3. Images and Graphics

Contents

3. Images and Graphics

3.1 Digital Image Representation

An digital image is a spatial representation of an object

(2D, 3D scene or another image - real or virtual).

Digital Image

Let $I, J, K \subseteq Z$ be a finite interval. Let $G \subset N_0$ with $|G| < \infty$ the grey scale level (intensity value of an picture element - *pixel*) of the image.

(1) A 2D-Image is a function: $f : I \times J \rightarrow G$

(2) A 3D-Image is a function: $f : I \times J \times K \rightarrow G$

(3) If $G = \{0,1\}$ the function is a binary / bit image, otherwise a pixel image.

The Resolution depends on the granularity of the grid with pixels equally spaced.
High resolution images are often very large.

Example:

To display a 525-line television picture (NTSC) without noticeable degradation with a Video Graphics Array (VGA) video controller 640x480 pixels and 256 discrete grey levels gives an array of 307.200 8-bit numbers and a total of 2.457.600 bits.

3. Images and Graphics

3.2 Image Formats / Captured Image Format

Captured Image Format is specified by:

spatial resolution (pixels x pixels) and *color encoding* (bits per pixel)

Examples:

VideoPix™ / SunVideo™ card:

spatial resolution:	320 x 240 pixels
color encoding:	1-bit (binary image), 8-bit (color or greyscale)
	24-bit (color-RGB)

SunVideo™ card (capture and compression card with 30* fps (frames per second)):

CellB	30 fps	MPEG1 IP frames	17 fps
JPEG	30 fps	Capture YUV	30 fps
MPEG1 I frames	30 fps	Capture RGB-8/24	30 / 12 fps

* At least 25 frames per second are necessary if continuous motion is to be presented. However, it is possible to “cheat” a little and to work with smaller frame rates.

3. Images and Graphics

3.2 Image Formats / Stored Image Format

A **Stored image format** is a 2-dimensional array of values representing the image in a **bitmap** or **pixmap**, respectively.

The data of a bitmap is a binary digit, data in a pixmap may be a collection of:

- 3 numbers representing the intensities of red, green, and blue components of the color
- 3 numbers representing the indices to tables of red, green and blue intensities single numbers as index to a table of color triples
- single numbers as index to any other data structures that represents a color / -system
- 4 or 5 spectral samples for each color

3. Images and Graphics

3.2 Image Formats / Stored Image Format

Each pixel may have additional information i.e. the normal vector to the surface.

Further properties of images: width, height, depth, size, version, etc.

[In many cases is data compression recommended or mandatory]

Formats:

- RIFF (Resource Interchange File Format) including formats for bitmaps, vector drawings, animation, audio and video (extension is BRIM)
- GIF (Graphical Interchange Format), X11 Bitmap, Sun Rasterfiles, PostScript, IRIS, JPEG, TIFF (Tagged Image File Format) etc.

Graphics image formats are specified through:

– *graphics primitives*: lines, rectangles, circles, ellipses, text strings (2D), polyhedron (3D)

– *attributes*: line style, line width, color affect.

Graphics primitives and their attributes represent a higher level of an image representation. The graphics package determines which primitives are supported.

Advantages:

+ Reduction of the graphical image data
+ Easier manipulation of graphical images.

Disadvantage:

– Additional conversion step from graphical primitives and attributes to its pixel representation

Formats:

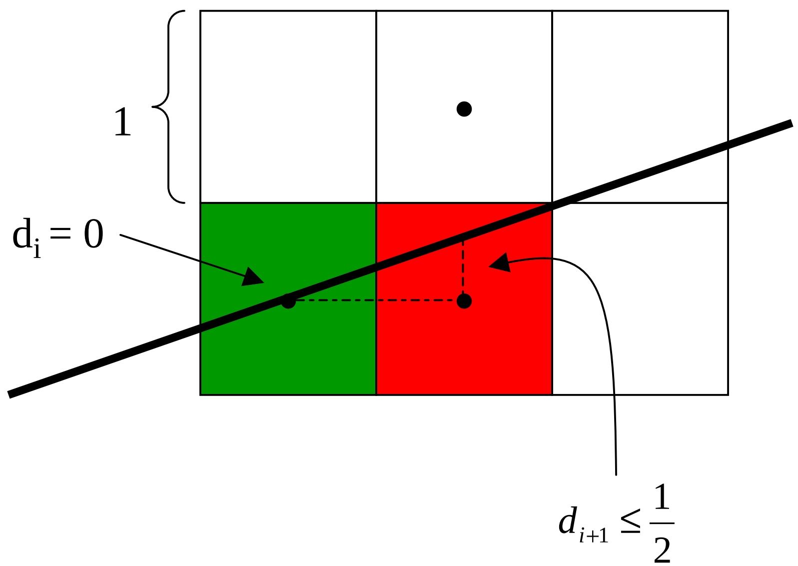
· SRGP (Simple Raster Graphics Package), one way conversion to bit-/pixmap
· PHICS (Programmer's Hierarchical Interactive Graphics Systems) and
· GKS (Graphical Kernel System) only image representation is in pixmap

3. Images and Graphics

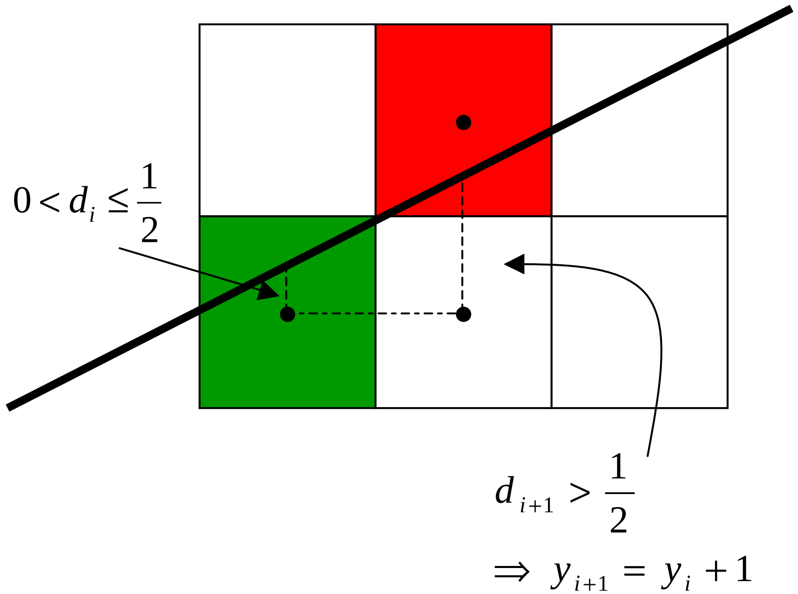
3.2 Image Formats / Object Representation Bresenham Line

Construct a line from left to right with a max. gradient $0 \leq m \leq 1$, $(\leq 45°)$, all other lines are achieved through reflections.

Next pixel is the one in the east.



Next pixel is the one in the north-east.



3. Images and Graphics

3.2 Image Formats / Object Representation Bresenham Line

To determine the next pixel (x_{i+1}, y_{i+1}) with $x_{i+1} = x_i + 1$,

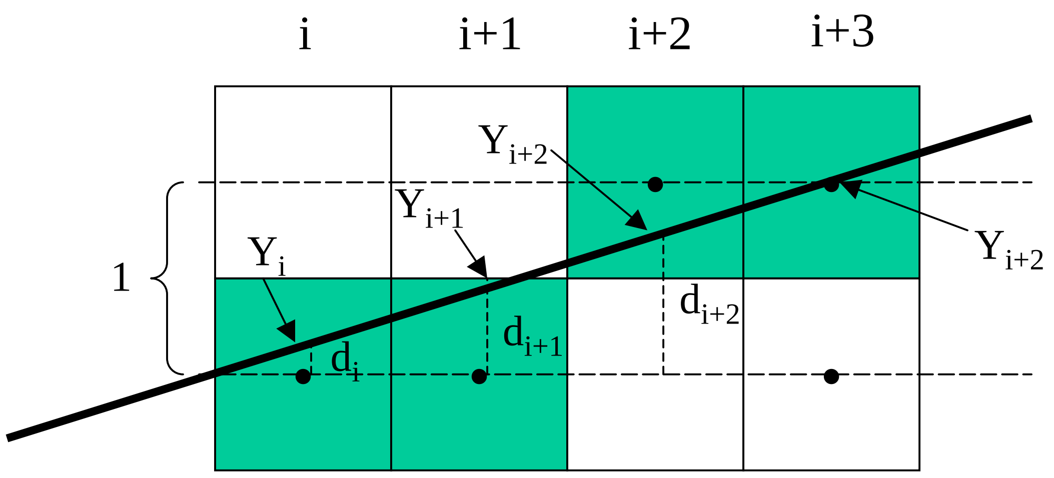
let d the control quantity in each step, calculated with:

$$d_{i+1} = \begin{cases} d_i + m & \text{if } d_i \le \frac{1}{2} \\ d_i + m - 1 & \text{if } d_i > \frac{1}{2} \end{cases} \quad \text{with } m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}, \quad d_0 = 0, \quad \text{and } y_{i+1} = \begin{cases} y_i & \text{if } d_{i+1} \le \frac{1}{2} \\ y_i + 1 & \text{if } d_{i+1} > \frac{1}{2} \end{cases}$$

Easier calculation by D_i instead of d_i as follows:

$$D_0 = (0 - \tfrac{1}{2}) 2\Delta x = -\Delta x, \quad \mathrm{M} = m 2\Delta x = 2\Delta y$$

$$D_{i+1} = \begin{cases} D_i + M & \text{if } D_i \le 0 \\ D_i + M - 2\Delta x & \text{if } D_i > 0 \end{cases} \quad \text{and} \quad y_{i+1} = \begin{cases} y_i & \text{if } D_{i+1} \le 0 \\ y_i + 1 & \text{if } D_{i+1} > 0 \end{cases}$$



$$d_{i+1} = d_i + m$$

$$d_{i+2} = d_{i+1} + m = d_i + 2m$$

$$d_{i+3} = d_{i+2} + m - 1 = d_i + 3m - 1$$

$$\left(\text{since } d_{i+2} > \frac{1}{2} \right)$$

$$y_{i+1} = y_i$$

$$y_{i+2} = y_{i+1} + 1 \left(\text{since } d_{i+2} > \frac{1}{2} \right)$$

$$y_{i+3} = y_{i+2} \left(\text{since } d_{i+3} \leq \frac{1}{2} \right)$$

Let Y_i = true value of curve,

then $Y_{i+1} = y_i + d_{i+1}$

$y_i \in \mathbb{Z}$; $Y_i \in \mathbb{R}$

On the correctness of the formulae for d_i and y_i:

$Y_{i+1} = Y_i + m;$ $\qquad$ $Y_{i+1} = y_i + d_{i+1}$

$$\Rightarrow y_i + d_{i+1} = y_{i-1} + d_i + m$$

$$\Rightarrow d_{i+1} = d_i + m + (y_{i-1} - y_i)$$

$$= \begin{cases} d_i + m & \text{if } y_i = y_{i-1} \quad \leftarrow \text{criterion: } d_i \le \frac{1}{2} \\ d_i + m - 1 & \text{if } y_i = y_{i-1} + 1 \quad \leftarrow \text{criterion: } d_i > \frac{1}{2} \end{cases}$$

$$= \begin{cases} d_i + m \quad (\text{and } y_i = y_{i-1}) & \text{if } d_i \le \frac{1}{2} \\ d_i + m - 1 \, (\text{and } y_i = y_{i-1} + 1) & \text{if } d_i > \frac{1}{2} \end{cases}$$

Changes of y_i:

$$y_i = \begin{cases} y_i\text{-}1 & \text{if } d_i \le \frac{1}{2} \\ y_{i-1}+1 & \text{if } d_i > \frac{1}{2} \end{cases} \qquad y_{i+1} = \begin{cases} y_i & \text{if } d_{i+1} \le \frac{1}{2} \\ y_i+1 & \text{if } d_{i+1} > \frac{1}{2} \end{cases}$$

3. Images and Graphics

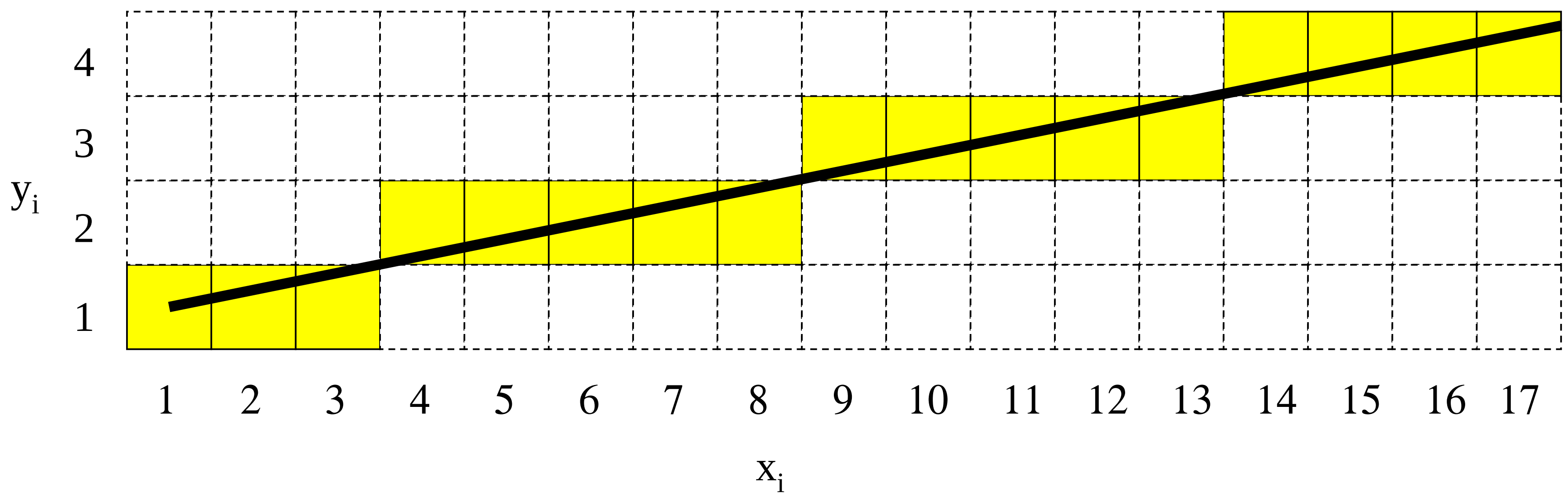
3.2 Image Formats / Object Representation Bresenham Line

Example:

$$m = 0.2 \quad \rightarrow \quad \Delta x = 1, \; \Delta y = 0.2,$$

$$M = 2\Delta y = 0.4, \qquad D_0 = -\Delta x = -1$$

x_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
y_i	0	0	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3
D_i	-1	-0,60	-0,20	0,20	-1,40	-1,00	-0,60	-0,20	0,20	-1,40	-1,00	-0,60	-0,20	0,20	-1,40	-1,00	-0,60



3. Images and Graphics

3.3 Image Synthesis

Image synthesis (generation) is an integral part of all computer graphical user interfaces and indispensable for visualising 2D, 3D and higher dimensional objects. E.g.:

Graphical User Interface (GUI): desktop windows system with icons and menu items

- Office Automation and Electronic Publishing: desktop publishing, Hypermedia systems
- Simulation and Animation for Scientific Visualisation and Entertainment

Pictures can be dynamically varied by adjusting the animation speed, portion of the total scene in view, the amount of details shown etc.

- *Motion Dynamics:* Objects are moved and enabled with respect to a stationary or also dynamic observer, e.g. flight simulator.
- *Update Dynamics:* Objects being viewed are changed in shape, color, or other properties, e.g. deformation of an in-flight aeroplane structure.

Application model

- represents data or objects to be pictured (stored in an application database)
- stores graphics image formats and connectivity relationships of the components
- should be application specific and independent of any particular display system
- converts image database representations to the graphics system format

Application program

- handles user inputs by sending commands to the graphics system describing *what* to display and *how* this objects should appear

Graphics system

- intermediary component between application programs and the display
- effects an output transformation from objects in the application model
- effects an input transformation from user actions to application
- consists of output subroutines collected in a graphics package to display images

Graphics hardware

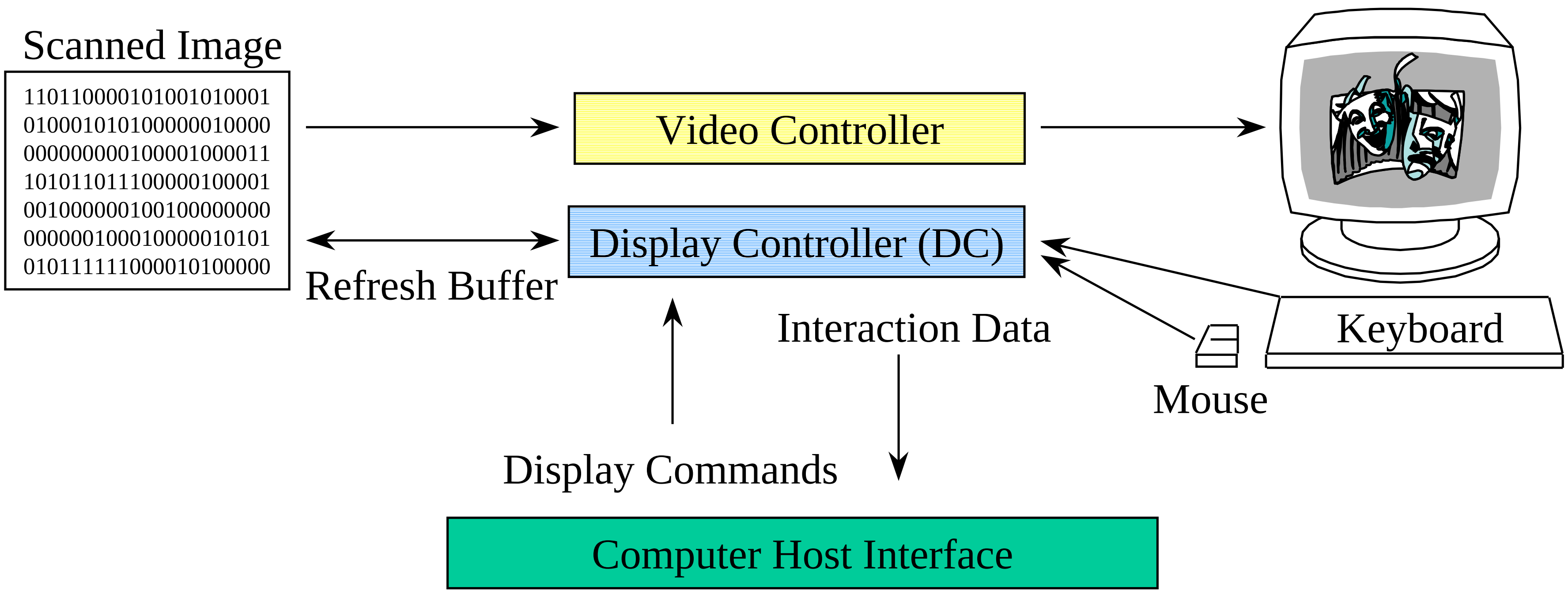
- receives input from interaction devices and outputs images to display device

3. Images and Graphics

3.4 Graphics Systems / Graphics Hardware

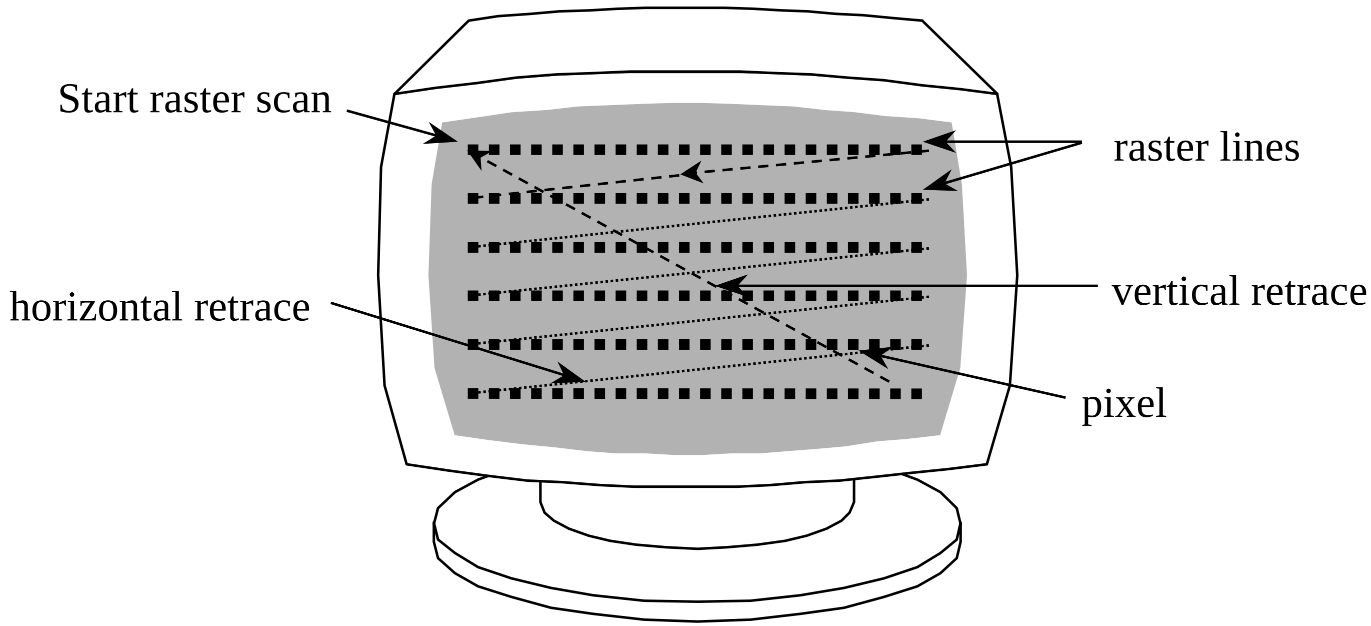
Input: mouse, keyboard, data tablet, touch-sensitive panel on the screen (2D input)
track-balls, space-balls, data glove etc. (3D and higher-dimensional input)

Output: raster display



3. Images and Graphics

3.4 Graphics Systems / Raster Display



To avoid flickering of images a 60 Hz or higher refresh rate is recommended.

At each pixel, the beam’s intensity is set to reflect the pixel’s intensity.

In color screens three beams are controlled (red, green, blue).

Monochrome displays use *achromatic light* that is determined by the quality of the light (intensity and luminance).

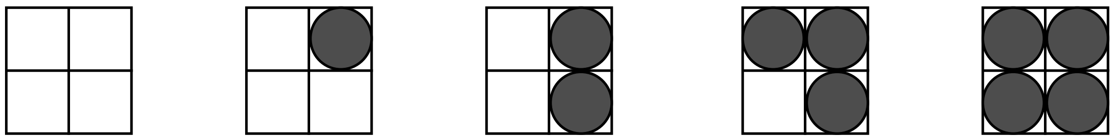
3. Images and Graphics

3.4 Graphics Systems / Dithering

How to “produce” more than two intensity levels if only two-level intensities may be represented?

Halftoning (*clustered-dot ordered dithering*) uses the human eye’s capability for *spatial integration* to produce different intensity levels, e.g. used by laser printers that are not able to display individual dots.

Example: 2 x 2 dither patterns (5 intensity levels) represented in dither matrices



Dispersed-dot ordered dithering (*monochrome dithering*) is used e.g. to extend the number of available colors of displays (at the expense of resolution).

Example:

Consider 3 bits (red, green, blue) per pixel and a 2 x 2 pattern area:

Each pattern can display 5 intensities for each color resulting in 5 x 5 x 5 = 125 colors.

3. Images and Graphics

3.5 Image Analysis

Pixel information is very often insufficient! There is no information about:

- *shape* of an object
- *orientation* of an object
- *three-dimensional interpretation* of an object

➔ Image Analysis (Image Processing) is required.
Some applications or special topics are:

- Image enhancement
- Pattern recognition
- Scene analysis
- Computer vision

Example: Traffic scenes taken by a camera installed in a car

Problems: Is there a traffic sign visible?
Which traffic sign?
Is a moving car in front of our car?
Which type of car?
Which relative speed to our speed?

3. Images and Graphics
3.5 Image Analysis

Image analysis is concerned with techniques for extracting descriptions about images:

- computation of perceived brightness and color
- partial or complete recovery of 3D data in a scene
- location of discontinuities corresponding to objects in a scene
- characterisation of the properties of uniform regions in a image

Image Enhancement

- improves image quality by eliminating noise (extraneous or missing pixels) or by enhancing contrast, i.e. X-ray images, computerized axial tomography (CAT)

Scene Analysis and Computer Vision

- deals with recognising and reconstructing 3D models of a scene from several 2D images, i.e. industrial robot sensing (relative sizes, shapes, positions, colors)

Pattern Detection and Recognition

- deals with detecting and clarifying standard patterns and finds distortions from these patterns, i.e. Optical Character Recognition (OCR)
- *static character recognition* (OCR), *dynamic recognition* (handwriting)

3. Images and Graphics
3.5 Image Analysis / Target Detection

Turning radar detector:

$\text{0000000000000000000}\underbrace{\text{111111111111111111}}_{\text{an object}}\text{0000000000000000000000000}$

Imperfect radar: some 0-1 mistakes:

000000010000110000000000111111001101111110010000000001000001110000000000000

Elimination of mistakes by **digital filtering:**

Simplest version: **symmetrical (2m-1, m) filter**

$$a_1 a_2 \ldots a_i \ldots a_n \rightarrow a'_1 a'_2 \ldots a'_i \ldots a'_n \quad \text{with} \quad a'_n = \begin{cases} 1 & \text{if} \displaystyle\sum_{i=h+m-1}^{h+m-1} a_i \geq m \\ 0 & \text{else} \end{cases}$$

Example:

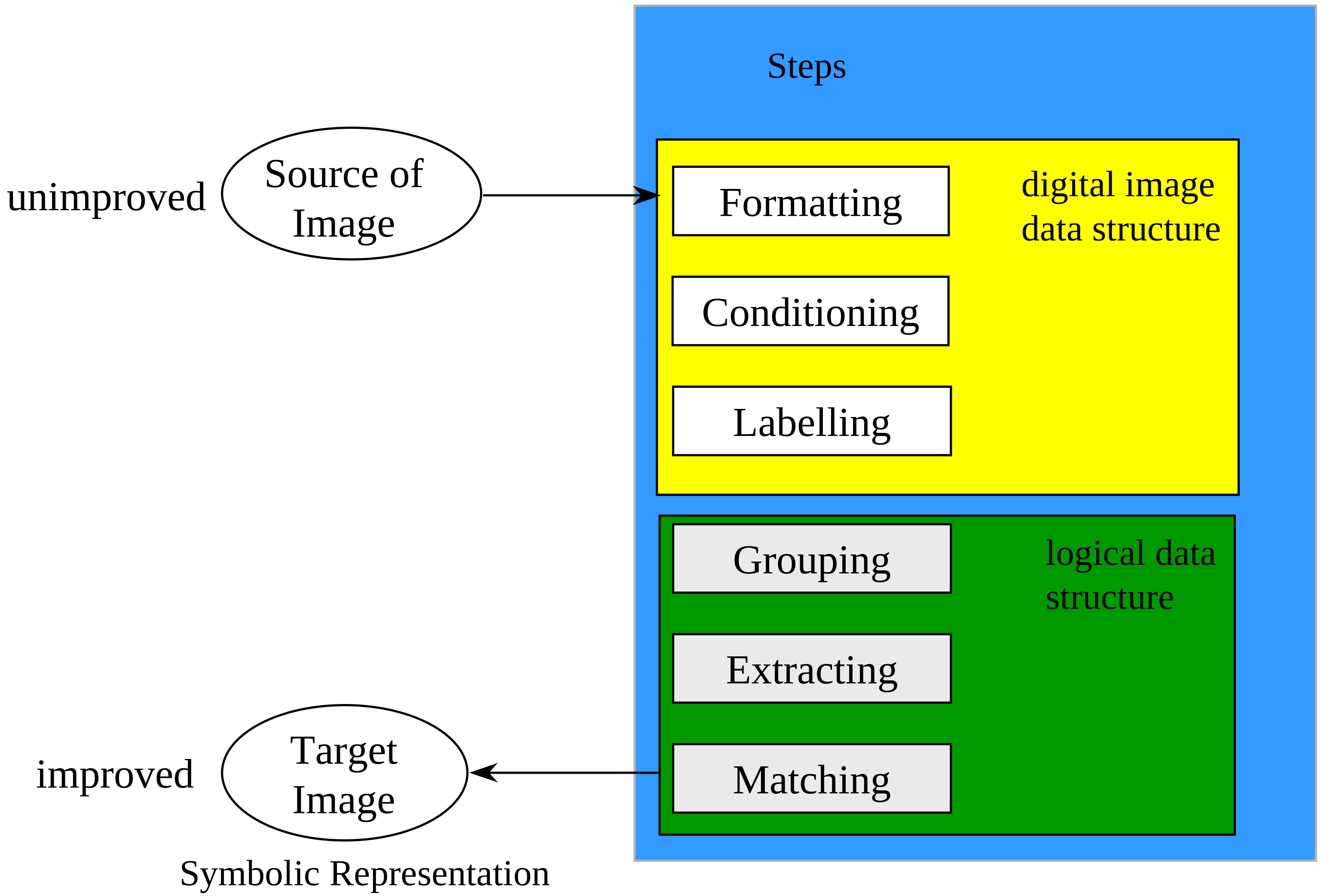
0000010000100011101101101111100010

0000000000000011111111111110000

Filtering by (5,3)-filter

3. Images and Graphics

3.5 Image Analysis / Image Recognition Steps



Formatting

- capturing of an image and transforming to a digital representation

Conditioning

- based on a model that assumes that the observed image is composed of an informative pattern modified by uninteresting variations
- estimates informative pattern on the basis of the observed image
- suppresses noise and perform background normalisation by suppressing uninteresting systematic or patterned variations
- applied uniformly and context-independent

Labelling

- based on a model that assumes that the informative pattern has structure as a spatial arrangement of events
- determines in what kind of spatial events each pixel participates
- labelling operation:
  - *edge detection, corner detection,*
  - *thresholding* (e.g. filters significant edges and labels them),
  - *identification of pixels* that participate in various shape primitives

3. Images and Graphics

3.5 Image Analysis / Image Transformation

Digital Image Data Structures are achieved through image transformations:

Let $I, J, G \subseteq N$ and $f: I \times J \to G$ an image.

$\tau : G^{I\times J} \to G^{I\times J}$ is an image transformation.

Examples:

1. Location dependent weakening of Intensity (e.g. effect with lens):

$$\tau(f)(x,y) = g(x,y) = f(x,y) + intensityfunction(x,y)$$

2. Threshold-Transformation (e.g. filtering of greyscales/colors):

Let $G := \{0,\ldots,r\}, r \in N$ and $s \in G$ the threshold,

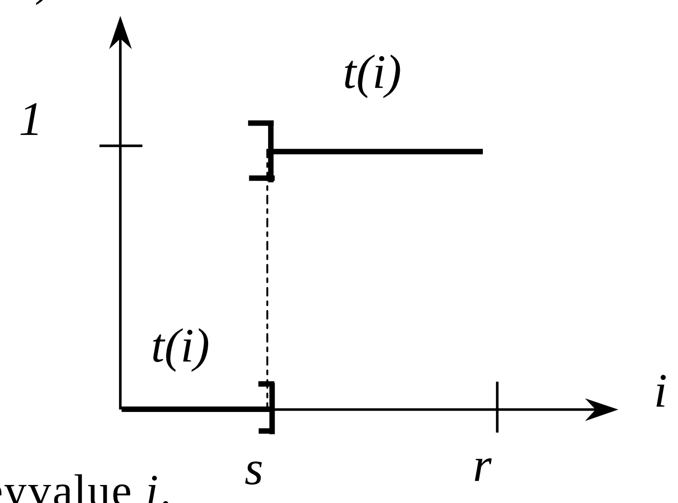
$f: I \times J \to G$ an image and

$t: G \to G'$ (here $G' := \{0,1\}$) is defined as:

$$t(i) := \begin{cases} 0 & \text{for } 0 \le i \le s \\ 1 & \text{for } s < i \le r \end{cases} \quad \text{so that}$$

$h_{t(f)}$ is the new Histogram of h_f defined as:

$h_f : G \to N$ with $h_f(i) = \left| f^{-1}(i) \right| =$ number of pixels with greyvalue i.



Example:

3. Histogram Spreading (e.g. to increase the contrast of an image)

Let $G := \{0,\dots,r\}, r \in \mathbb{N}$ and $f : I \times J \rightarrow G$ an image.

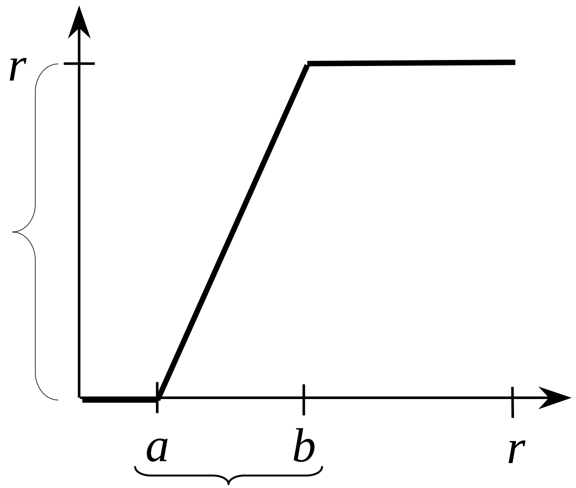
Assume the image f uses only grey values in the scale of $0 < a \leq i \leq b < r$.

With t defined as follows, the whole greyscale values of G are used.

$$t(i) := \begin{cases} 0 & \text{for } 0 \leq i \leq a \\ \dfrac{r}{b-a}(i-a) & \text{for } a < i < b \\ r & \text{for } b \leq i \leq r \end{cases} \quad \text{so that } h_{t(f)} \text{ is the new Histogram, } t(i) \in [0, r]$$

Analogous to spreading the greyscale can be compressed and used only in partial intervals.
A combination of both is e.g. useful to manipulate film materials.

Range of new greyscale values



Range of old values

3. Images and Graphics

3.5 Image Analysis / Grouping and Extracting

Grouping:

- identifies events by collecting or identifying maximal connected sets of pixels participating in the same kind of event (neural networks!)
- determines new sets of entities
- changes the logical data structure
- entities of interest after grouping are sets of pixels
- e.g. *line-fitting* is a grouping operation, where edges are grouped into lines

Extracting:

- computes for each group of pixels a list of properties:
- centroid, area, orientation, spatial moments, grey tone moments, spatial-grey tone moments, circumscribing circle, inscribing circle, number of holes in a region, average curvature in an arc, etc.
- measures topological or spatial relationships between two or more groupings, i.e. clarifies whether two groupings touch, are spatially close or layered

3. Images and Graphics

3.5 Image Analysis / Matching

Matching:

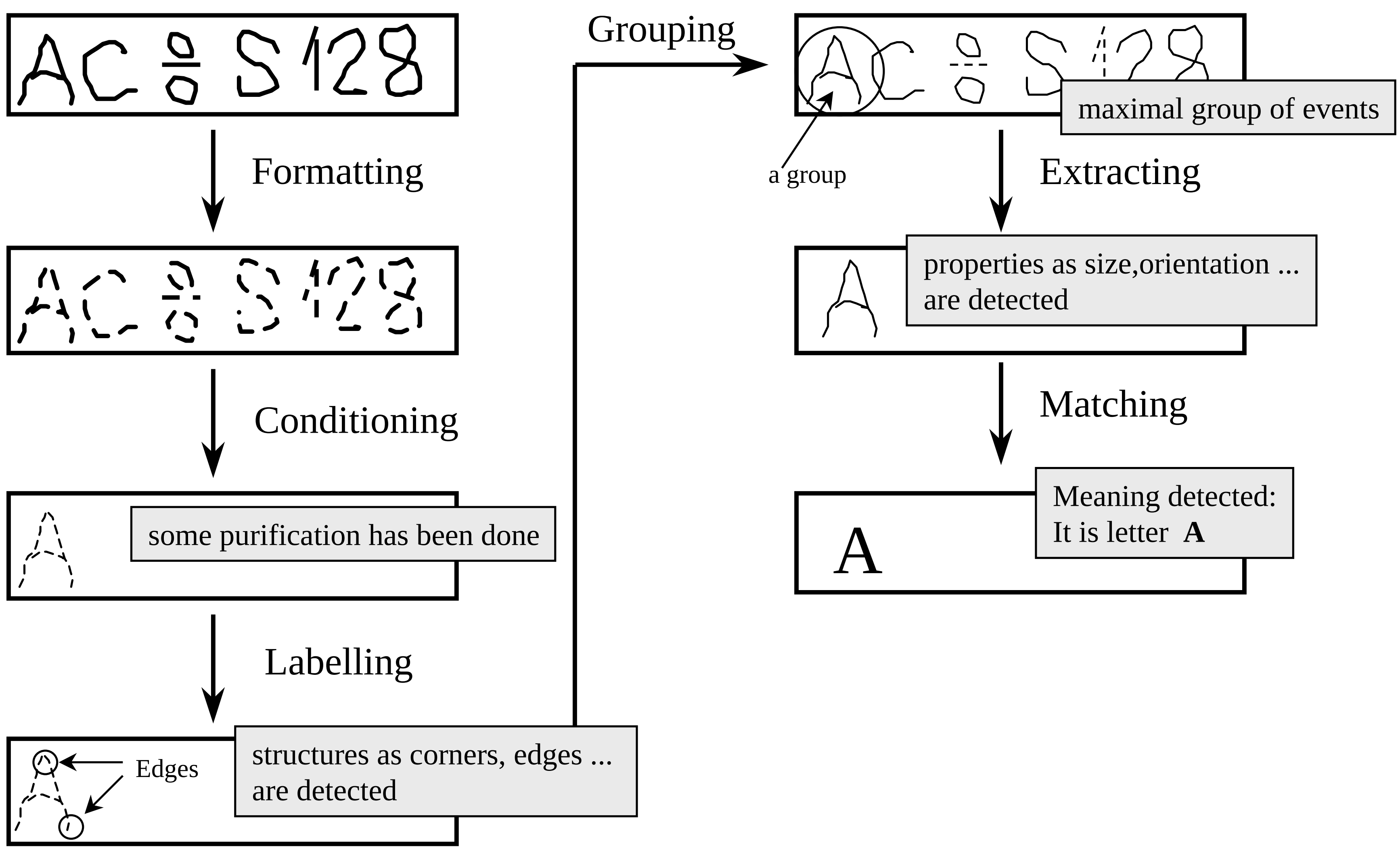
- determines the interpretation of some related set of image events recognised previously with the extracting step
- associates events with some given 3D objects or 2D shapes
- *template matching* is a classical example of a wide variety of matching operations, compares examined pattern with known and stored models and chooses the best match

Conclusion:

- Conditioning, labelling, grouping, extracting and matching constitute a canonical decomposition of the image recognition problem.
- Each step prepares and transforms the data to facilitate the next step.
- On any level the transformation is an unit process and data are prepared for the unit transformation to the next higher level.
- Depending on the application, the sequence of steps has more than one level of recognition and description process.

3. Images and Graphics

3.5 Image Analysis / Detection of license plates



3. Images and Graphics

3.6 Image Transmission

Examination of the network for image transmission:

- network must accommodate bursty data transport due to large size of images and different compression results
- reliable transport over the network is required
- time dependence is not a dominant characteristic of the image (in contrast to audio and video)

Image representation format used for transmission:

- *raw image:* Image generated through the video digitizer and transformed in its digital format (*size = spatial_resolution x pixel_quantization), e.g.* 640 x 480 pixels and quantization of 8 bits requires transmission of 307.200 bytes
- *compressed image:* Downsized image with methods such as JPEG or MPEG.
- *symbolic image:* Image represented through symbolic data as image primitives, attributes and other control information (e.g. computer graphics, image = [red background, yellow circle of 5 cm radius with center point(x,y), ...).