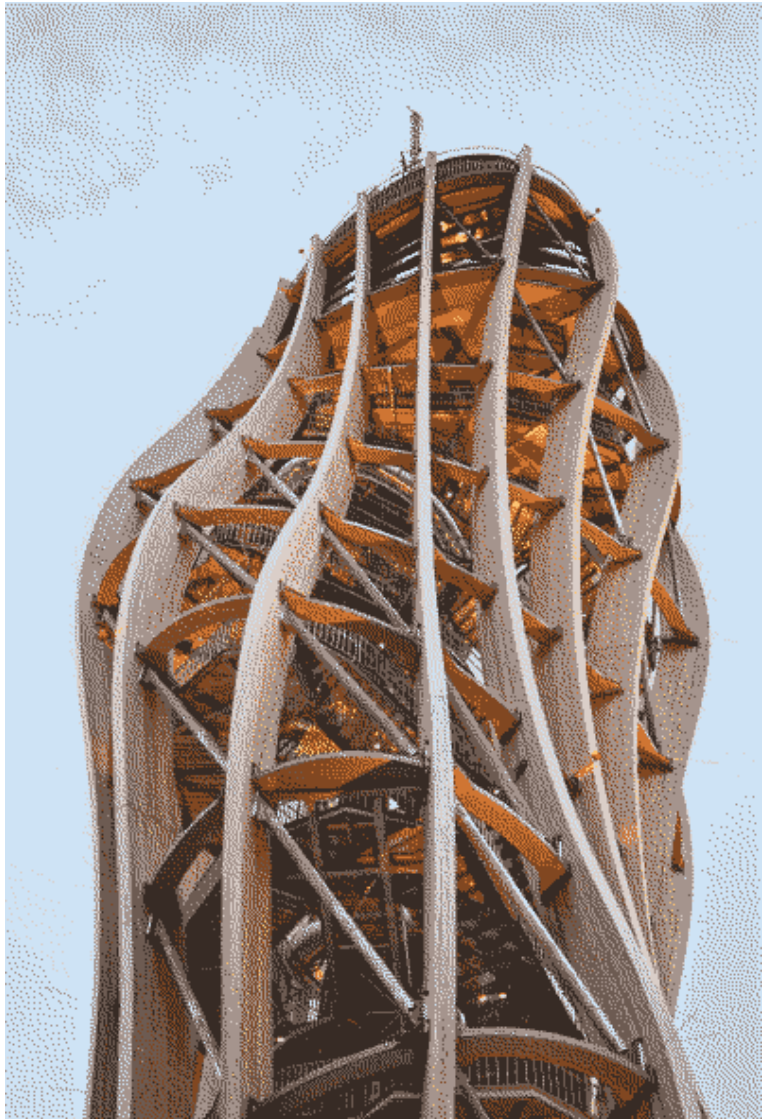
Dithering Example: B & W





Dithering Example: 8 Colors





Floyd-Steinberg Dithering

- Color quantization error is pushed to neighbor pixels

$$\begin{bmatrix} & & * & \frac{7}{16} & \cdots \\ \cdots & \frac{3}{16} & \frac{5}{16} & \frac{1}{16} & \cdots \end{bmatrix}$$

```
for each y from top to bottom
   for each x from left to right
      oldpixel  := pixel[x][y]
      newpixel  := find_closest_palette_color(oldpixel)
      pixel[x][y]  := newpixel
      quant_error  := oldpixel - newpixel
      pixel[x+1][y  ] := pixel[x+1][y  ] + 7/16 * quant_error
      pixel[x-1][y+1] := pixel[x-1][y+1] + 3/16 * quant_error
      pixel[x  ][y+1] := pixel[x  ][y+1] + 5/16 * quant_error
      pixel[x+1][y+1] := pixel[x+1][y+1] + 1/16 * quant_error
```

src. Wikipedia