Computer Graphics Charles University

Monochrome Image Reproduction

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

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

Preception of Grey

- Grey has a single **attribute**
  - **intensity** (physical quantity)
  - **brightness** (subjective human perception)
- The relationship between intensity and brightness is **non-linear**
  - Humans perceive brightness in a **relative** fashion (healthy eyes perceive 1% difference)
  - For equally spaced grey values, it is therefore necessary to use a **logarithmic intensity scale** … $\mathbf{I_j = I_0 * r^j}$

of Shades of Grey

- The number **n** of required display colours depends on the dynamic range of the output device (assuming **r = 1.01**):

Device	**dynamic ($1/I_0$)**	**n**
Display	100-3000	460-800
Photography	100	460
Transparency	1000	700
B&W print	100	460
Colour print	50	400

Halftoning and Dithering

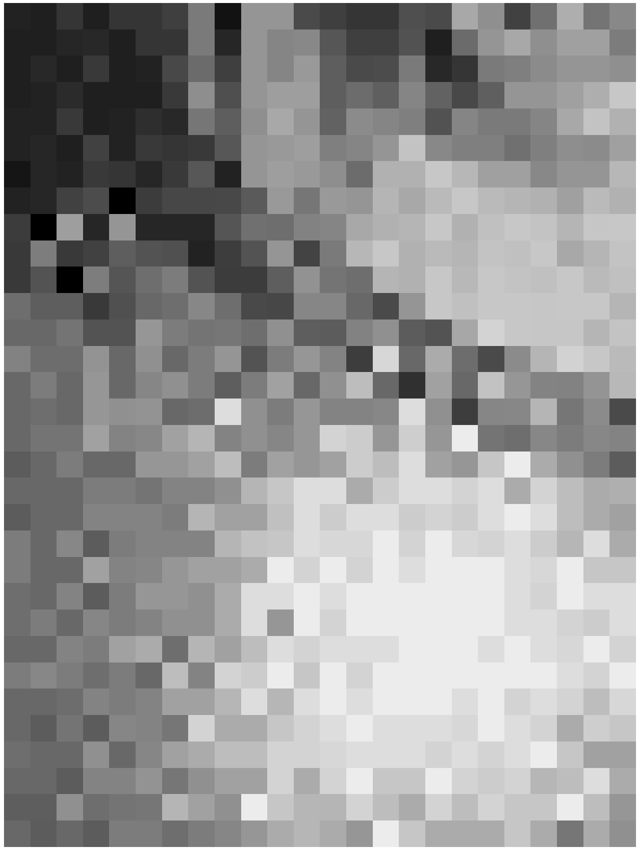
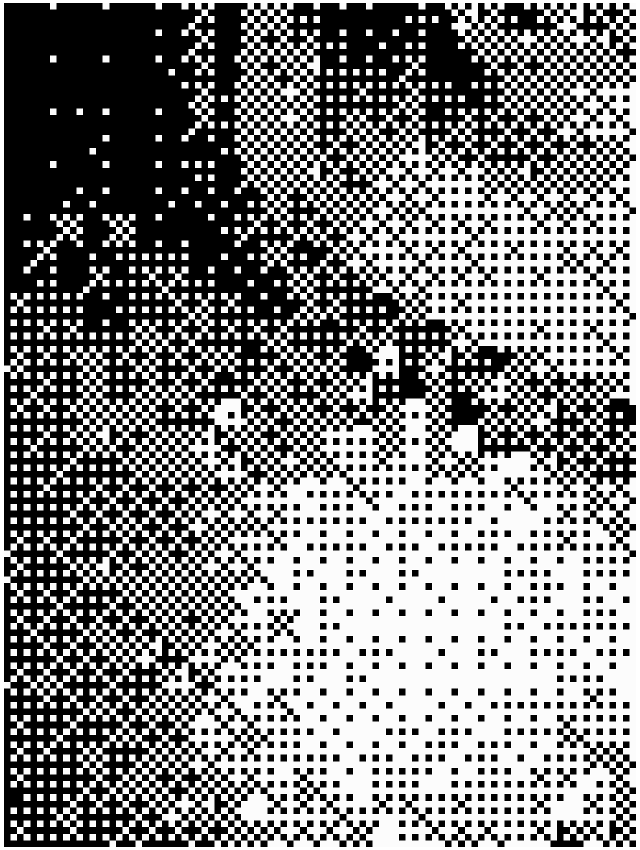


Image reproduced with only a few shades of grey



B&W output device

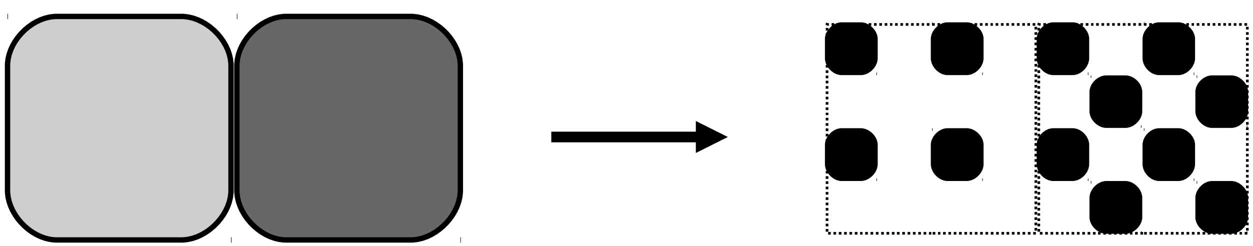
Halftoning and Dithering

- Mimic the appearance of grey on devices with **small coulour spaces**
  - Increases colour resolution at the expense of spatial resolution
  - Typical use: **B & W printers** or displays
- **Halftoning**: the output can enlarge the image resolution (1 : N)
- **Dithering**: no image enlargement (1 : 1)

Halftoning

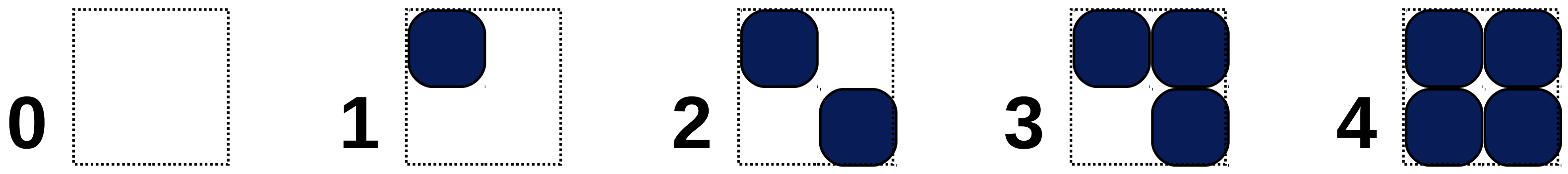
- Situation: the output device is only capable of displaying **black dots** (**1**) on a **white background** (**0**)
- For each input pixel (with range **[0, N²]**) draw a square of **N×N output pixels**
  - The resulting hue of grey depends on the number of black dots in the N×N square

shades no. 4 and 8 (scale 1:16)

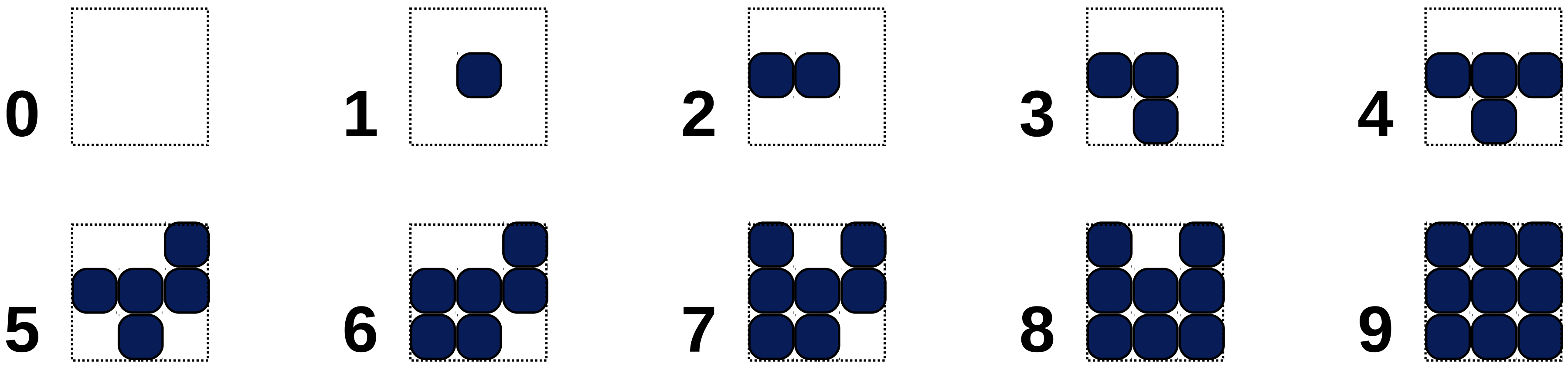


Halftoning Rasters

Regular grid 2×2



Grid 3×3



Incremental Raster

- A halftone raster is **incremental**, if:
  - The pattern for shade **k** contains exactly **k** black pixels
  - Two neighbouring patterns (**k** and **k+1**) only differ in one pixel (**k+1** only has one additional black pixel)
- An incremental raster can be stored in a **matrix** of size **N×N** that contains the integers **[0, N²-1]**
  - e.g. $$\mathbf{M} = \begin{matrix} \mathbf{6} & \mathbf{8} & \mathbf{4} \\ \mathbf{1} & \mathbf{0} & \mathbf{3} \\ \mathbf{5} & \mathbf{2} & \mathbf{7} \end{matrix}$$

Regular Raster I

I) Size 2×2:

$$\mathbf{M}^{(2)} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}$$

II) Step N×N → 2N×2N:

$$\mathbf{M}^{(2N)} = \begin{bmatrix} 4\mathbf{M}^{(N)} & 4\mathbf{M}^{(N)} + 2\mathbf{J}^{(N)} \\ 4\mathbf{M}^{(N)} + 3\mathbf{J}^{(N)} & 4\mathbf{M}^{(N)} + \mathbf{J}^{(N)} \end{bmatrix}$$

The matrix $\mathbf{J}^{(N)}$ is of the type **N×N** and contains the same units

Regular Raster II

$$M^{(4)} = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

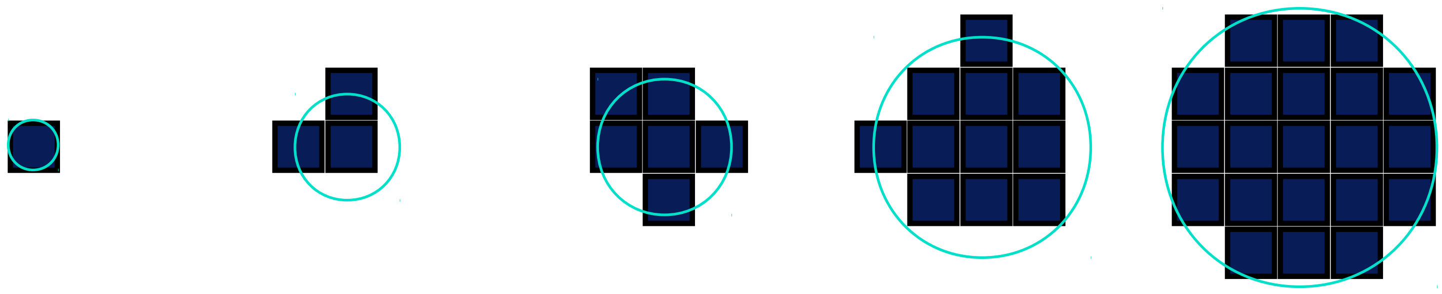
- Regular sampling points are always **evenly distributed**
- Regular raster is suitable for screens and some printers (dot matrix with low resolution)

Regular Raster for Printers

Shade 8 on screen and on a high res printer

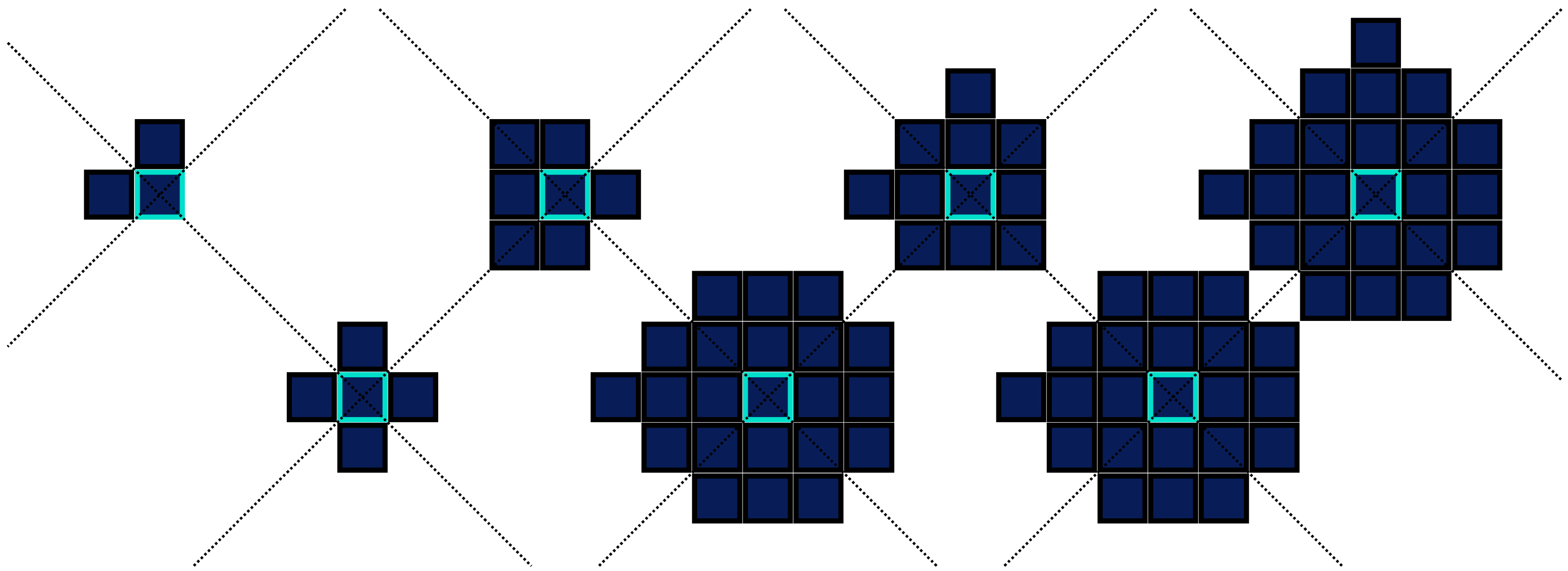
- For darker shades, the dots start to **run into each other** („dot gain“)
- Darker shades depend on individual dots that are not represented well on some printing technologies

Dot Grid („screen”)



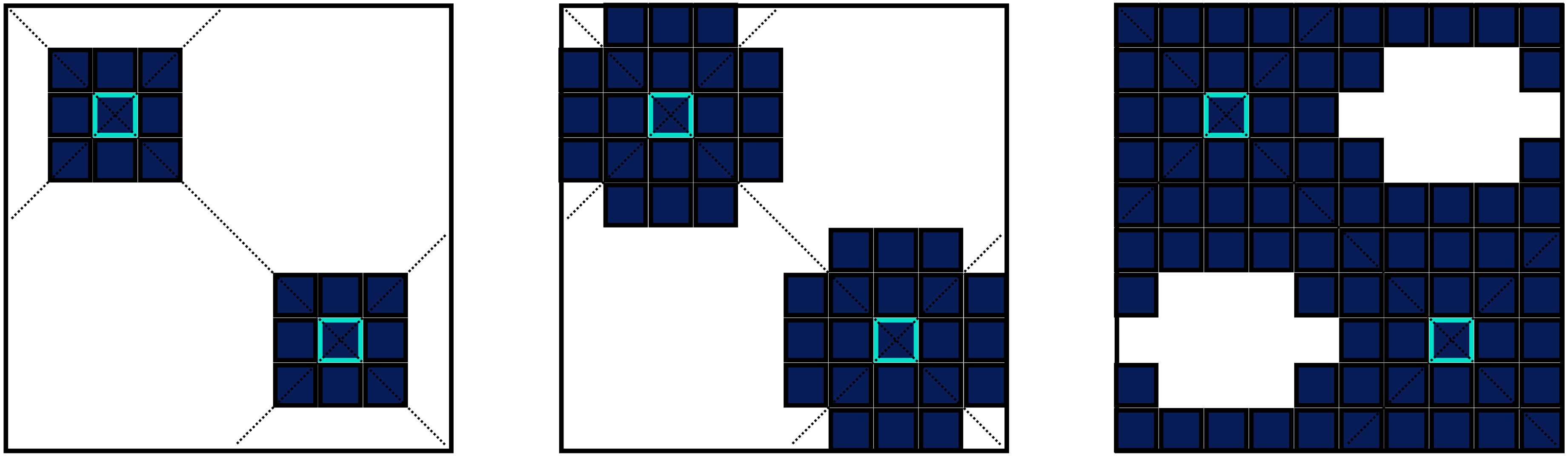
- Each pattern is formed by a **dot** of increasing size
  – No individual dots are printed (up to shade #1)
  – Dot gain will not be as damaging
  – Resolution is lost, though

Dot Raster – Rotation



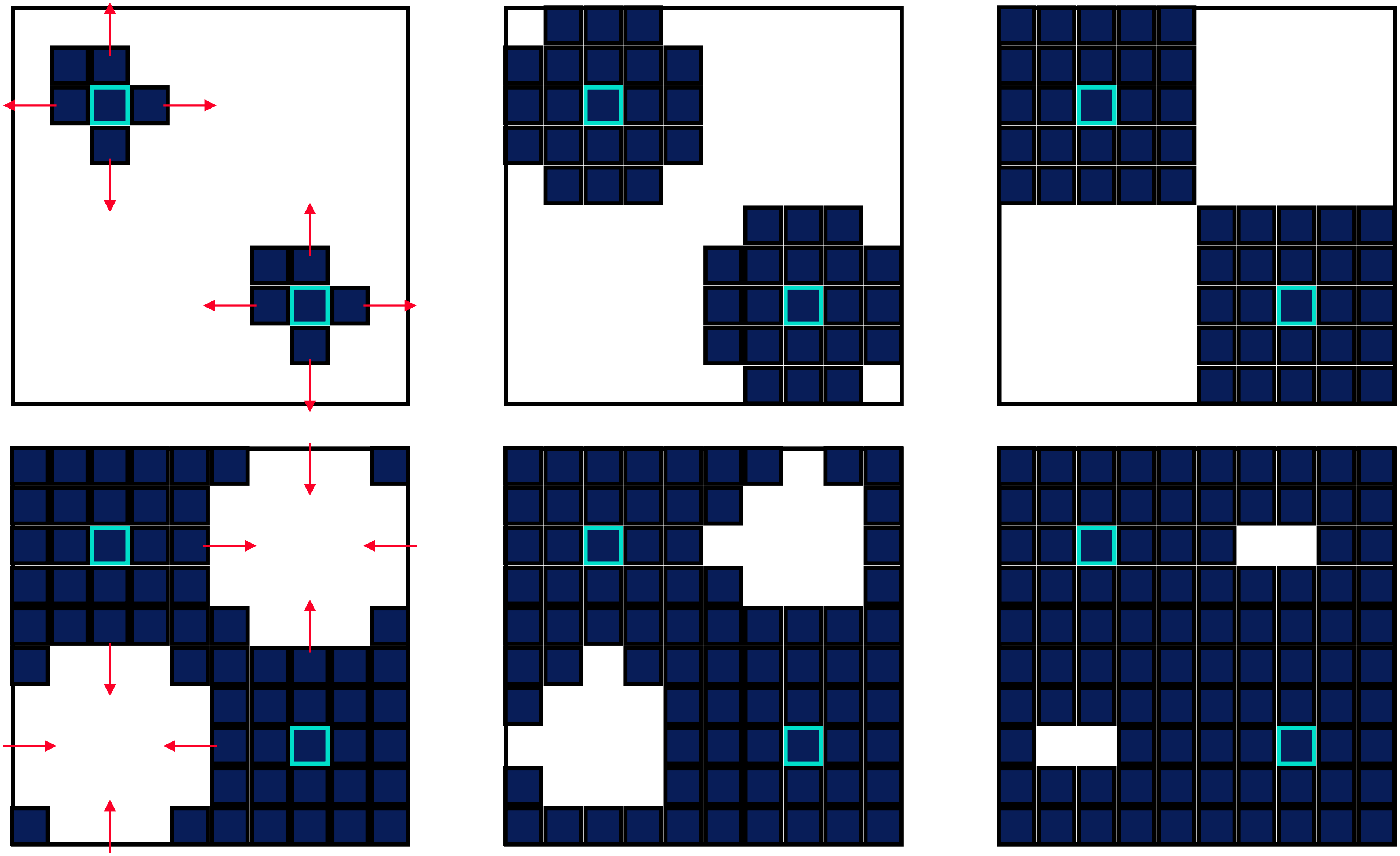
- Dot rasters are often **rotated** (by **<u>45°</u>**, 15°, 75°,..)
  - – Eliminates vertical and horizontal lines (visible to the eye)
  - – For rational directions this can be stored in a matrix

Dot Raster Variants



- **Square dots** (difficulties with subtle shading gradients – „vignettes”)
- **Circular dots** (plus many modifications)

Construction of a Dot Raster

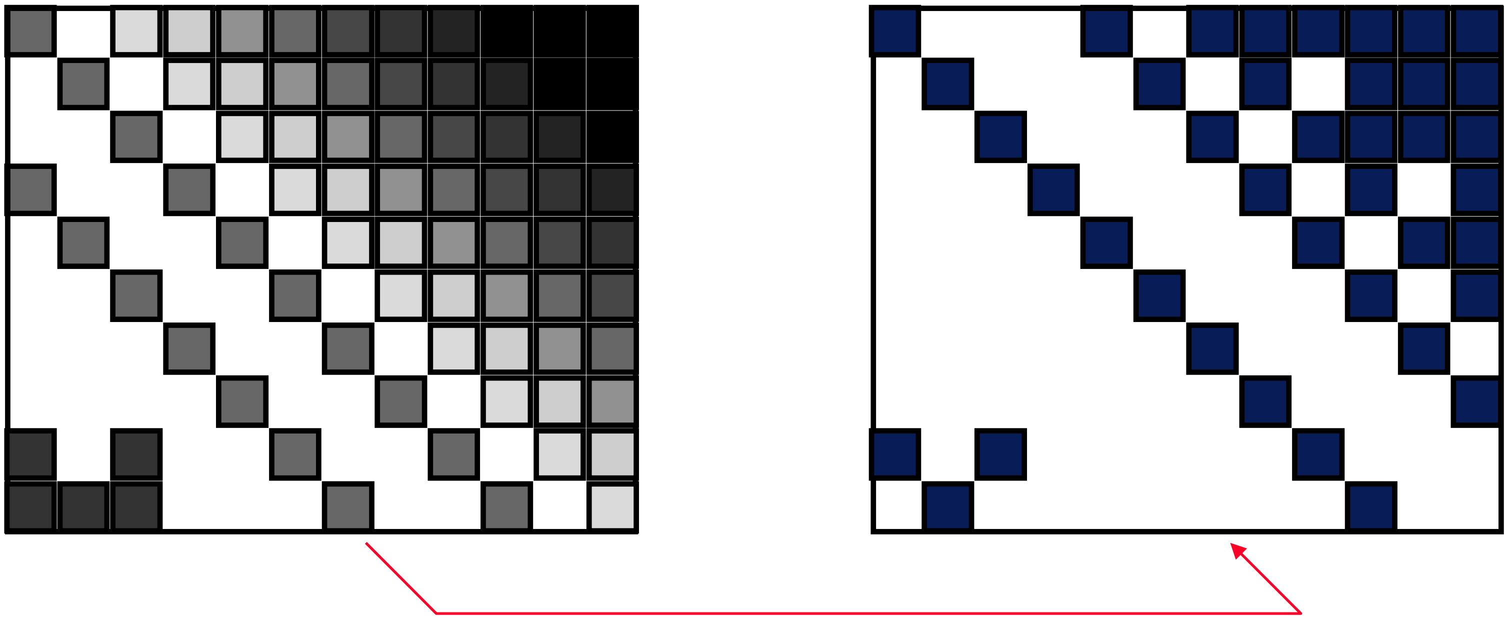


Matrix Dithering

- Imaging at a **scale of 1:1** (one input pixel for each output pixel)
- Any **halftone matrix** can be used
  - Frequently, regular rasters are used
- Several neighbouring pixels share one matrix:

```
procedure MatrixDither ( x, y, color : integer );
begin
  if M[ y mod N, x mod N ] < color
 then PutPixel(x,y,1)
 else PutPixel(x,y,0);
end;
```

Matrix Dithering



- **Small details** (lines) are very distorted
- When using a **non-incremental raster** boundaries between adjacent shades can be highlighted

Random Dithering

- Noise and disorder are less annoying to the human eye than regular dithering patterns
- Very simple implementation:

```
procedure RandomDither ( x, y, color : integer );
begin
  if Random(MaxGray) < color
 then PutPixel(x,y,1)
 else PutPixel(x,y,0);
end;
```

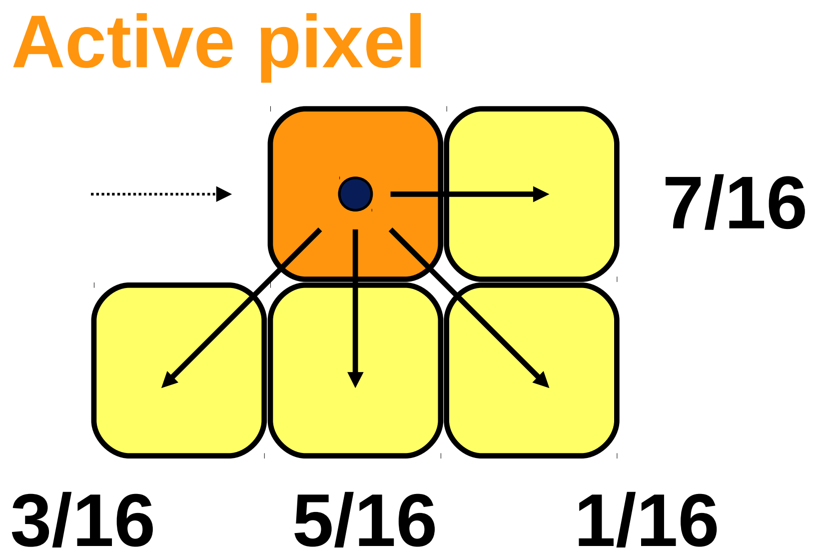
- For **B&W images** the result is too noisy
  – Better results for larger numbers of output colours

Error Distribution Techniques

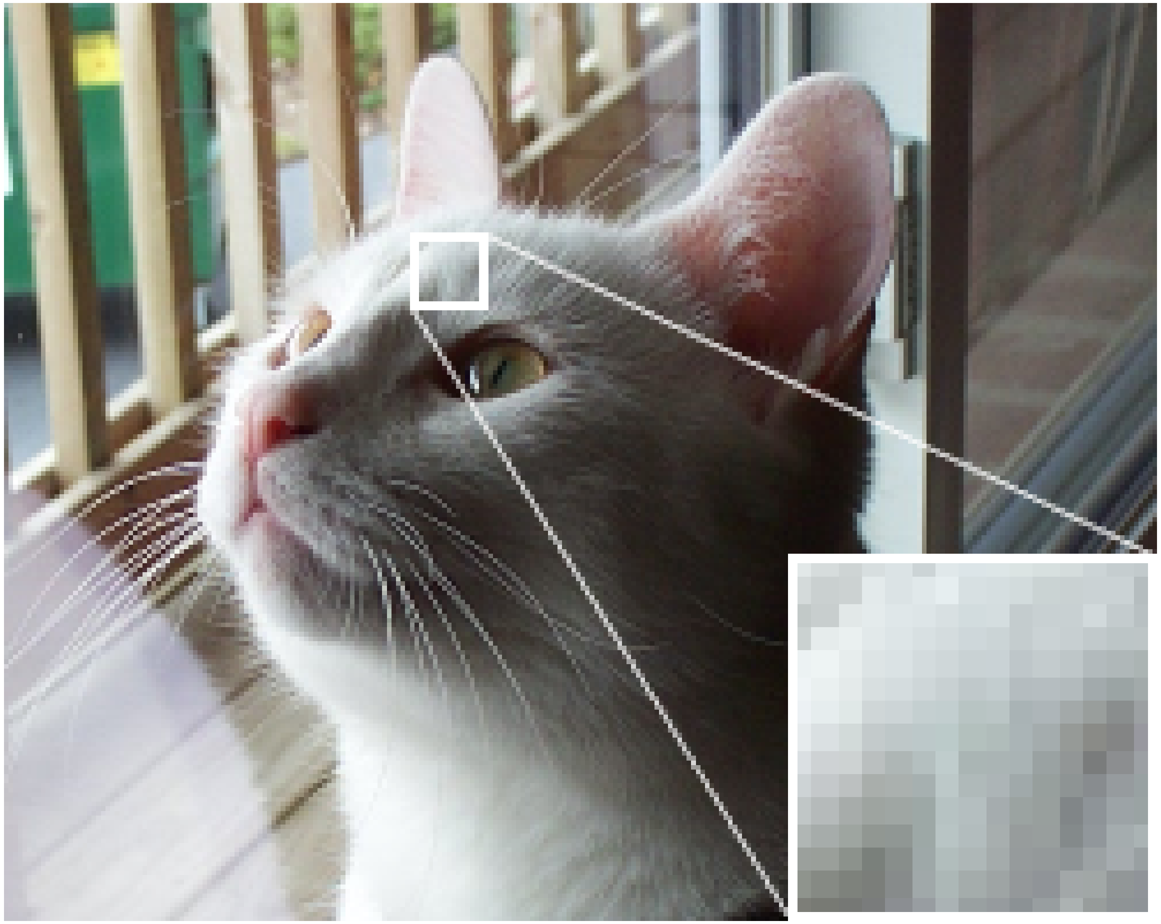
- The intensity of each pixel is rounded to the next displayable value, and it is directly drawn:
  - **0/1** for B&W devices
  - **0, 1, .. K** for multi-tone devices
- The difference (error) between the printed pixel value, and its actual value, is passed to neighbouring pixels
  - This maintains the local ratio between black and white pixels
  - The error is only spread to pixels that have not been drawn yet

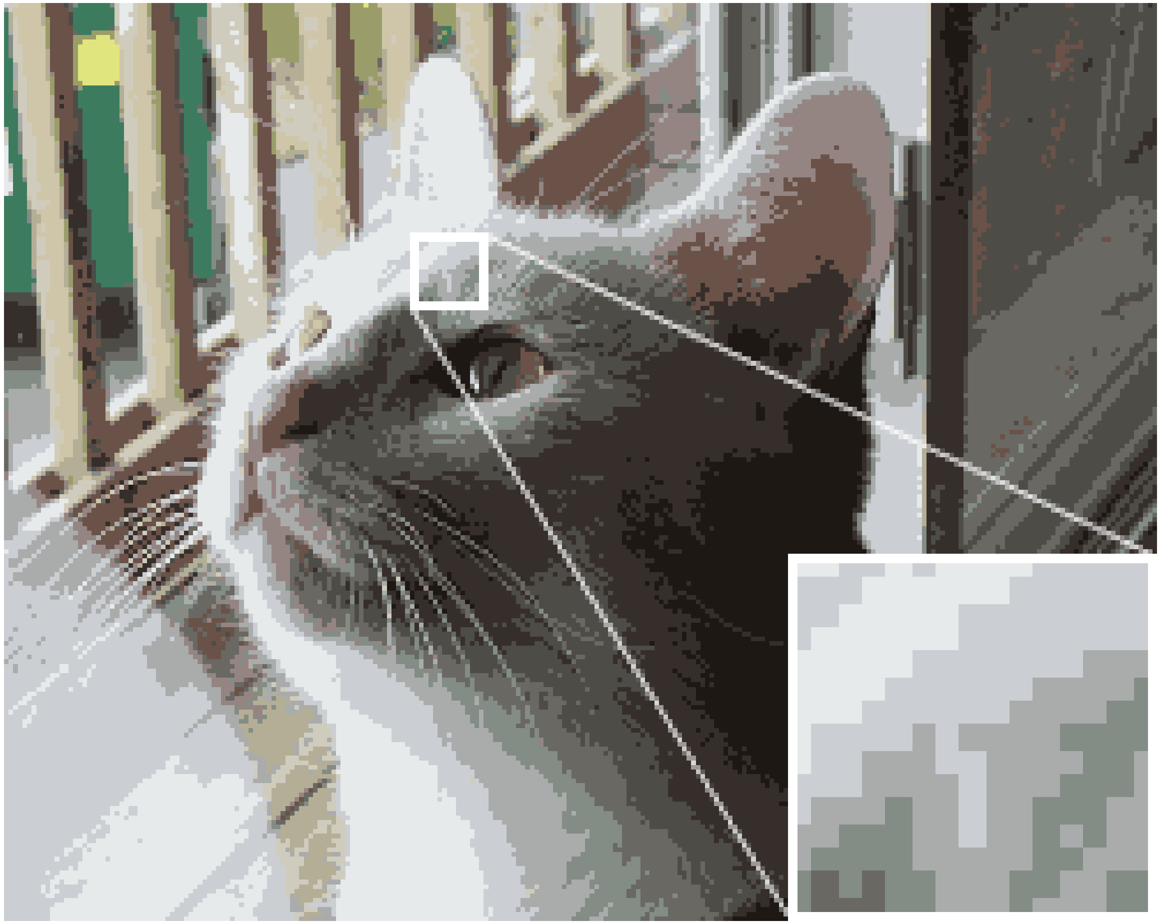
Error Distribution

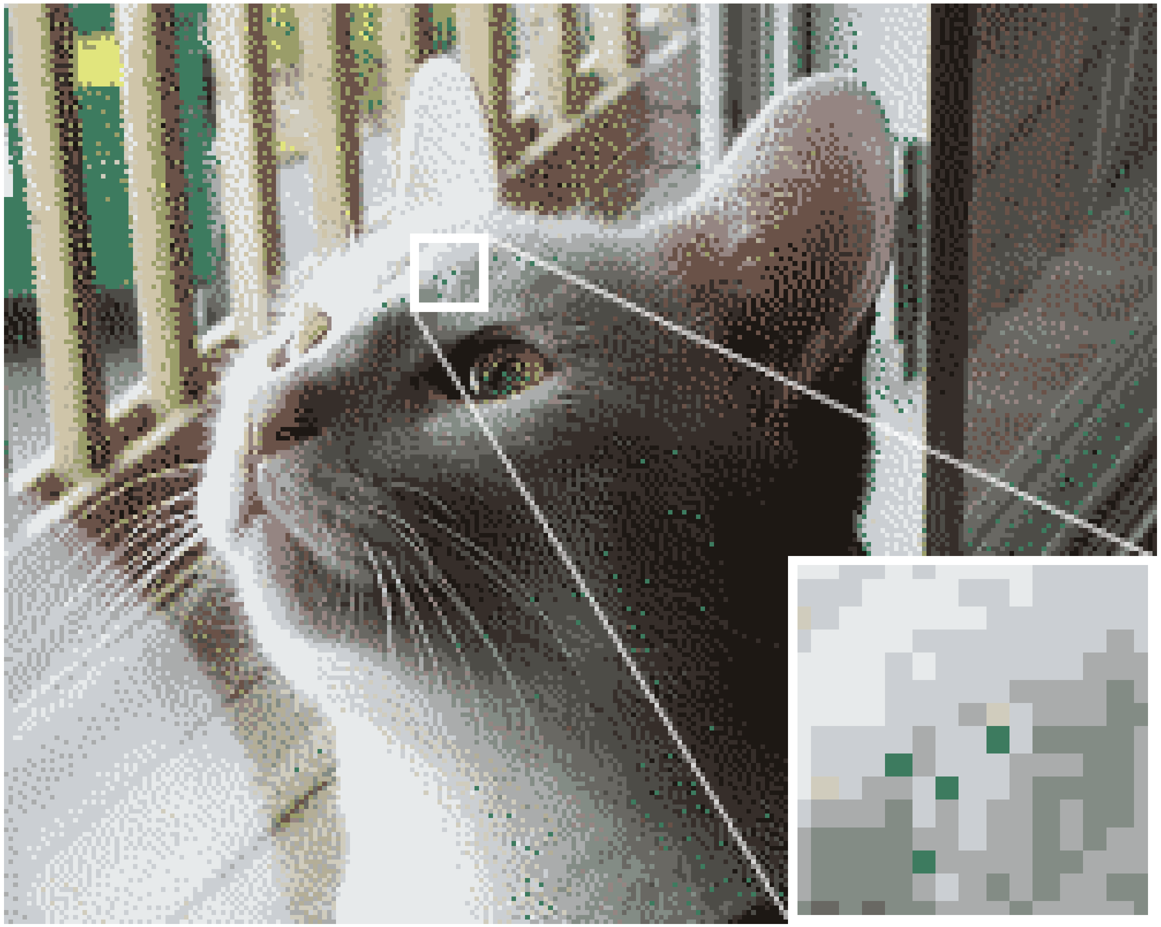
Floyd-Steinberg:



- Drawing proceeds along **scanlines**
  - One can alternate between left to right and right to left
- To **accumulate the error** for the upcoming scanline, one has to use a **buffer**







Other Distribution Filters

F. Sierra:

	■	1/2
1/4	1/4	0

J. Jarvis, C. Judice, W. Ninke:

		■	7	5	
3	5	7	5	3	/ 48
1	3	5	3	1	

Stucki:

		■	8	4	
2	4	8	4	2	/ 42
1	2	4	2	1	

Error Distribution Techniques

- **High output quality** on monitors
  - Appearance is pleasing to the human eye
- **Disadvantages:**
  - One has to work on a **scanline basis**
  - One cannot return to **previous parts of a scanline** (one cannot use shape filling routines)
  - It is necessary to use a **buffer** of at least 1 scanline size
  - More **time-cosuming**

Random Error Diffusion

- **Visible artefacts**
  - Small artefacts, strings of beads, ...
- **Reduction of regularity**
  - **Alternating directions** for neighbouring lines
  - **Randomisation** … adding „**blue noise**“
    - » Randomisation via distribution filter
    - » Randomisation via rounding limit
    - » Various alternative filters (PDF)

Multiple Output Colours

- We assume **K+1 output tones** are possible
  - **0** ÷ **K** (**0** .. white, **K** .. black)
- Our dithering method can map **M+1 input tones** to two output colours:
  - input: **0** ÷ **M**
  - output: **0** / **1**
- As output of the combined method, **K*M+1 tones** are possible

Multiple Output Colours

```
function Dither ( x, y, color : integer ) : integer;
  { output colour: 0 to M, returns 0 or 1 }
...


procedure MultiDither ( x, y, color : integer );
  { output colour: 0 až K*M, used shade: 0 tp K }
var base : integer;
begin
  base := color div M;          { 0 <= base <= K }
  PutPixel( x, y, base + Dither( x, y, color mod M ) );
end;
```

Literature

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 563-573
- **R. Ulichney**: ***Digital Halftoning***, MIT Press, 1987
- **D. Lau, G. Arce**: ***Modern Digital Halftoning***, M. Dekker, 2001

End

Further information:

- **J. Jarvis, C. Judice, W. Ninke**: ***A Survey of Techniques for the Image Display of Continuous Tone Pictures on Bilevel Displays***, CGIP vol.5, #1, March 1976