Multimedia-Systems: Image & Graphics

Prof. Dr.-Ing. Ralf Steinmetz
Prof. Dr. Max Mühlhäuser

MM: TU Darmstadt - Darmstadt University of Technology,
    Dept. of Computer Science
TK - Telecooperation, Tel. +49 6151 16-3709,
    Alexanderstr. 6, D-64283 Darmstadt, Germany, max@informatik.tu-darmstadt.de Fax. +49 6151 16-3052

RS: TU Darmstadt - Darmstadt University of Technology,
    Dept. of Electrical Engineering and Information Technology, Dept. of Computer Science
KOM - Industrial Process and System Communications, Tel. +49 6151 166151,
    Merckstr. 25, D-64283 Darmstadt, Germany, Ralf.Steinmetz@KOM.tu-darmstadt.de Fax. +49 6151 166152
GMD - German National Research Center for Information Technology
httc - Hessian Telemedia Technology Competence-Center e.V
Scope

Usage
- Learning & Teaching
- Design
- User Interfaces

Services
- Content Processing
- Documents
- Security
- Synchronization
- Group Communications

Systems
- Databases
- Programming
- Media-Server
- Operating Systems
- Communications
- Opt. Memories
- Quality of Service
- Networks

Basics
- Computer Architectures
- Image & Graphics
- Compression
- Animation
- Video
- Audio
Contents

1. Images and Graphics
2. Coding of Images
3. Analysis of Images
4. Output of Image and Graphics: e.g. Dithering
1. Images and Graphics

Digital Image:
- $N$ rows and $M$ columns containing
  - $N \times M$ picture elements (Pixels)
- Continuous function defining a rectangular view of the real world

Graphics:
- Primitives (lines, circles, ...) and
- Attributes (style, color, ...)

- above to be considered "object graphics"
- contrast: "pixel graphics": cf. digital image

- pixel graphics into object graphics: "understanding" (tough research)
- object graphics into pixel graphics: "rendering" (display)
Input: Images and Graphics

Recording of real world images:
- Projection of real world to image plane following the central projection equation:

\[ r = F \times \frac{W_1}{W_3}; \quad s = F \times \frac{W_2}{W_3} \]

Generation of Graphics by e.g. conversion
- from low-level images (pixels) at display time
- to high-level graphics (primitives and attributes)
2. Coding of Images

Picture Elements: Pixel

- Color,
- gray-value images and
- binary images (e.g., values 1 for black, 0 for white)

Example

- gray-value images contain different number of brightness levels:

  - 2 levels
  - 4 levels
  - 256 levels
Image Formats

Capturing / Recording format:
- **Spatial resolution** [pixel x pixel]
- **Planes**
- **Color coding** [bits/pixel]

Storage format:
- **2-dimensional matrix representing pixels**
- **Example**:
  - Bitmap matrix containing binary values
- **Color image: four important approaches**
  - 3 numbers representing intensities for red, green, blue (RGB) or
    "true color" if numbers fine-grained enough, e.g., 8-bit; very common
  - 3 numbers representing pointers to color table (1 color per entry - R,G, or B)
  - 1 number as pointer (index) to color table: "color lookup table CLUT"
    (most common, apart from true color)
    - e.g., 8 bit-pointer: 256 colors possible
    - e.g., CLUT-entry 3 Bytes: one for each main color (256-out-of-16Mio)
  - index to arbitrary data structures representing colors
Postscript

History:
• Developed 1984 by Adobe
• First time fonts became important to the general public

Functionality:
• Integration of high-quality text, graphics and images
• Programming language
  • full-fledged
  • with variables, control structures and files

Postscript Level-1:
• Earliest version developed in 1980ies
• Scalable font concept (in contrast to fixed-size fonts available until then)
• Problem: no patterns available to fill edges of letters resulting in medium quality

Postscript Level-2:
• High-quality pattern filling
• Greater number of graphics primitives
• Color concept both device-dependent or device-independent

Follow-up: Adobe’s Portable Document Format (PDF)
Graphics Interchange Format (GIF)

History:
• Developed by CompuServe

Goal
• to exchange images platform-independently

Main components:
• header (identification and version)
• application (creator software of image)
• data
• trailer (end of GIF-data)

Compression: Lempel-Ziv-Algorithm
• localizes bit patterns which occur repeatedly
• variable length-coding of repeated patterns

Comment
• Well-suited for image sequences
  (as more than one image can be part of a GIF-file)
Tagged Image File Format (TIFF)

History
- Developed by Aldus Co. and Microsoft

Functionality
- to support platform-independent exchange of images
- Wide distribution as well-suited for scanners and fax devices
- Main components:
  - baseline (constraints with regard to displaying devices)
  - extensions (constraints with regard to special devices)

Compression
- Various/many color models
  - binary images
  - gray-value images
  - RGB
  - CIE (perception-based colors)
- various algorithms,
  - like Lempel-Ziv, runlength encoding (also denoted as PackBits compression),
  - FAX groups 3 and 4 and JPEG,
  - huffman encoding
X11-Bitmap (XBM)

- example of the UNIX-world
- monochrome images,
- no compression as pixel are coded as 8-bit ASCII

Example

```c
#define xbm_image_width 8
#define xbm_image_height 8
static unsigned char xbm_image_bits [] = {
    0x01,
    0x02,
    0x04,
    0x08,
    0x10,
    0x20,
    0x40
};
```
X11-Pixmap (XPM)

- example of the UNIX-world
- color images

Some details
- Hot spot:
  - identifies cursor position where mouse selection can be applied
- Coded as string array (header and list of strings)
- Color substituted by ASCII value
- Transparency color: symbols ("s None")

Example
- (hot spot in row 4, column 1)

```c
static char *demo_xpm[] = {
"8 8 1 4",
" s None c None",
"X c black",
"X , , X",
"X , , X",
" X , , X",
" X , , X",
" X , , X",
" X , , X",
" X , , X",
};
```
3. Analysis of Images

as part of content processing

covers

- Image improvement
- Pattern detection and recognition
  - from segmentation to Optical Character Recognition
- Scene analysis
- Computer vision
- ...

• ==> later chapters / MM II - lectures
4. Output of Image and Graphics: e.g. Dithering

Problem:
- Image quality using binary images (black and white)

Solution
- Machine representation: halftoning
- Example:
  - area of 2x2 pixel using 2 colors
  - 5 different gray values possible:

- Main application: laser printer

Problem:
- staircase appearance of lines / curves (due to processing raster, pixels)

Solution
- Anti-aliasing: use gray (or mix-color) values for exposed margin-pixels (which reach into background)
- Main application: monitor
- May yield blurring effect, e.g., for small fonts
Consideration: Pixel graphics vs. graphics objects

Application software may be based on either pixel or object graphics
• consider drawing tool
• reckon: differences? ... wrt. (e.g.,):
  capabilities, "image compliance", bandwidth, distributed version...

Graphics subsystem .... may be based on either?
• "high-level" API (OpenGL, DirectX, ...): objects / vectors
• "low-level" API (Microsoft GDI, X-protocol): mostly "drawing primitives"
• "frame buffer": pixels at last
• reckon about so-called "application-sharing" software
  • application domains:
    a) teleworking, b) remote diag (shared input), c) teleteaching (1-to-many)
  • pixel-based and graphics-based versions exist
  • smart software combines both ... reckon why