Computer Graphics Charles University

Raster Image File Formats

© 1995-2016 Josef Pelikán & Alexander Wilkie

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

Raster Image Capture

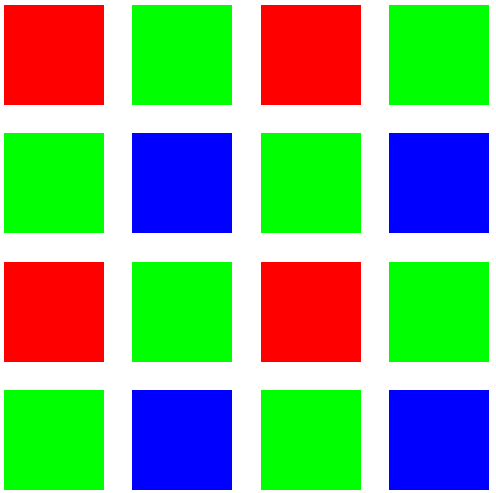
- **Camera**
  - **Area sensor** (CCD, CMOS)
  - Colours: usually a Bayer mask
  - Data readout a bottleneck
  - Raw data has to be processed before use
    - » Specialised processors (DIGIC..)
- **Scanner** (film, copier)
  - Usually a **linear sensor** (1D)
  - Simple read-out, mechanical scan needed

Area Sensors

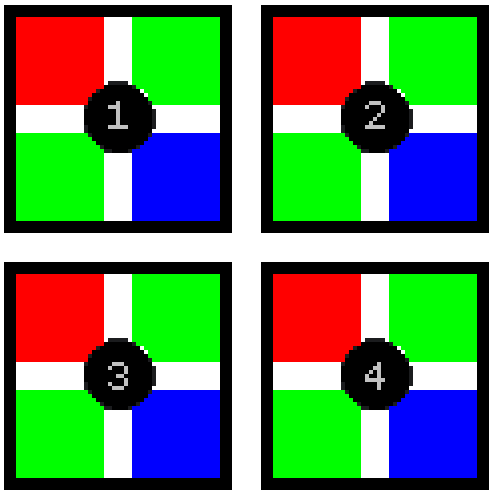
- **Size and resolution**
  - Larger size – less noise
    … but the lens has to be better
  - Greater resolution – more noise per pixel
- **Sensor sensitivity** (ISO)
  - Amplification only before ADC
  - Greater sensitivity (gain) – more noise
- **Colour capture**
  - Bayer mask, only removed during RAW processing

Bayer Mask

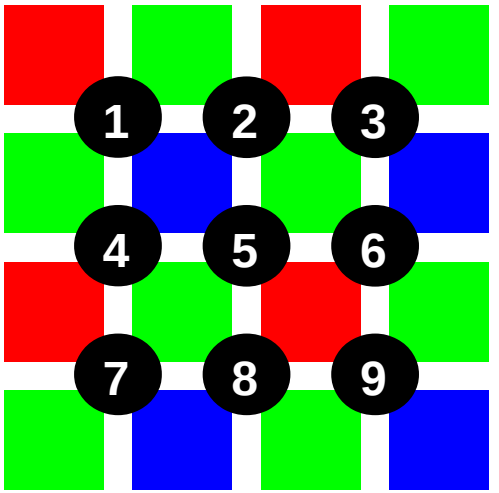
- RGB colour filters
  - Each component measured separately
  - Sensor is less effective, but easy production



Sensor mask



Colour value



Efficient computation

Pixel Colour Formats

- **Colour palette** (8 bit)
  - Global colour table (palette, „colormap“)
  - A pixel contains the **index** of a colour in the table
- **B&W, greyscale pixels** (1 bit, 8-16 bit)
  - 1-bit „bitmask“ (e.g. fax transmissions)
  - Greyscale values, gamma corrected
- „**true-color**“ (24-48 bit)
  - Most common colour space these days (RGB), gamma
- „**hi-color**“ (15-16 bit)
  - „Reduced“ full colour, 5-5-5 or 5-6-5 bit (RGB)

Graphics Formats

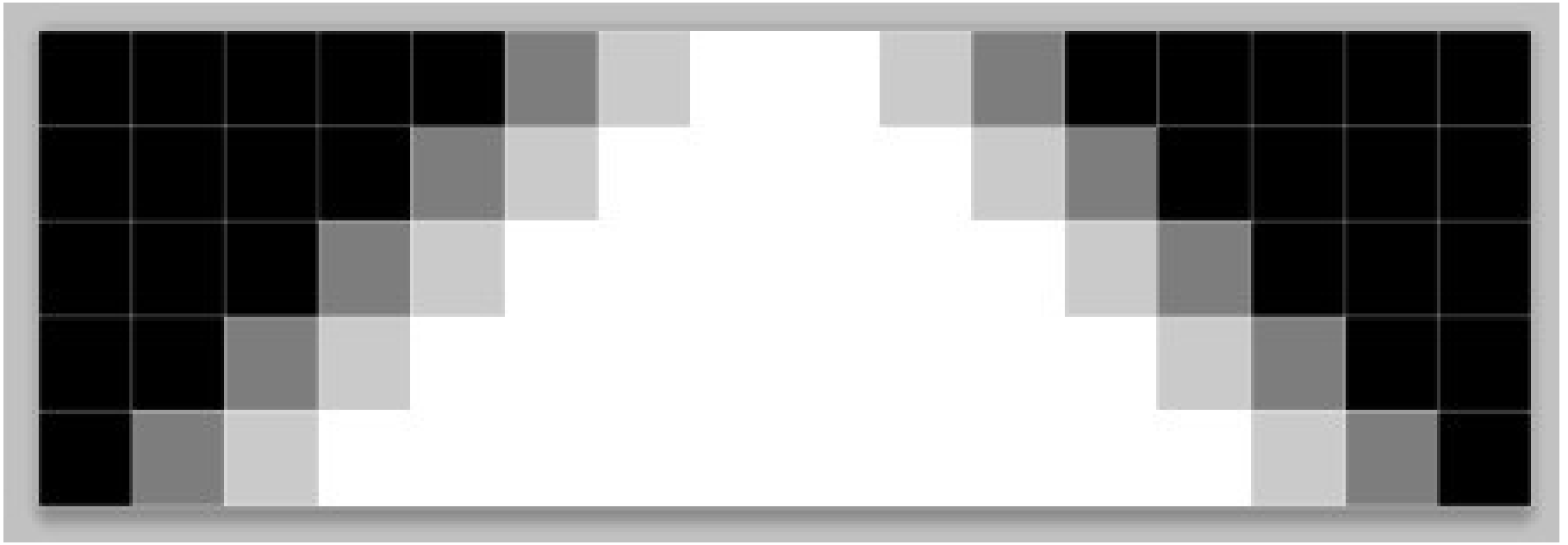
- **Raster**
  - Regular **matrix of pixels** („bitmap”)
  - MS-Windows Bitmap (BMP), Portable Network Graphics (PNG), CompuServe GIF, Interchange File Format (IFF), JFIF (JPG), PBM/PGM/PPM/PFM, Macintosh (PICT), Targa (TGA), Tagged Image File Format (TIFF), ...
- **Vector**
  - A sequence of **objects** or **commands** (scalable)
  - CorelDraw!™ (CDR), Scalable Vector Graphics (SVG), AutoCAD™ (DXF), Adobe Illustrator™ (AI), Adobe PDF™, PostScript™, Windows Metafile (WMF), ...

Raster Graphics Formats

- **Colour Information**
  - Palette, greyscale, true colour, alpha channel, ...
- **Compression**
  - **Lossless** / **Lossy**
  - **RLE**: TGA, BMP; **LZ***: PNG, GIF, TIFF; **JPEG**: JFIF, TIFF
- **Storage arrangement**
  - Interlaced/progressive/tiled/... (PNG, GIF, TGA, JFIF, ..)
- **Non-graphical information**

  (annotations, copyright, date, colourspace..)
  - All modern formats (TIFF, PNG, GIF, ..)

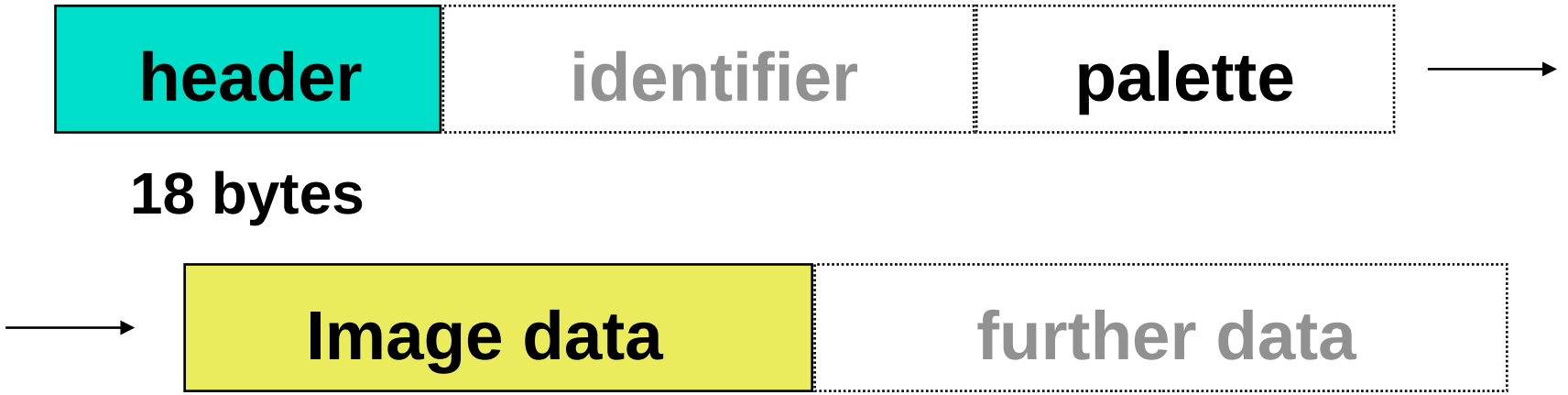
PGM / PBM / PPM

- Very simple **raster format**
- Simple text header + txt or binary data
  - No compressions
  - Pixel formats: B/W (P1/4), grey (P2/5), RGB (P3/6)
- 

Targa Format (Truevision Inc.)

- Simple **raster format**
- **Hardware oriented**
  - **Targa** video-adapters (Targa 16, Targa 24, ..)
- Several different **colour formats**
  - RGB, RGBα, grey, palette, attribute bits
  - Several **compression methods** (**RLE**)
- Various types of **interleaving** (network transmission)

Structure of TGA Files



File header:

- Colour format (palette, RGB, RGBα, grey)
- Identification (ASCII text, maximum 256 chars)
- Compression: without, RLE, Huffman, delta-modulation
- Image size: $[X_0, Y_0]$, width, height
- Orientation (portrait, landscape)

TGA Pixel Formats

Palette, greyscale

8 or 16 bits

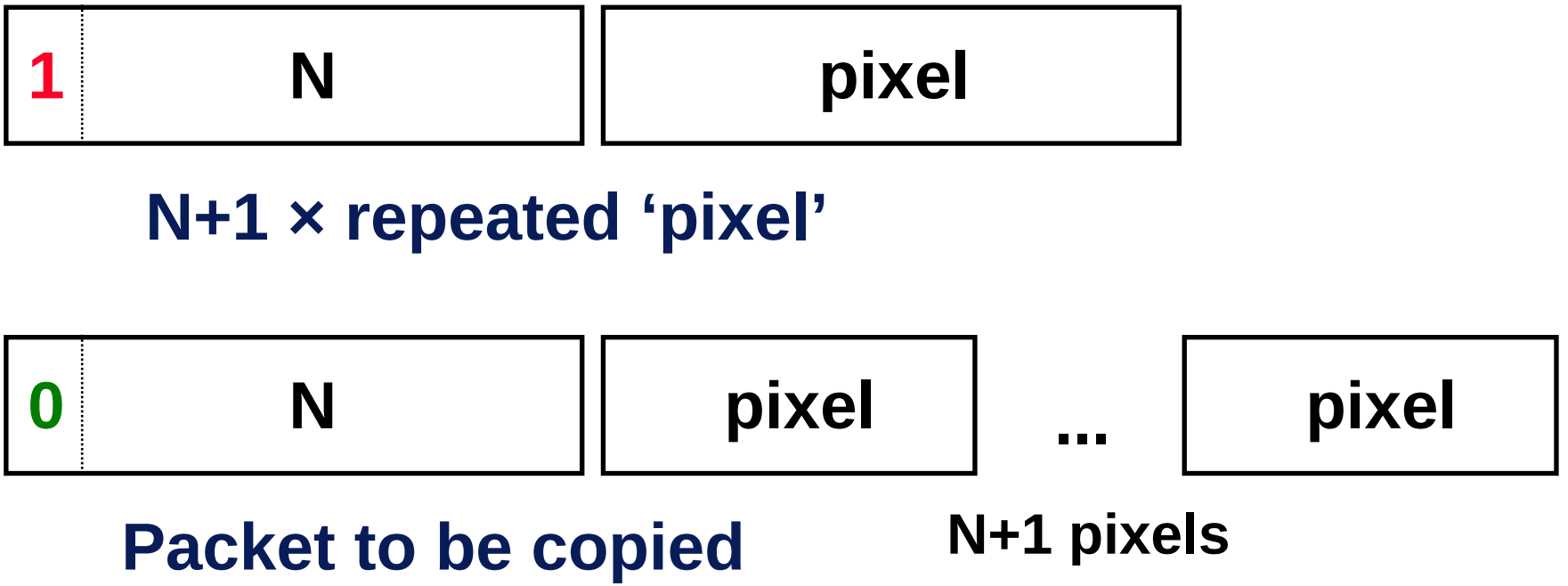
RGB 16

attribute **16 bits**

RGB 24

RGB 32

RLE Compression in TGA Files



Maximal packet size is 128 pixels

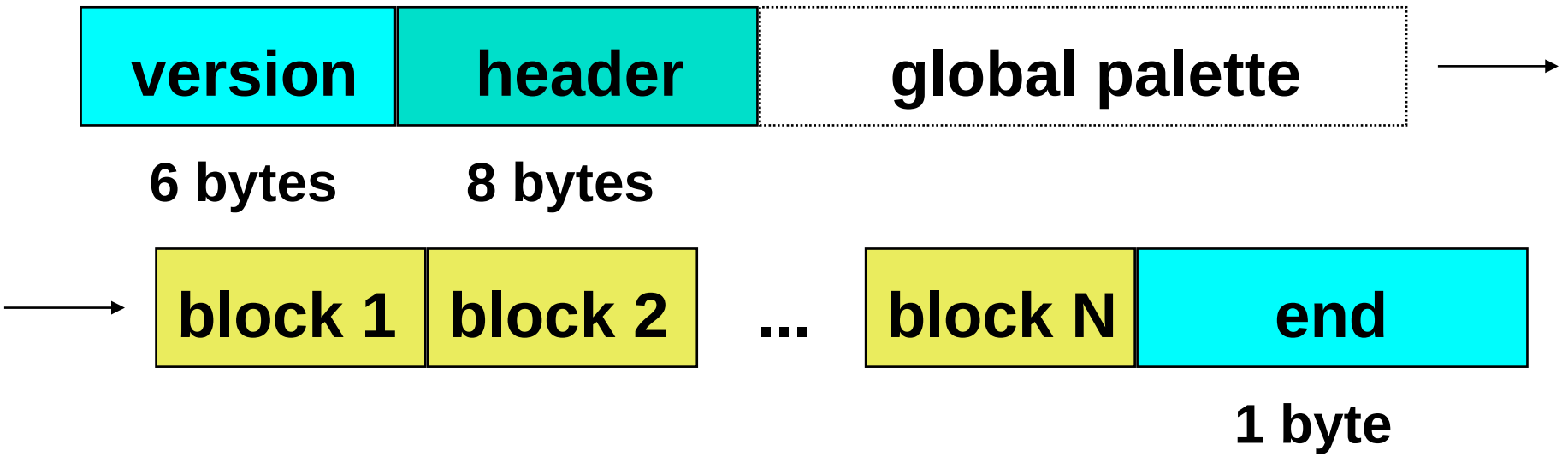
– Larger packets do not yield significant benefits

GIF Format (CompuServe Inc.)

Graphics Interchange Format (version 87a, 89a)

- **Raster format**, relatively hardware-independent
- **Only palette images** (max. 256 colours)
- **LZW compression** with dynamic coding
  – patent of UniSys Inc. (valid until 1995)
- Optional 4-phase interleaving (network transmission)
- **Further feaatures**: more than one image per file (animations!), single transparency colour (bad compositing)

GIF File Structure



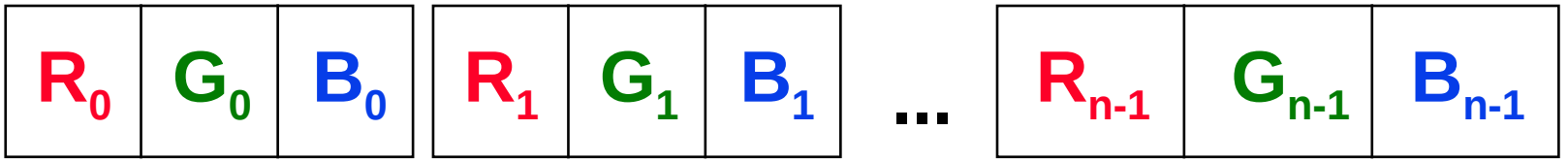
Version: **'GIF87a' or 'GIF89a'**

Global header:

- Width and height **of the virtual screen**, bits per pixel, background colour, „pixel aspect ratio" (4/1 to 1/4)
- **Global palette**: length, grading

GIF File Format

Palette:



(n×3) bytes

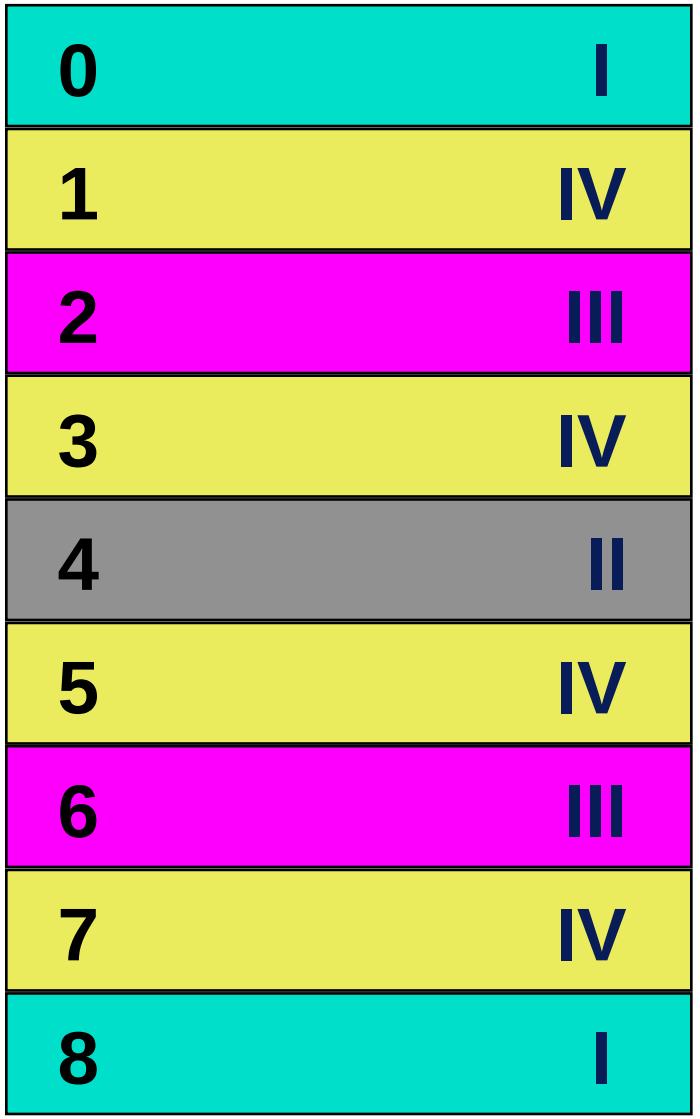
Blocks:

- Image or other data (application data)
- **Growable format**: older versions of the decoder simply skip unknown blocks

GIF Image Blocks

- **Dimensions**
  - $[X_0, Y_0]$, width, height
- **optional local palette**
  - Number of colours, sorting
- Optional – **interlacing**
  - 8 phases of the picture
- **Image data**
  - Initial length of LZW code, data

Interleaving



I. phase: lines 8i

II. phase: lines 4 + 8i

III. phase: lines 2 + 4i

IV. phase: lines 1 + 2i

GIF Expansion Blocks (v. 89a)

Graphics control block:

- **Animation parameters**
- **Interaction**
- Definition of **transparent colours**

Comment Block (free text)

Text Block

- Text to be written on the image (simple font)

Application block:

- Free binary data (e.g. FractInt parameters)

LZW Compression (Lempel-Ziv-Welch)

- **Dictionary compression technique**
  - **dirctionary**: assignments „**phrase → code**”
  - **phrase**: a pixel sequence
  - **code**: **n**-bit word (**3 ≤ n ≤ 12**)
- Changes during encoding
  - **Dictionary** (adaptive data encoding)
  - **Codeword length** „**n**“ increases by one to 12

Structure of the Coding Algorithm

1. **initialisation**
   - All single pixel phrases are stored in the dictionary
   - **Act :=** (empty string)
2. read more pixels into **K**
3. is the phrase **Act + K** already in the dictionary?
   - **Yes**: **Act := Act + K**
   - **No**: output code phrase **Act**
     put **Act + K** in the dictionary
     Act := K
4. if there is still input, repeat steps 2 and 3
5. output the code phrase **Act**

Adding Phrases to the Dictionary

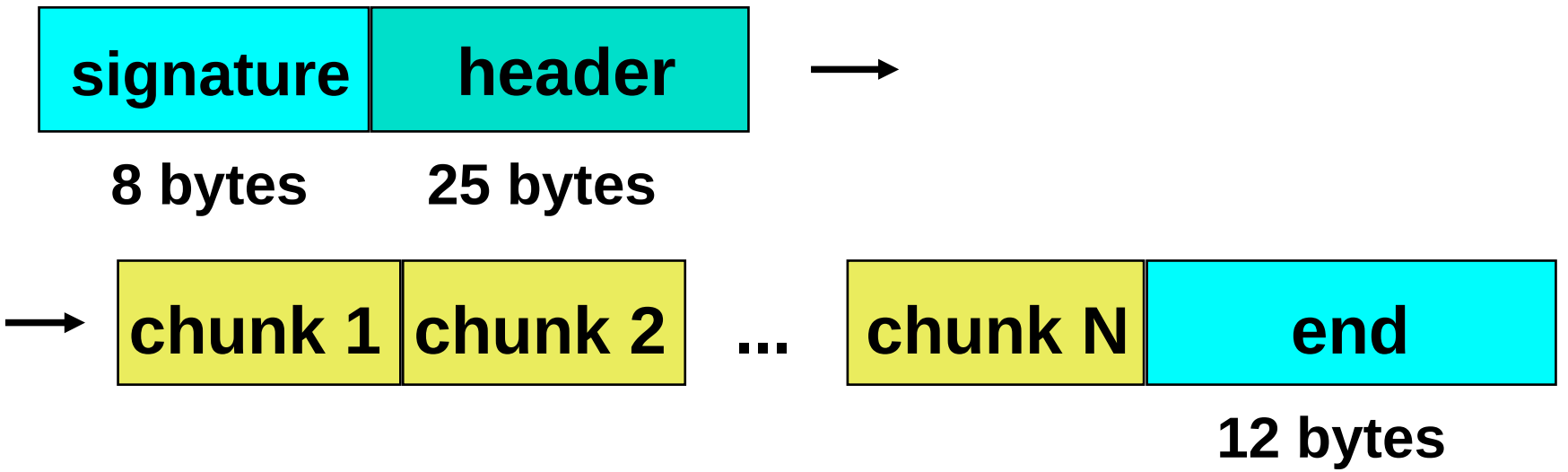
- **Initial dictionary**:
  - codes **0 ÷ 2^p- 1** .. single pixel phrases
  - code **2^p** = „**reset**“ (dictionary overflow)
  - code **2^p + 1** = end symbol (EOF)
  - first free code phrase = **2^p + 2**
  - initial code word length: **n = p + 1** bits
- **After issuing code 2^p, increase n by 1**
  - maximum value of **n** is **12** (**4094** fráze)
  - in case of overfill freeze dictionary (less common) or send the „**reset**” **code** (reinitialisation)

PNG Format (Portable Network Graphics)

W3C Consortium

- **Raster format** developed for WWW
- Several **colour formats**
  - palette, grey, true-color, continuous transparency
  - 8 ÷ 16 bits per channel
- Information to compensate **HW properties**
  - Gamma, gamut, white point
- **DEFLATE** compression based on LZW77
- Optional **progressive loading** in 7 phases

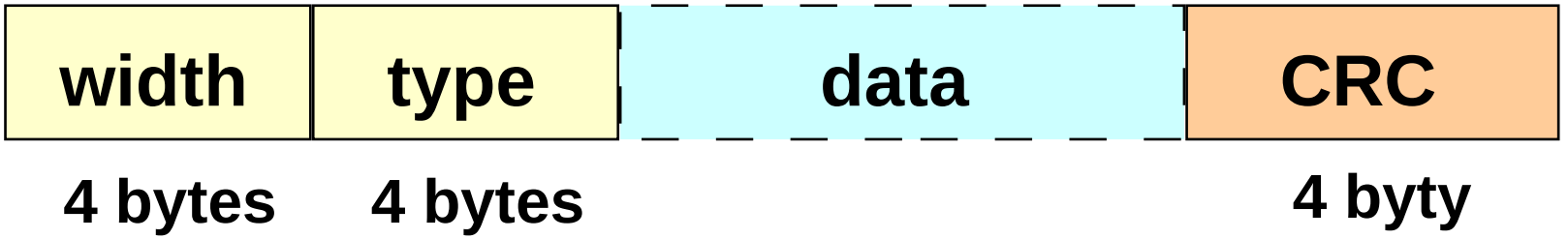
PNG File Structure



Header:

- **Image width and height, bit depth**
- **Colour format** (paleta, grey, true-color, transparency)
- Compression, prediction, interleaving

PNG chunk



- **Image or other data** (palette, transparency, HW compensation, text, ..)
- **Uniform format** (unknown chunks are to be skipped)

LZ77 Compression (Lempel-Ziv)

- **Lossless** compression method (sliding window)
- Encodes **data sequences**
  - **Phrase:** sequence of characters (**pixels**)
- A code is a **triplet** [offset, width, character]:
  - **offset**: relative distance of phrase start
  - **width**: phrase length in pixels
  - **character**: the pixel that follows the phrase
- Generalised encoding runs

```
begin
     fill view from input
     while (view is not empty) do
     begin
          find longest prefix p of view starting in coded part
          i := position of p in window
          j := length of p
          X := first char after p in view
          output(i,j,X)
          add j+1 chars
     end
end
```

DEFLATE Compression / PNG

- **Two phases:**
  - **LZ77** for scanlines
  - **Huffman encoding**
    - » offset
    - » Length, character
- Additionally: **selectable prediction**
  - The standard defines five prediction filters
  - They can be switched at each line start

PNG Interlacing

- **7 unequal phases**
  - in the first phase, **1/64 of pixels** are transmitted

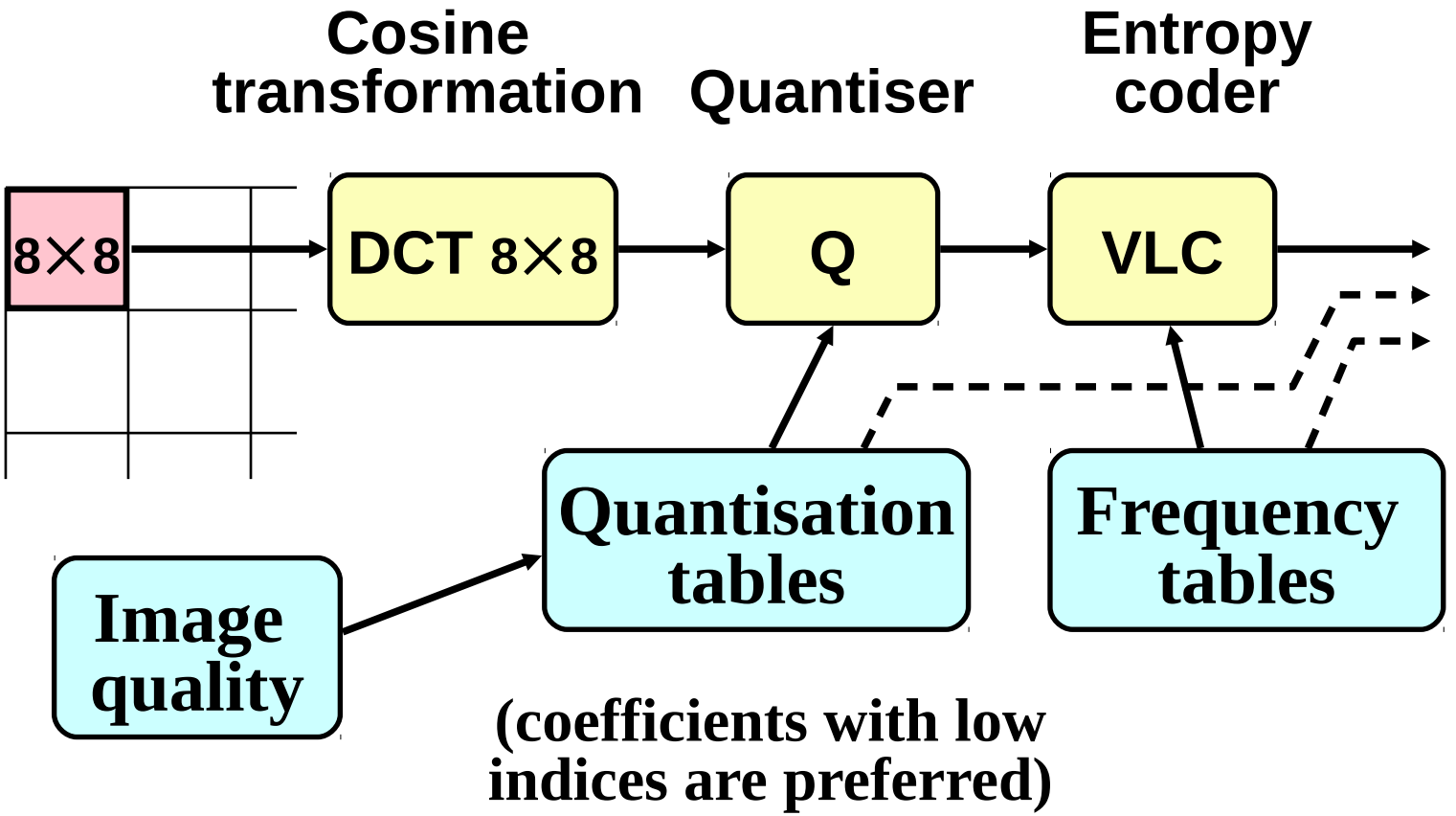
1	6	4	6	2	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7
3	6	4	6	3	6	4	6
7	7	7	7	7	7	7	7
5	6	5	6	5	6	5	6
7	7	7	7	7	7	7	7

JPEG Compression

Joint Photographic Experts Group (1990)

- **Lossy compression** of raster data
- Suitable for **natural images** (photos, renderings)
- Not so suitable for **discrete graphics** (fonts)
  - Compression artefacts
- Optional **output quality** parameter
- Options: progressive mode, hierarchical coding
- **The file format is actually called JFIF**
  - **J**PEG **F**ile **I**nterchange **F**ormat

Lossy JPEG Compression

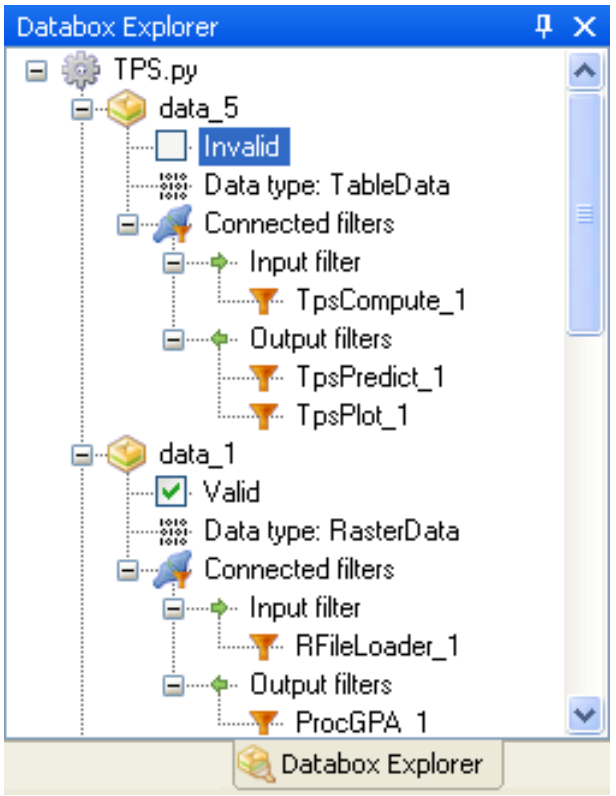


JPEG Colours

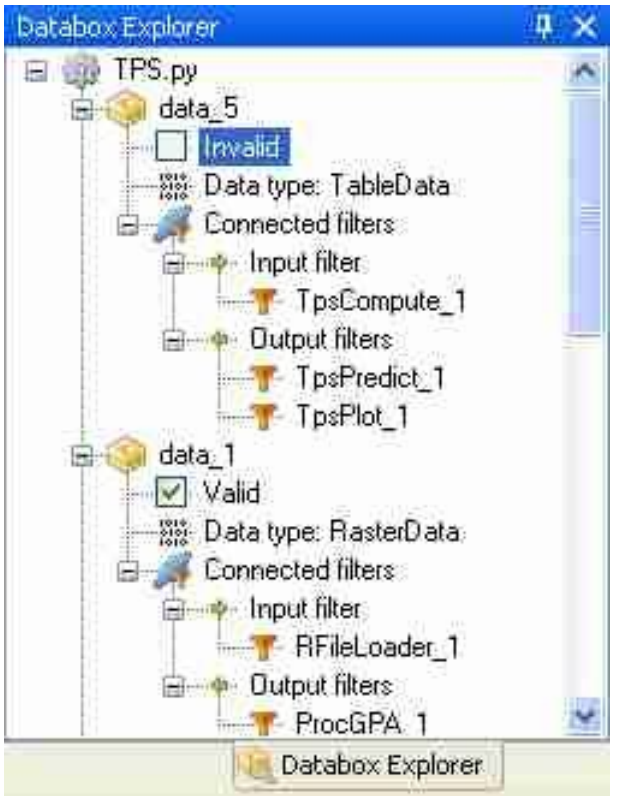
- According to standard **CCIR 601**
  - Also used in video equipment
- **8 bits** for each channel:
  - **Y**: **luminance** channel
  - C_b resp. C_r: **colour diffence channels**

$$Y = 0.299\,R + 0.587\,G + 0.114\,B$$
$$C_b = -0.1687\,R - 0.3313\,G + 0.5\,B + 128$$
$$C_r = 0.5\,R - 0.4187\,G - 0.0813\,B + 128$$

JPEG Artefacts

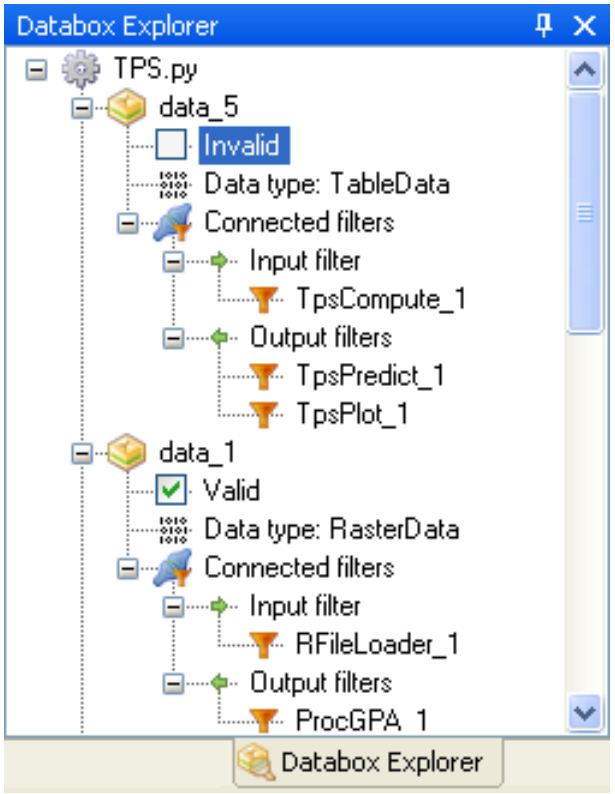


PNG (lossless)
12.3 KB



JPEG (quality 20%)
8.4 KB

Efficient compression: screenshot

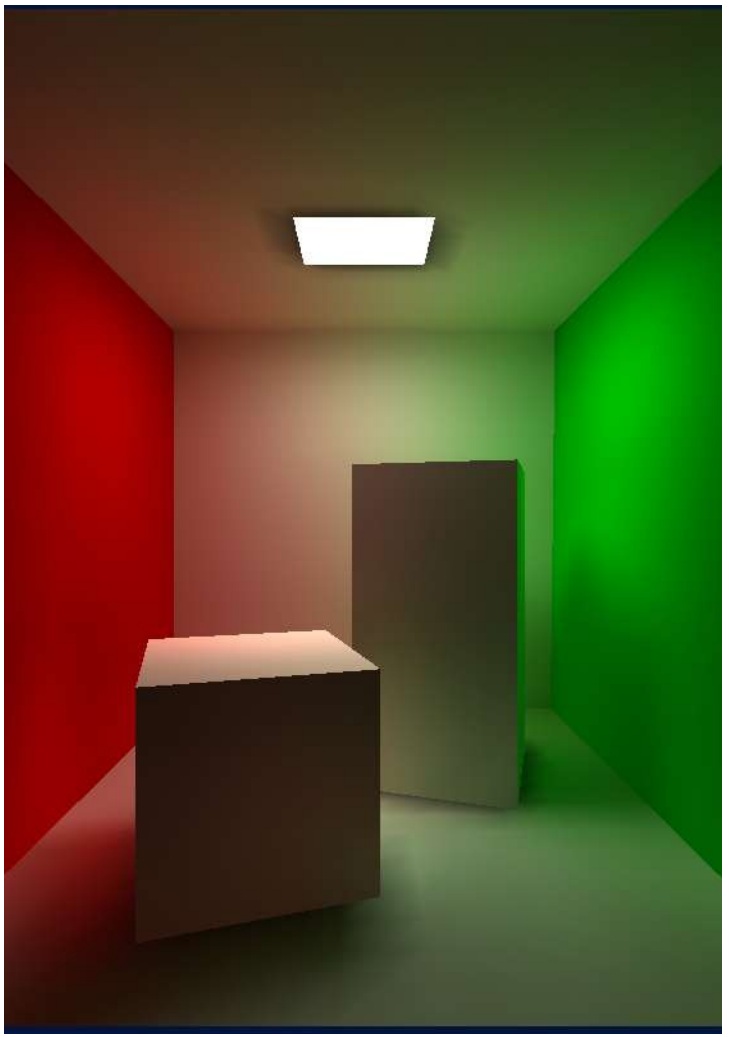


PNG (8 bit)	**5.8 KB**
JPEG (24 bit, q=20%)	8.4 KB
GIF (8 bit)	8.7 KB
PNG (24 bit)	**12.3 KB**
JPEG (24 bit, q=60%)	15.6 KB
JPEG (24 bit, q=90%)	26.5 KB
JPEG (24 bit, q=100%)	45.0 KB
PPM (24 bit)	**242.0 KB**

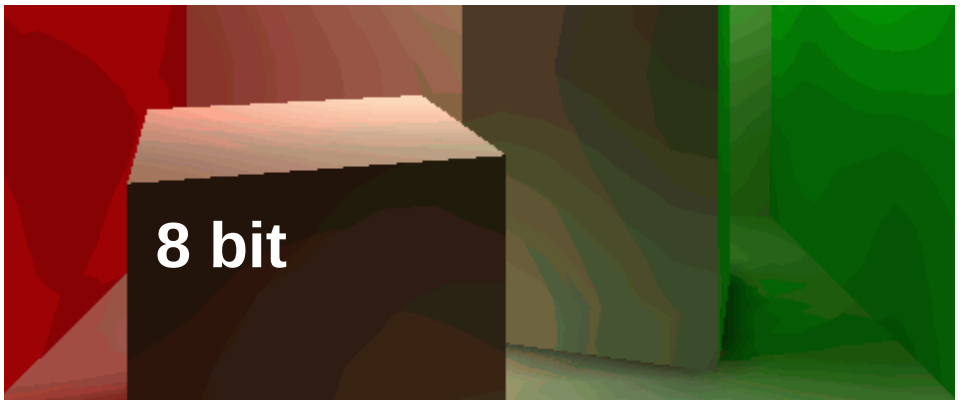
Efficient compression: photo

JPEG (24 bit, q=20%)	16 KB
JPEG (24 bit, q=60%)	**37 KB**
JPEG (24 bit, q=90%)	**87 KB**
PNG (8 bit)	158 KB
GIF (8 bit)	191 KB
JPEG (24 bit, q=100%)	245 KB
PNG (24 bit)	488 KB
PPM (24 bit)	**1052 KB**

Efficient compression: rendering



JPEG (24 bit, q=20%)	9 KB
JPEG (24 bit, q=60%)	**17 KB**
PNG (8 bit)	26 KB
JPEG (24 bit, q=90%)	**39 KB**
GIF (8 bit)	59 KB
JPEG (24 bit, q=100%)	136 KB
PNG (24 bit)	140 KB
PPM (24 bit)	**1876 KB**



End

Further information:

- **Kay D. C., Levine J. R.:** ***Graphics file formats***, MGWH, 1994
- **Wikipedia: Image_file_formats**