Computer Graphics Charles University

Colour Reproduction

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

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

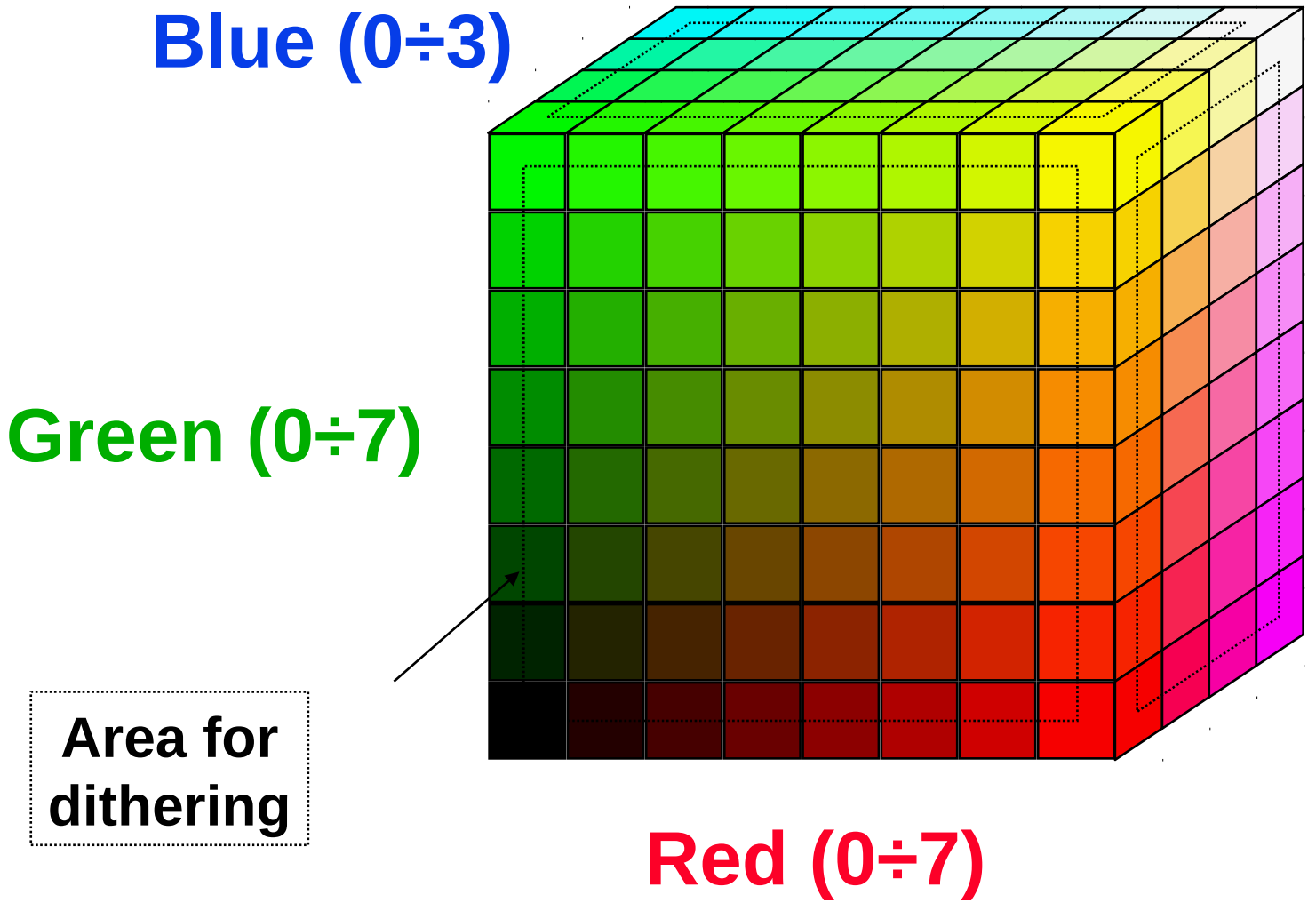
Colour Capabilities of HW

- „**True-color**” vs. „**pseudo true-color**”
  - Output components : **RGB**, **CMY(K)**
  - At least 5 bits per pixel (typically 8)
  - Displays: **15**, **16** (5-6-5), **24-bit colour**
  - Enlargement of the colour space: dithering
- Devices with a **colour palette** („colormap”)
  - Fixed or choosable palette
  - Number of colours: **16 - 4096** (usually **256**)
  - Reduction in number of colours („color quantization”)

Colour Reproduction via Palette

- Convert colours to **greyscale**
  - **Y** component (**0.2989 R + 0.5866 G + 0.1144 B**)
- **Universal colour palette** + dithering
  - E.g. **3-3-2 palette** (256 colours), 6-7-6 (252 colours)
  - Matrix, error diffusion dithering
- **Adapted colour palette** (+ dithering)
  - Palette optimised for one concrete image
  - Palette construction algorithms: **„top-down”** (Heckbert) a **„bottom-up”** (cluster analysis)

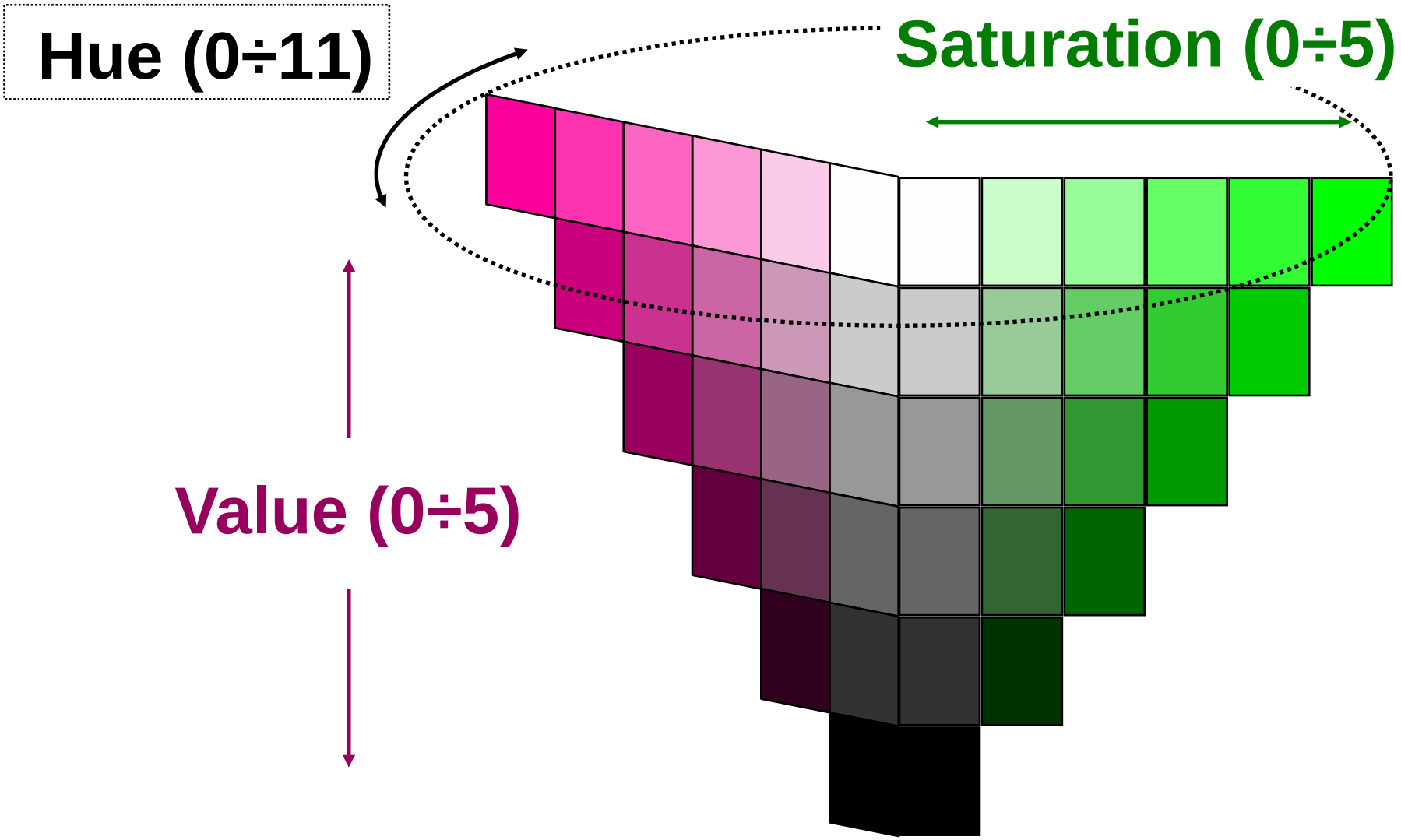
Universal „3-3-2 palette”



Universal Palettes

- **„3-3-2”**: 8×8×4 colours (256 colours)
  - Easy to convert (without multiplication)
- **„6×7×6”**: 6×7×6 colours (252 colours)
  - Uniform coverage of RGB space
- **„7×12×3”**: 7×12×3 colours (252 colours)
  - Takes the sensitivity of the human eye into account
- Palettes for **other colour systems**
  - E.g. 12×(1+2+3+4+5+6) for **HSV** (186 colours)

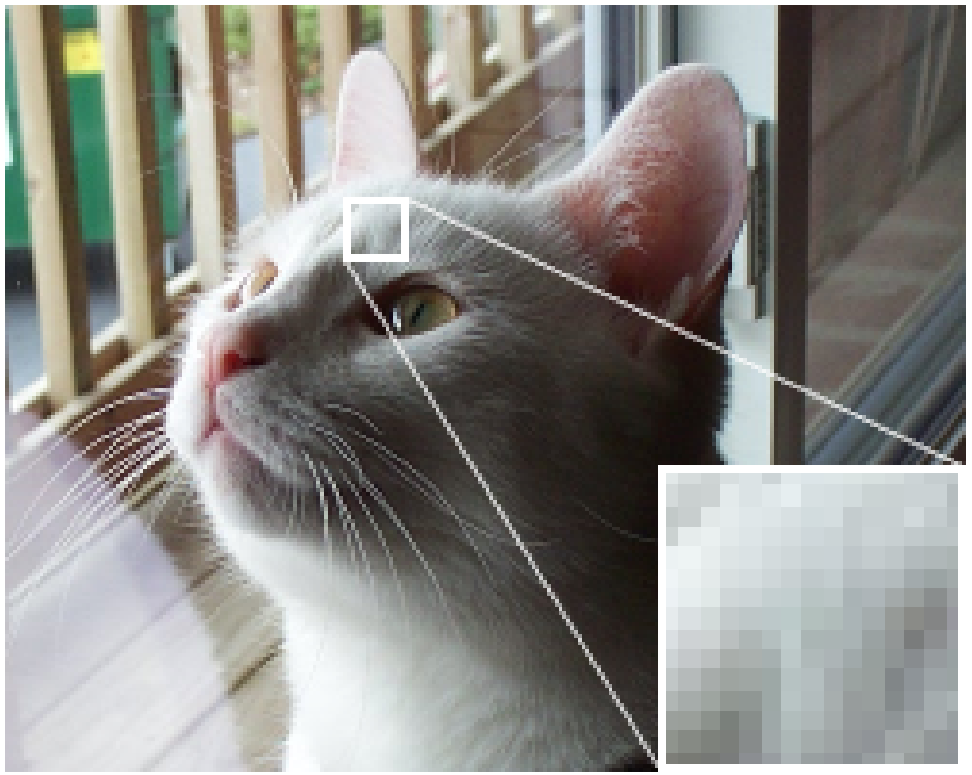
Universal Palette for **HSV**

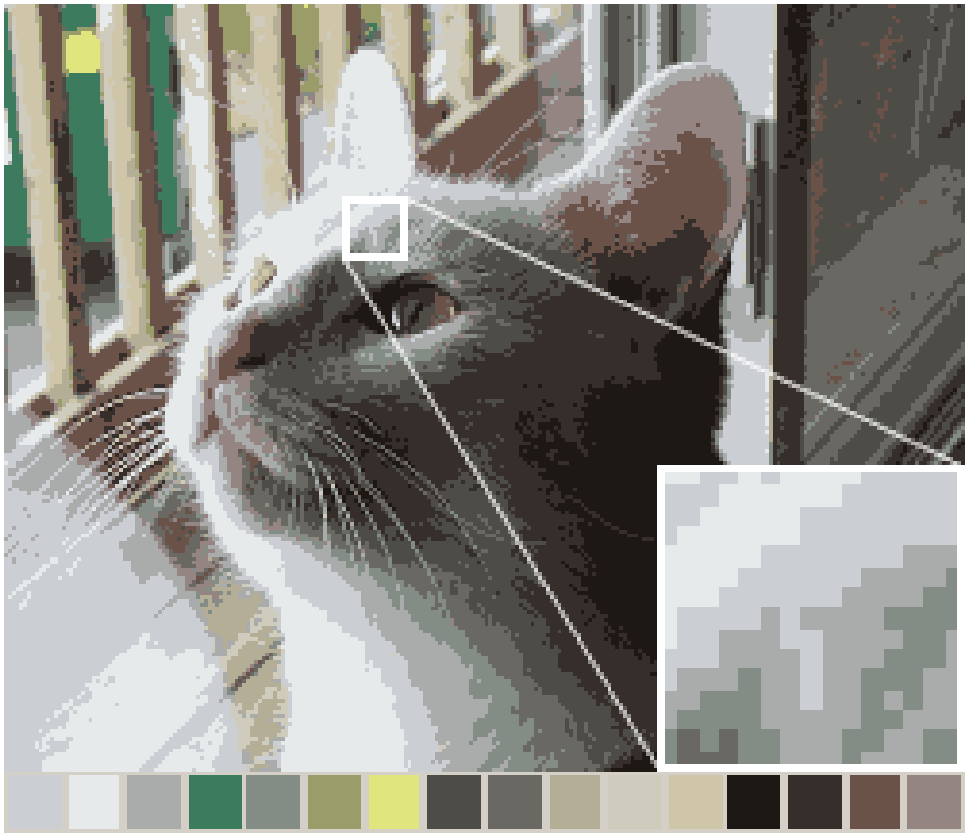


Construction of Adapted Palettes

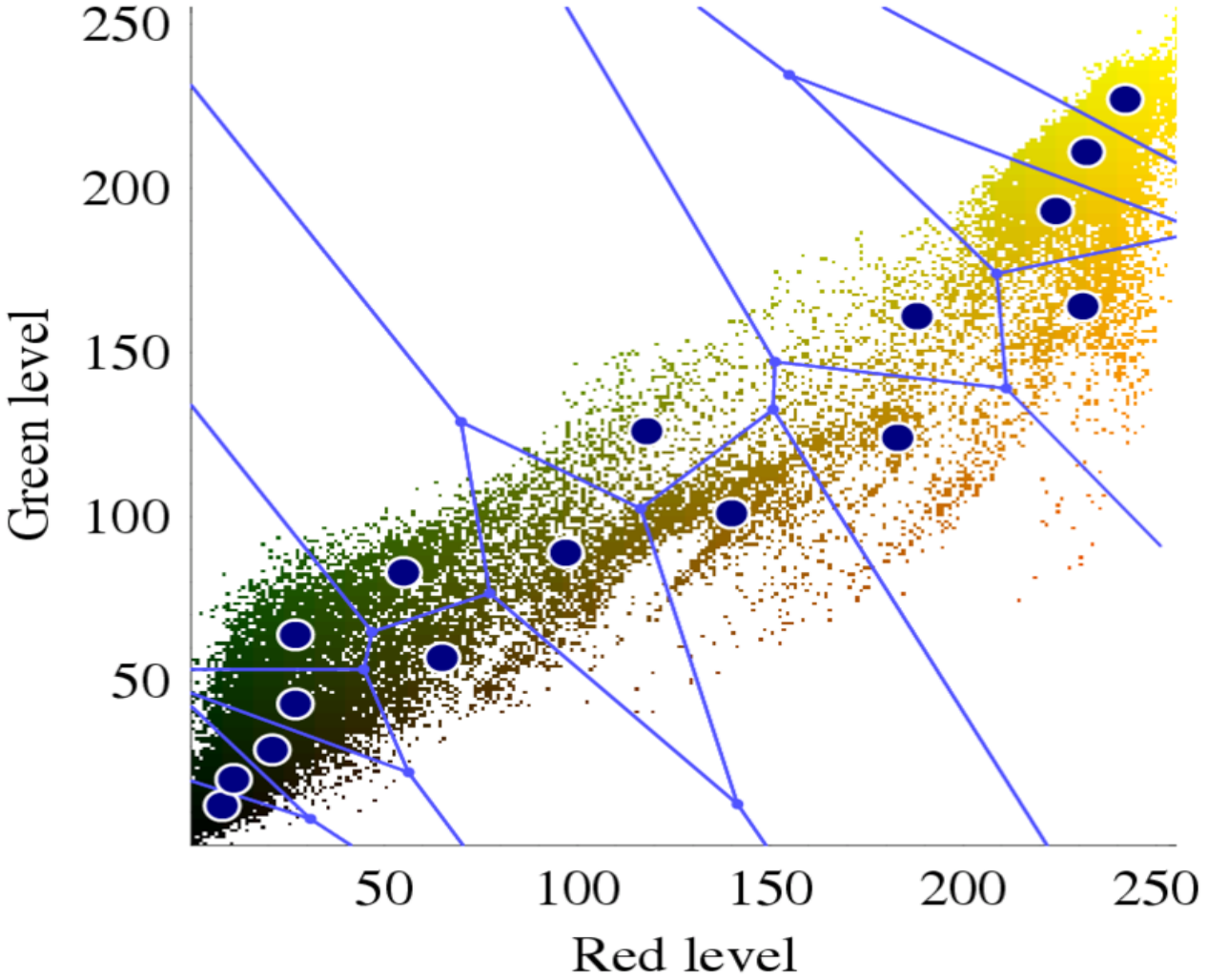
- Special palette derived for the display of **one particular image**
  - Calculating this is a pre-process, and can take long
- **„Top down”** construction
  - Splitting the space of used colours until the desired number of group colours has been reached (e.g. 256)
- **„Bottom up”** construction
  - Grouping of used colours, until only a suitable number of groups remains (cluster analysis)

Palette Example





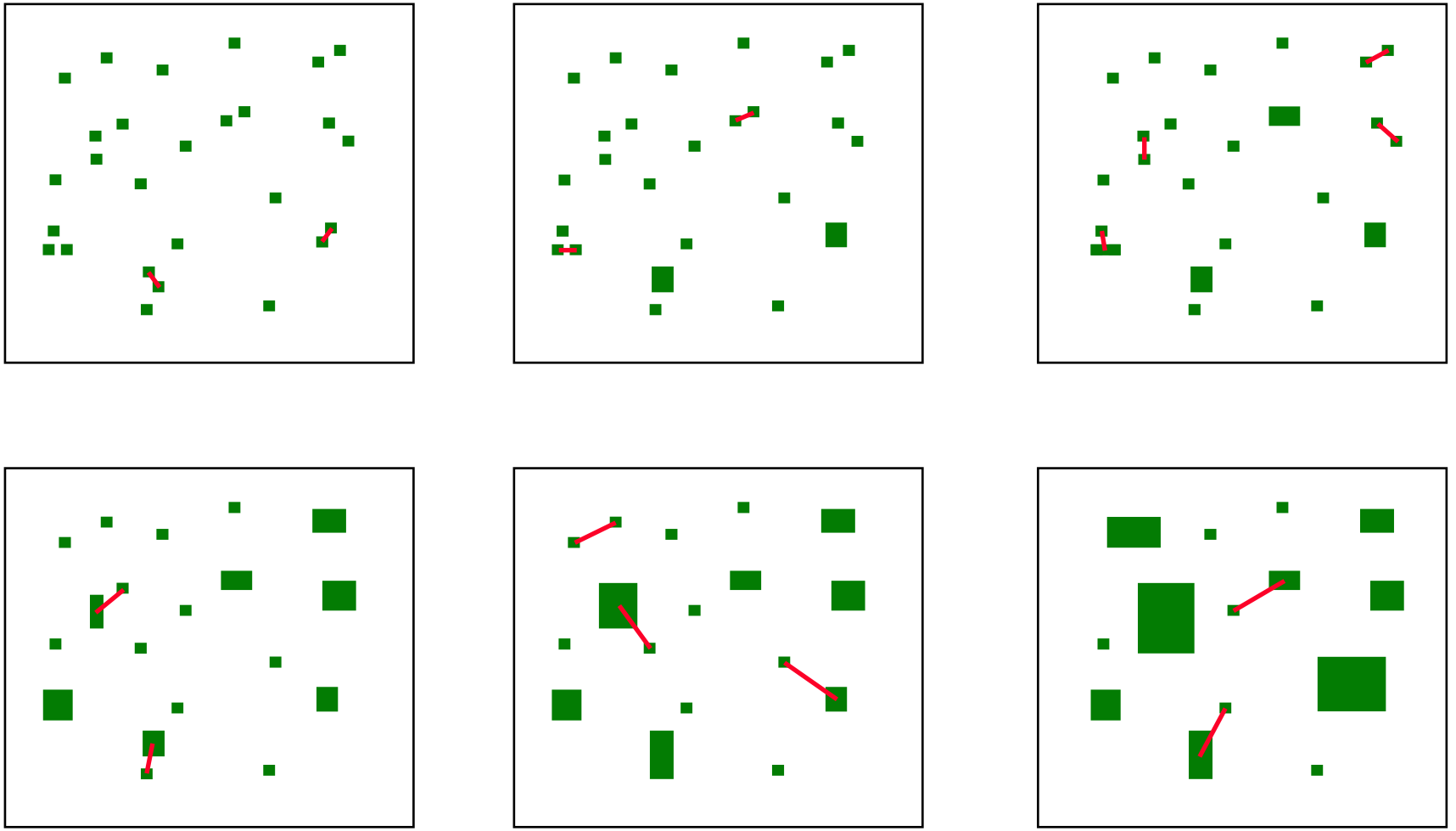
Quantisation Example



Cluster Analysis

1. Create **a colour histogram** of the image
   - A list of all colours, and their frequencies
   - At the start, each colour is a separate group
2. Find the two **closest** groups and merge them
   - Similarity criterion: **distance** $\mathbf{\min\{|C_i - C_j|\}}$, **diameter** $\mathbf{\max\{|C_i - C_j|\}}$, **scattering** $\mathbf{\mathrm{sqrt}(\Sigma(C_i - \varnothing C)^2/n)}$
3. Repeat step 2 until target number of groups **N** has been reached (e.g. 256)
   - Potentially, a large number of steps is necessary!

Computation Sequence



Octree Algorithm

- Saves **memory** and **calculation time**
  - Faster search for the closest colour groups
  - The price is a significant loss of quality!

1. The **first N different colours** form groups
2. The rest of the image is loaded, and for each pixel with a new colour:
3. Of the **N+1 groups** the two closest are merged
   - Algorithm is not symmetrical (results vary with starting point)

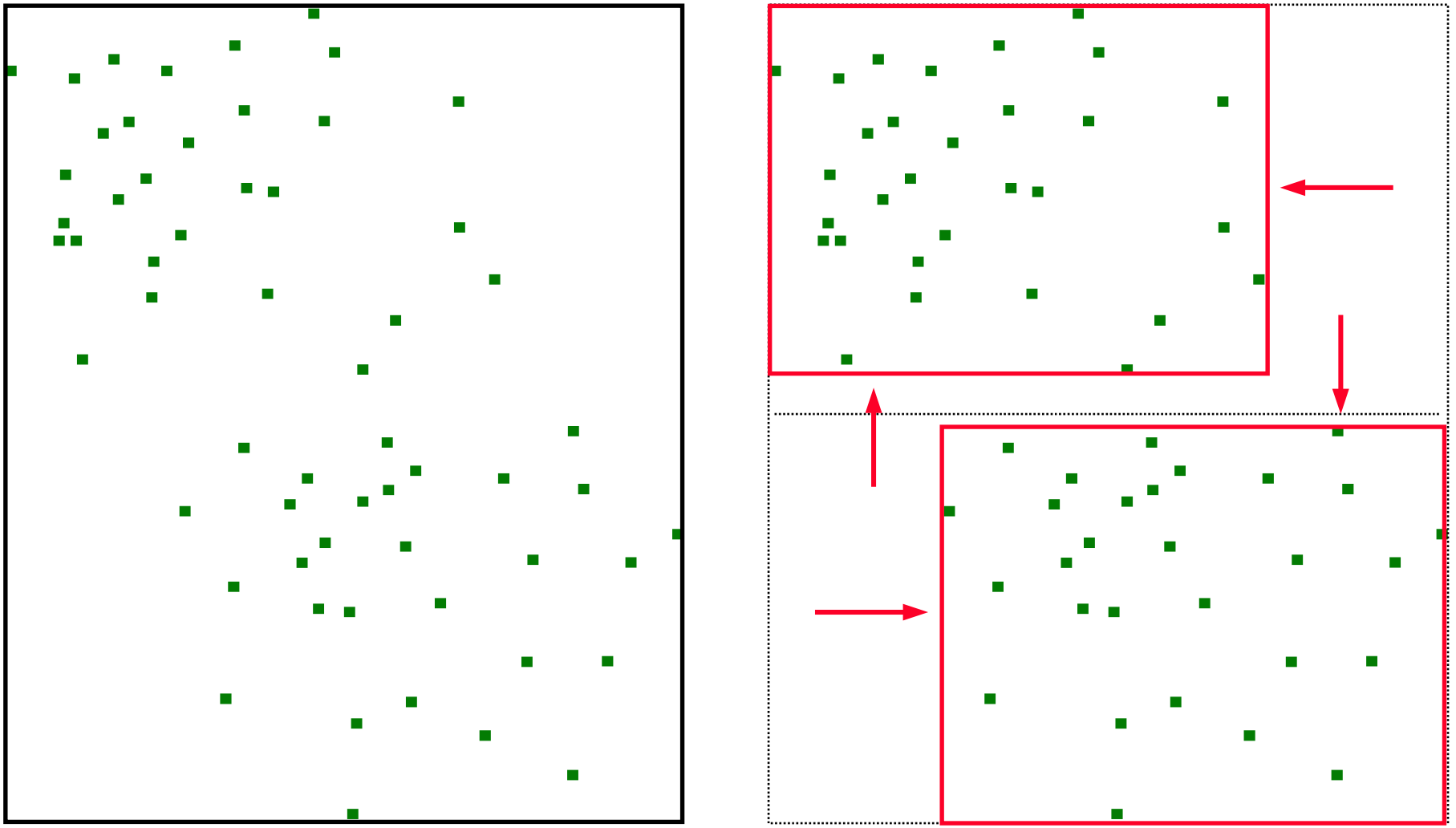
Heckberts Algorithm („median cut”)

❶ Create a **colour histogram** of the image
- In the beginning, all colours are in a single group (convex hull)

❷ Select the **„largest” group** of colours, and split it into two
- Various methods for selecting and splitting a colour group

❸ Step ❷ is repeated until a target number of groups **N** is reached (e.g. 256)
- For the final result, dithering is used

Group Splitting

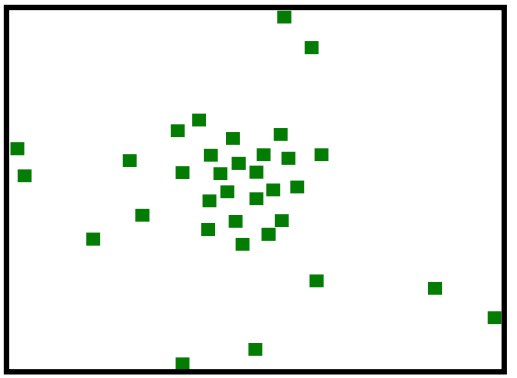


Criteria for Colour Grouping

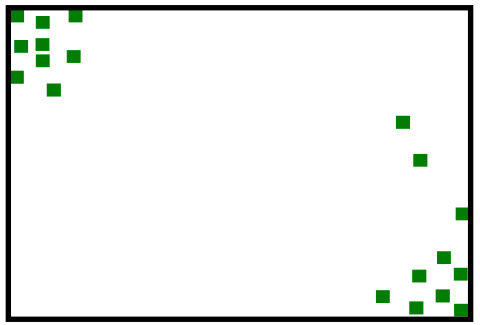
- **Size of the bounding box** (longest edge length)
  - The longest edge is divided in half
- **Subjective weight of the bounding box**
  - Individual components are weighted perceptually
- **Number of colours** (number of pixels)
  - Split along the longest edge of the block
- **Distribution of colours** (weighted by # pixels)
  - Divide the longest edge of the average values

Subdivision Criteria

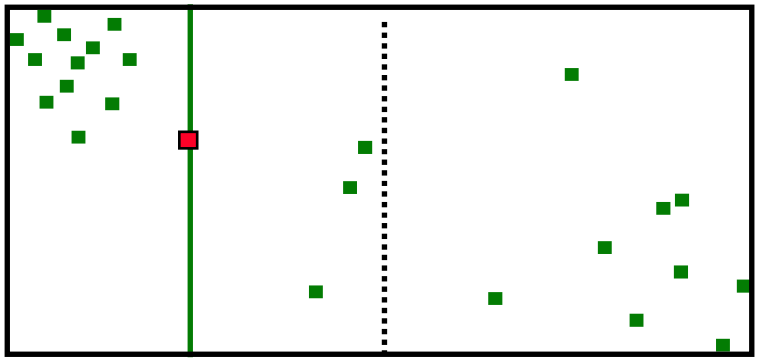
Few colours



More colours, small spread



Less colours, large spread



Median split

Implementation

- **Histogram creation** is costly in time and memory
  - Sparse histogram storage (saves memory)
  - Data structures with fast access (hash tables)
- **Colour remapping**
  - **Rounding** (search for nearest groups)
  - **Dithering** using the closest colours in the palette (error diffusion: searching for the nearest colour, and remembering the error of the last decision)

Colour Printing

- **Small number of basic colours** (2-8)
  - Very high basic resolution (thousands of dpi)
  - Universal four colour print: **CMYK**
- Every primary is **half-toned**
  - Individual half-tone rasters („screens”) tend to have a resolution of **60 ÷ 480 lpi** („lines per inch“)
  - Screens with square, elliptical, circular and special dots („Monet”, random raster, ..) are used.

Duotone Image

Raster Superposition

- All half-tone rasters are **rotated** against each other
  - This prevents, or reduces, interference patterns
  - Classical angles for **CMYK** printing: **0°**, **15°**, **45°**, **75°** („Offset angles”)
  - Another classical set: **7.5°**, **22.5°**, **37.5°**, **52.5°**, **67.5°**, **82.5°** („Flexo angles”)
  - Angles with rational numbers are preferrable for implementation reasons

End

Further information:

- **Jiří Žára a kol.**: ***Počítačová grafika***, principy a algoritmy, 335-342