2. Signal Representations

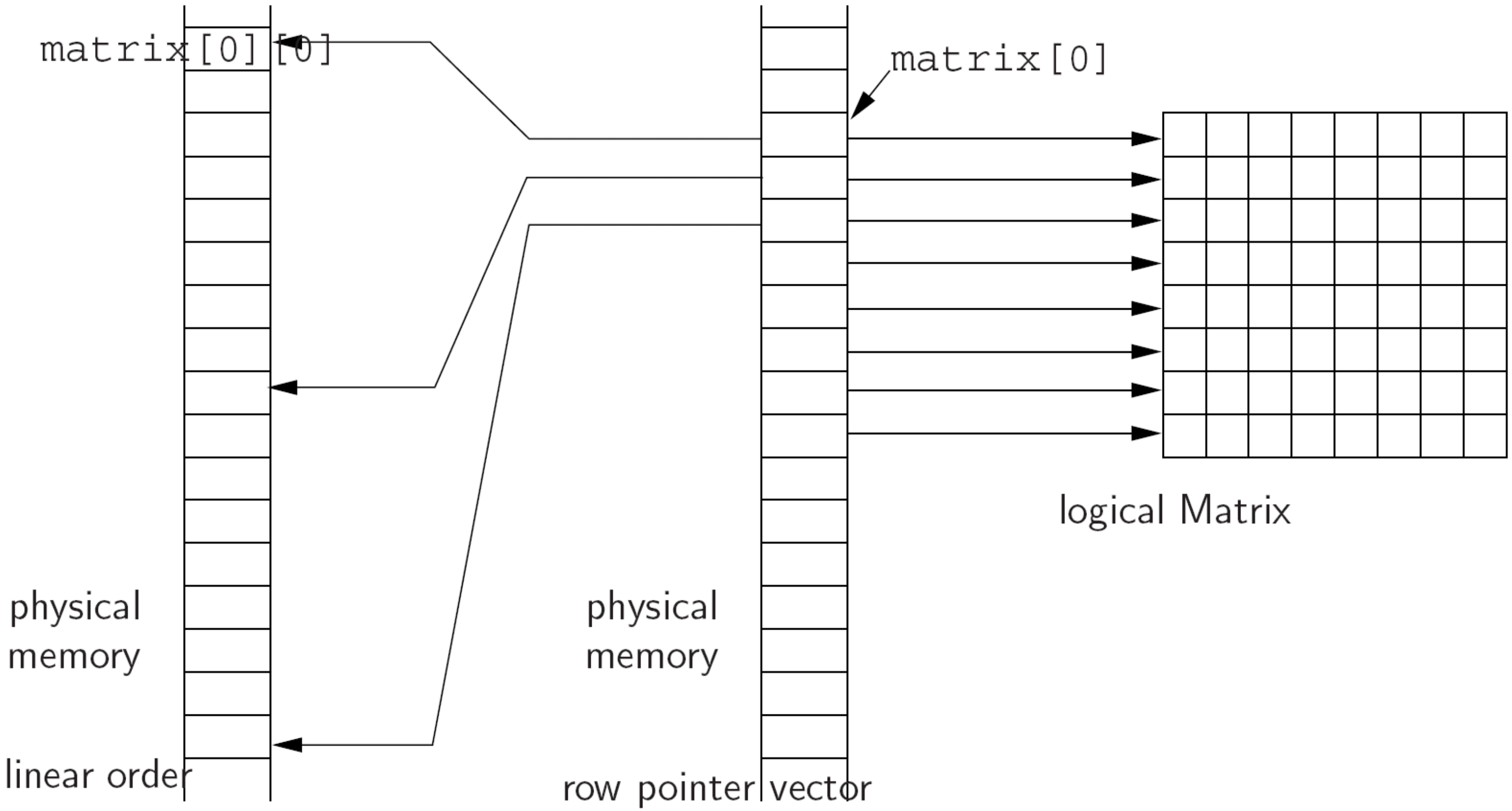
Aim: Learn how audio and video signals can be represented

2.1. Representation of Grey Level Images

Image

- Matrix of small elements
- Elements are called pixels

Representation of a Grey Level Image as an Array



Definition of a Matrix Class

```
template<class T> class Matrix {
  unsigned int xsize,ysize;               // sizes
  T ** matrix;                            // parameterized array
 public:
  Matrix();                               // default constructor
  Matrix(const Matrix&);                  // copy constructor
  Matrix(int, int);                       // constructor with matrix size
  ~Matrix();                              // destructor
  T* operator[] (int);                    // access to vector
  void operator= (const Matrix&);         // assign matrix
  void operator= (const T& v);            // assign v to each element
  const T* operator[] (int) const;        // read only access
  operator T**(){ return matrix; }        // efficient access
  unsigned int sizeX() const {return xsize;}
  unsigned int sizeY() const {return ysize;}
};
```

Definition of a Matrix Class

```
#include "def.h"
#include "Matrix.h"

template <class T> Matrix<T>::Matrix(int x, int y)
{
   xsize= x; ysize= y;
   T* array = new T[x*y];          // vector of size x*y
   matrix = new T*[y];              // generate T matrix
   for (int i = 0; i < y; ++i)
      matrix[i] = & (array[i*x]); // fill in vector pointers
}

template <class T> T* Matrix<T>::operator[] (int i)
   { return matrix[i]; }

template <class T> const T* Matrix<T>::operator[] (int i)
const
   { return matrix[i]; }
```

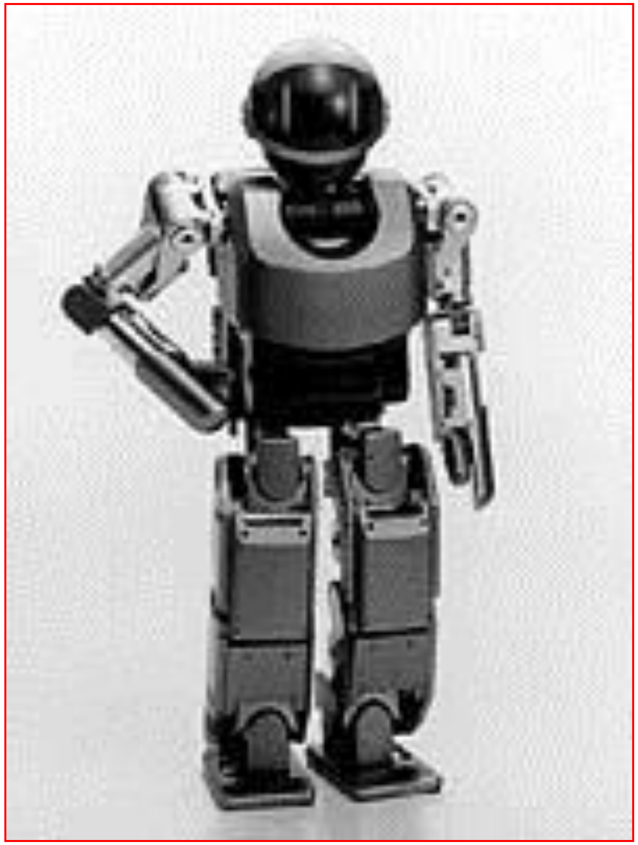
Usage of the Matrix Class

```
// define a matrix of integers

Matrix<int> m1(256,256);

// define a larger matrix of floats

Matrix<float> m2(512,256);

// access one element (cold be made secure, by checking the
   indexes)

int c1= m1[2][100];

float c2= m2[5][120];
```

Definition of Gray Level Image Class

```
class GrayLevelImage_V0 {
  float focus;            // focal length
  float aperture;         // lens aperture
  float scaling;          // pixel side relation
  char * description;     // textual information
  Matrix<byte> image;     // the pixels
 public:
  GrayLevelImage_V0(int,int);                // constructor
  ~GrayLevelImage_V0();                      // destructor
  int isEqual(const GrayLevelImage_V0&);  // test equality
  // etc.
  byte * operator [] (int i) { return image[i]; }
// delegation
  const byte * operator [] (int i) const { return image[i]; }
// delegation
  unsigned int sizeX() const { return image.sizeX(); }
  unsigned int sizeY() const { return image.sizeY(); }
  float focalLength() const { return focus; }
};
```

Image



Example for ppm Format (Grey-Level)

http://netpbm.sourceforge.net/doc/ppm.html

"Magic number"
type of coding

width height

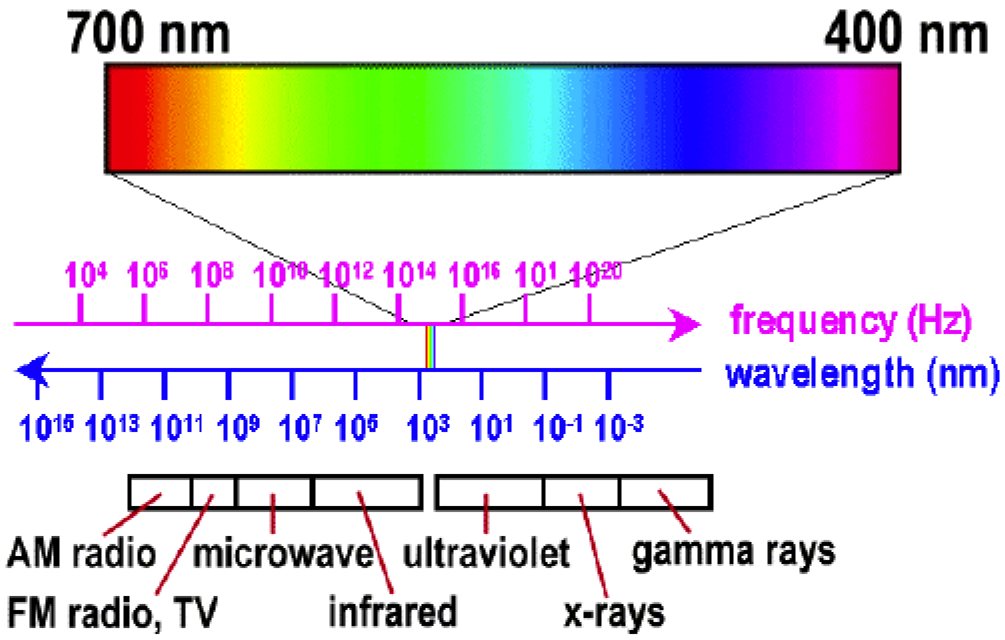
Maximum grey level
value

pixel

```
P2
# CREATOR: XV Version 3.10a  Rev: 12/29/94 (PNG patch 1.2)
150 200
255
252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 250
250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250
250 250 250 250 250 250 251 251 251 251 251 251 251 251 250 250 250
250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250
250 250 250 250 251 250 250 249 248 247 246 246 250 250 250 250 250
250 250 250 250 250 250 250 250 250 251 250 250 250 250 250 250 250
250 250 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252
252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252
252 252 252 252 252 252 252 252 253 253 253 253 253 253 252 252 252
252 252 252 252 252 252 252 252 252 252 252 252 252 250 250 250 250
250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250
250 250 250 251 251 251 251 251 251 251 251 250 250 250 250 250 250
250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250
250 251 251 251 251 251 251 250 250 250 250 250 250 250 250 250 250
250 250 250 250 250 250 251 250 250 250 250 250 250 250 250 250 252
252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252
252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252 252
252 252 252 252 252 253 253 253 253 253 253 252 252 252 252 252 252
252 252 252 252 252 252 252 252 252 252 250 250 250 250 250 250 250
250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250 250
…
```

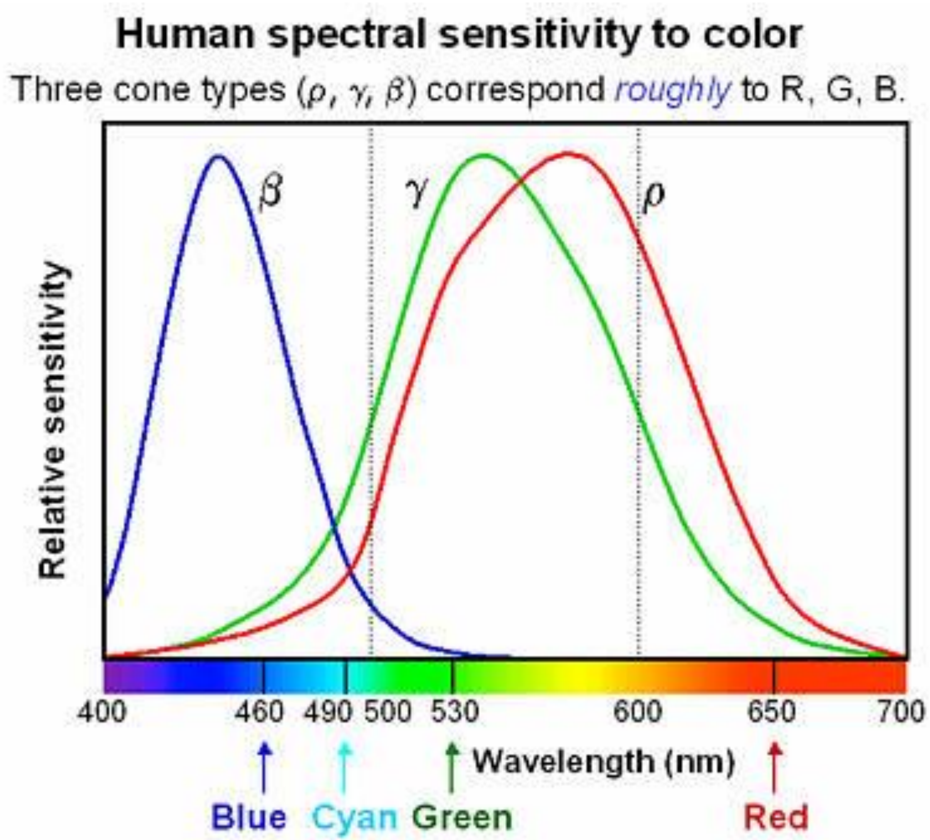
2.2. Representation of Color Images

Light: Electromagnetic Radiation



Only small fraction of spectrum is visible

Human Perception of Light



Three different cone type in the human eye are sensitive to red (R), green (G) and blue (B)

RGB-Color Space

Red

Green

Blue

Image

Example for ppm Format (Color)

http://netpbm.sourceforge.net/doc/ppm.html

P3
\# CREATOR: XV Version 3.10a Rev: 12/29/94 (PNG patch 1.2)
150 200
255
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
252 253 248 252 253 248 252 253 248 252 253 248 252 253 248
252 253 248 252 253 248 252 253 248 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 254 250 247
254 250 247 254 250 247 254 250 247 254 250 247 254 250 247
254 250 247 254 250 247 255 251 248 254 250 247 254 250 247
253 249 246 252 248 245 251 247 244 250 246 243 250 246 243
252 251 246 252 251 246 252 251 246 252 251 246 252 251 246
252 251 246 251 252 247 251 252 247 251 252 247 251 252 247
250 252 249 250 252 249 250 252 249 250 252 249 250 252 251
250 252 249 251 251 249 251 252 247 251 252 247 251 252 247
251 252 247 251 252 247 251 252 247 251 252 247 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249
253 254 249 253 254 249 253 254 249 253 254 249 253 254 249

…

Different magic number!
P3: RGB color encoding

Three consecutive numbers encode the red, green and blue parts of the pixel

Structure for Color Image

```
struct ColorImage_V0 {          //
Version 0
  Matrix<byte> * r;             //
color channel red
  Matrix<byte> * g;             //
color channel green
  Matrix<byte> * b;             //
color channel blue
};
```

YUV-Color Space

- Used in video
- Y is the grey level
- Linear transform

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

XYZ-Color Space

- Used for image compression
- Y is the grey level
- X and Z: chrominance
- Linear transform

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.111 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

HSI Color Space

- I: intensity

$$I = \frac{R+G+B}{3}$$

- S: saturation $S = \sqrt{R^2 + G^2 - R*G - R*B - B*G}$

- H: hue (Farbtönung; Färbung)

$$H = \begin{cases} 1 & \text{if } \mathrm{G} = \mathrm{B} \\ (\alpha - \tan^{-1}((R-I)\sqrt{3}/G-B)))/(2\pi) & \text{else} \end{cases}$$

JPEG

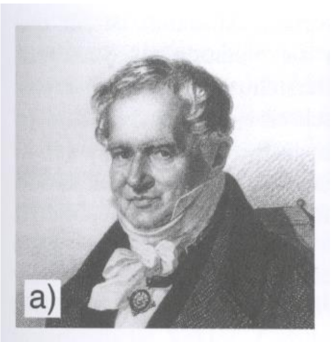
- Compressed format
- Lossy compression
- Compression level can by adjusted by user
- Software implementation non trivial

Steps in JPEG Compression

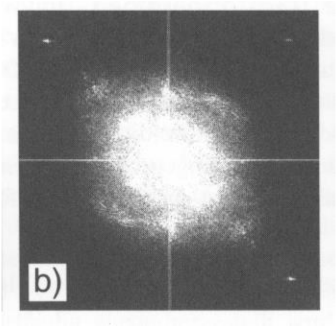
1. Transform the image into an optimal color space.
2. Downsample chrominance components by averaging groups of pixels together.
3. Apply a Discrete Cosine Transform (DCT) to blocks of pixels, thus removing redundant image data.
4. Quantize each block of DCT coefficients using weighting functions optimized for the human eye.
5. Encode the resulting coefficients (image data) using a Huffman variable word-length algorithm to remove redundancies in the coefficients.

Bildkompression

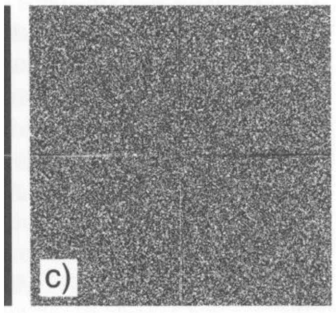
Original



Betrag der FFT



Phase der FFT



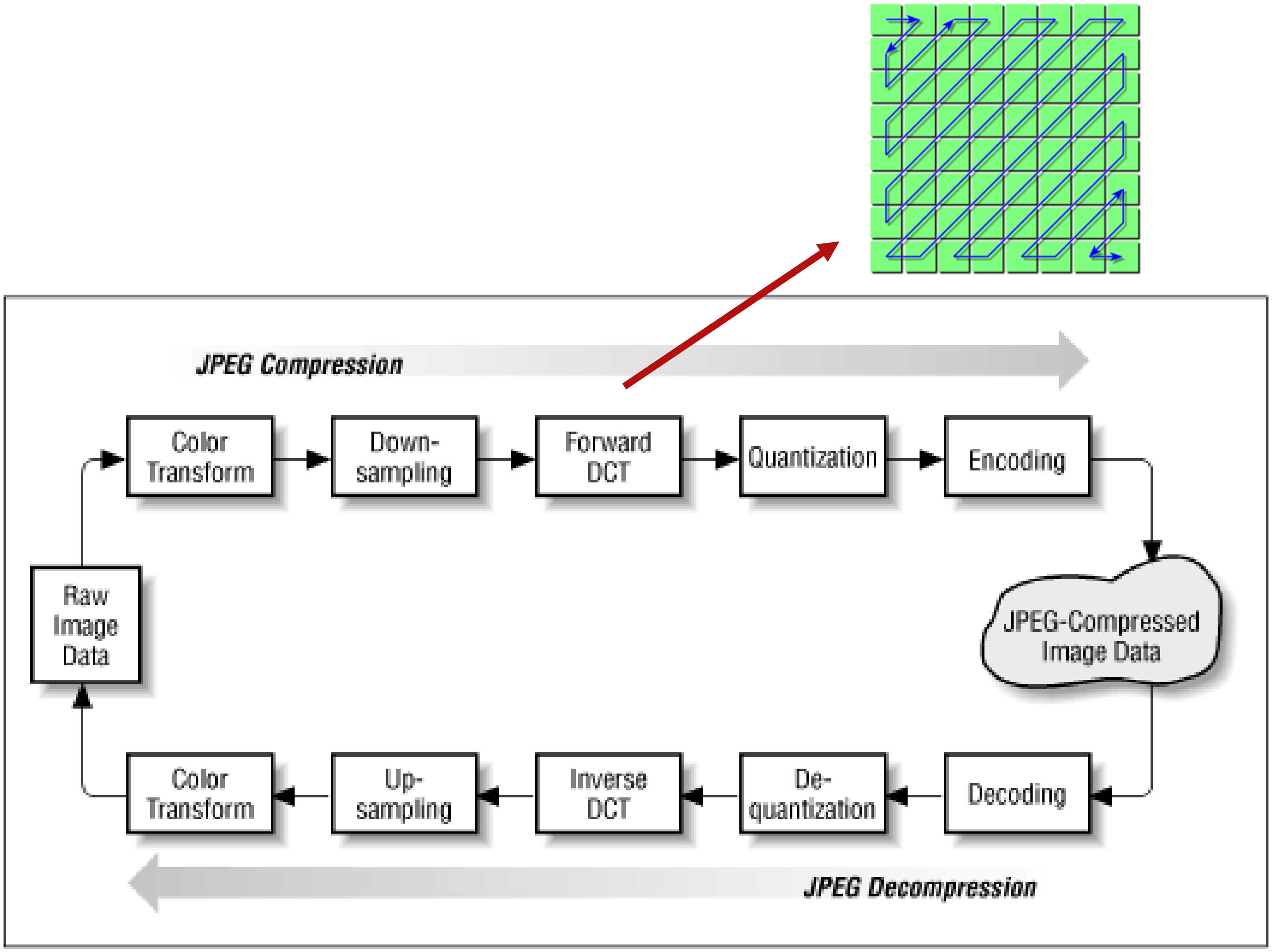
Idee: Speichere nur einen Teil der Parameter

Rücktransformation:
5% der DCT-Parameter

40% der DCT-Parameter

Aus: Beate Meffert und Olaf Hochmuth „Werkzeuge der Signalverarbeitung“

Steps in JPEG Compression



2.3. Representation of Audio

Parameters of audio encoding

- Sampling rate (e.g. 44100 for CDs)
- Data size (e.g. 8 bit or 16 bit)
- Data encoding (e.g. linear)
- Channels (e.g. 1 for mono)

Wav File Format

endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1ID	4
little	16	Subchunk1Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2Size	4
little	44	data	Subchunk2Size

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

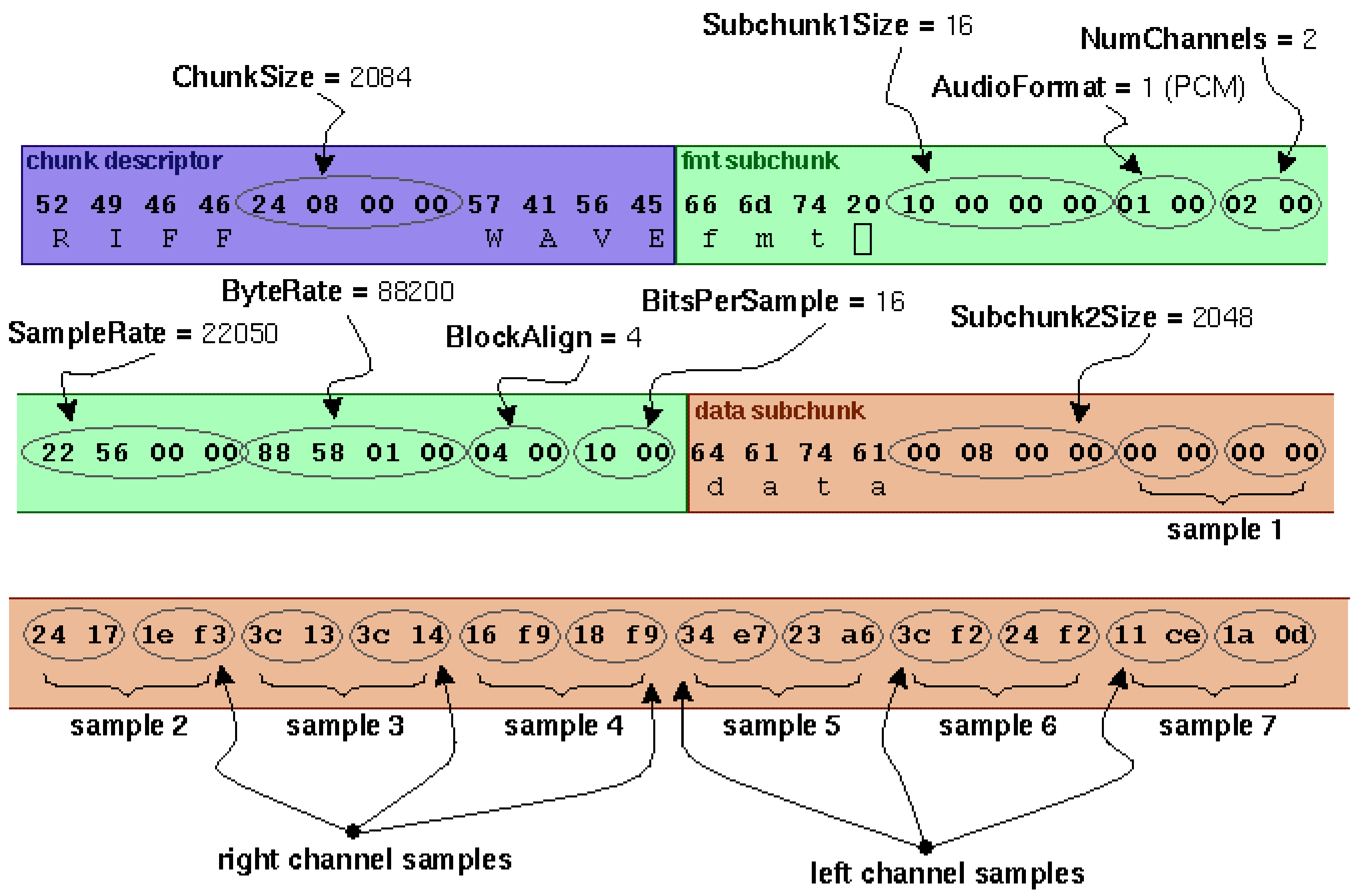
The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

Indicates the size of the sound information and contains the raw sound data

Wav File Format



sox - Sound eXchange : universal sound sample translator

- Converts different sound formats
  - e.g. wav to raw ASCII
- Available on most UNIX computers
- Cutting, rate conversion, filters, …

Images and Audio in Maple

- See maple script

Summary

- Grey level image: matrix of pixel
- Color images:
  - Color spaces
  - Compression
- Audio:
  - Formats and conversion