



# Raster Graphics

© 1995-2016 Josef Pelikán & Alexander Wilkie  
CGG MFF UK Praha

[pepca@cgg.mff.cuni.cz](mailto:pepca@cgg.mff.cuni.cz)

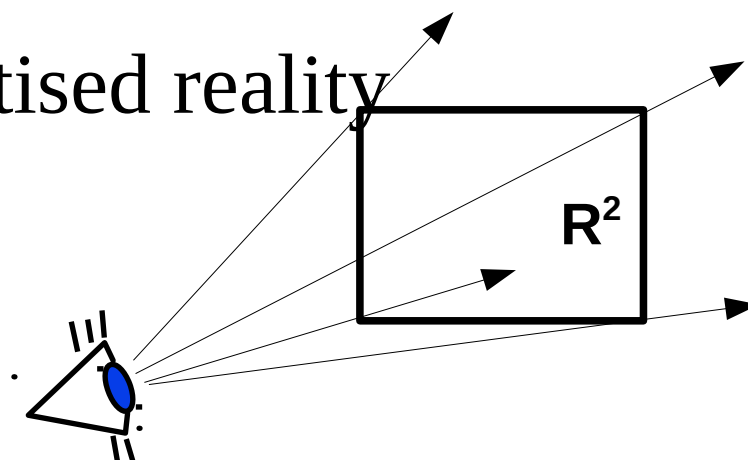
<http://cgg.mff.cuni.cz/~pepca/>



# 2D Image function

- ◆ „**Window**“ into non-discretised reality

- Mapping  $\mathbf{R}^2 \rightarrow$  „colour”
- Infinitely scalable image



- ◆ **Discretisation of images**

- Sampling the plane in a regular grid
- Pixel array
- In practice: camera sensors
- Second discretisation – pixel values (more about this later)



# Raster vs. Vector Approach

## ◆ Raster approach:

- Individual pixels are directly addressed
- Data is resolution-dependent  
(re-scaling to larger sizes is problematic)

## ◆ Vector approach:

- Complex objects are directly displayed  
(lines, curves, fonts, surfaces)
- Data is not resolution-dependent  
(it can be directly scaled to the output device)

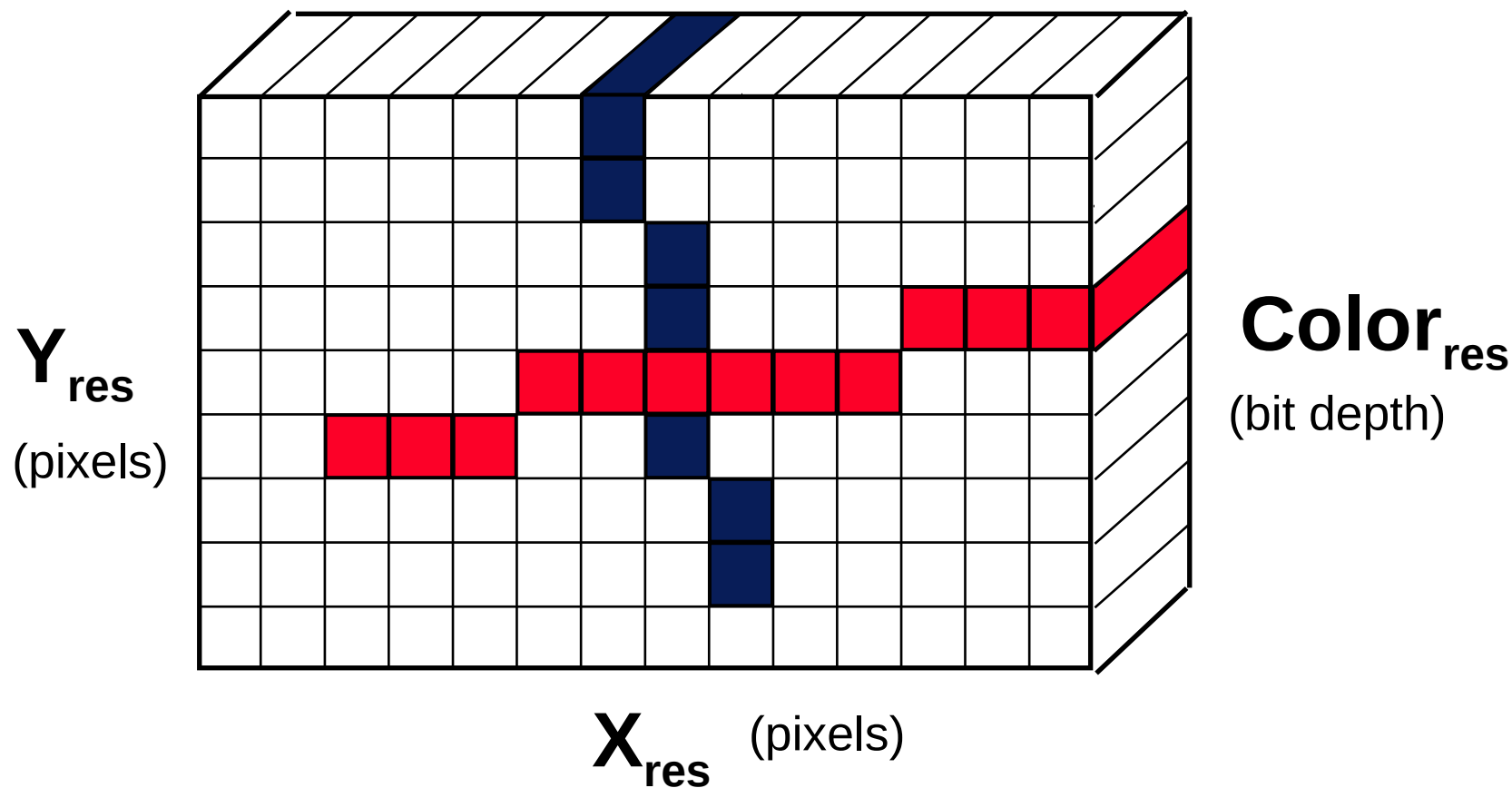


# Graphical Output

- According to **output technology**:
  - Vector output (outdated displays, plotters)
  - Raster output (displays, printers)
  
- According to **communication protocol**:
  - Vector devices (GPUs, plotters, PostScript<sup>®</sup>)
  - Raster devices (normal video cards, printers in direct mode)



# Raster Images



Example: **640×480×8** bits, **1600×1200×24** bits



# Pixel Formats

## ➤ **Integer formats**

- Older (classical) approach
- Usually 8 bits,  $3 \times 8$  bits or  $4 \times 8$  bits
- Quantisation artefacts!

## ➤ **Floating point formats**

- HDR graphics („High Dynamic Range“)
- Usually  $3 \times$  float (96bit) or  $3 \times$  half (48bit)
- No problems with loss of accuracy
- But display issues!



# Vector Displays (outdated!)

- ✦ A vector display receives **drawing commands** and stores them in memory („display list”)
  - **MoveTo(x,y), LineTo(x,y), Circle(x,y,r), ..**
- ✦ The commands in memory are **cyclically re-drawn**
  - Large numbers of commands lead to blinking
  - Cathode ray tubes with long-glowing phosphor needed
- ✦ Lines are **not jagged** (like on a raster display)

# The SVG Vector Format



## ✦ W3C standard

- All common web browsers support it (HTML5)
- Support for animation
- User-defined coordinate systems, 2D transformations, clipping...

## ✦ Based on XML syntax

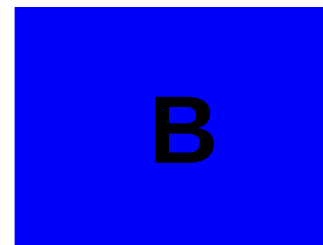
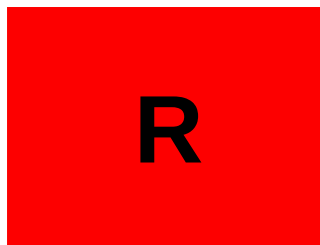


```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100">
  <path d="M30,1h40l29,29v40l-29,29h-40l-29-29v-40z" stroke="#000" fill="none"/>
  <path d="M31,3h38l28,28v38l-28,28h-38l-28-28v-38z" fill="#a23"/>
  <text x="50" y="68" font-size="48" fill="#FFF" text-anchor="middle"><![CDATA[410]]>
</text>
</svg>
```



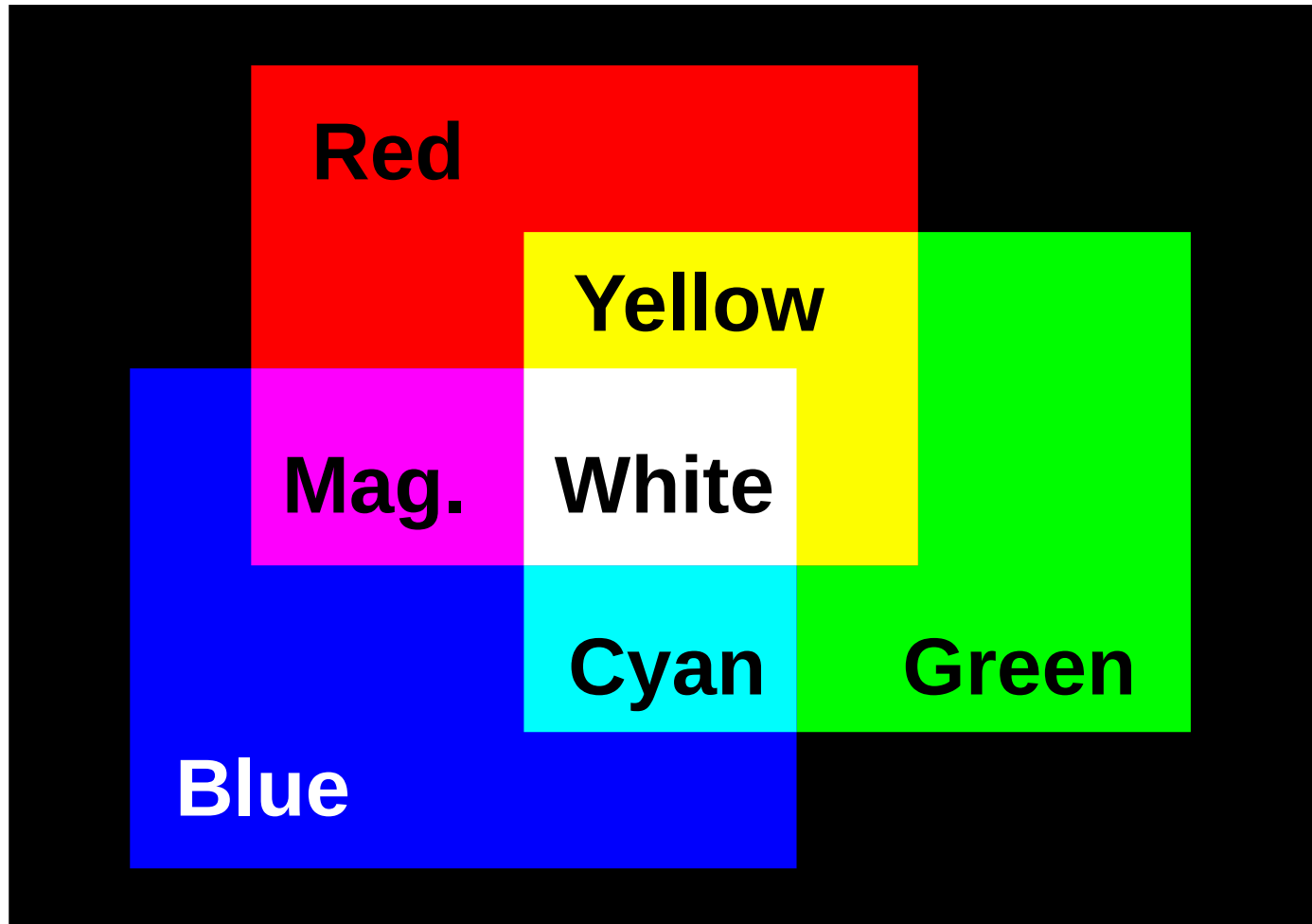


# The RGB Colour System



- ◆ **Basic colour components:** red, green, blue
  - Advantageous for active displays (monitors)
  - Also well-suited for the human visual system
- ◆ **Additive colour mixing**
  - Black background („zero colour“, inactive display)
  - White is the composition of all three primaries

# Additive Colour Mixture



# End



Further information:

- ➡ **J. Foley, A. van Dam, S. Feiner, J. Hughes:**  
***Computer Graphics, Principles and Practice*, 8-15,  
145-199**