Principles of 2D Printing

Sai-Kit Yeung

ISTD, SUTD

Notes courtesy by Wojciech Matusik, slides from Victor Ostromoukhov & Daniel Cohen-Or

Plan for Today

- 2D Printing
- Halftoning
- Digital Halftoning

History of Printing

- Woodblock printing (AD 220)
- Movable type printing (AD 1040)
  - Developed in Asia (ceramic, wood, metal)
  - Metal movable type in Europe (Johannes Gutenberg, 1439)
- Type-founding and typesetting

http://en.wikipedia.org/wiki/History_of_printing

Lithography (1796)

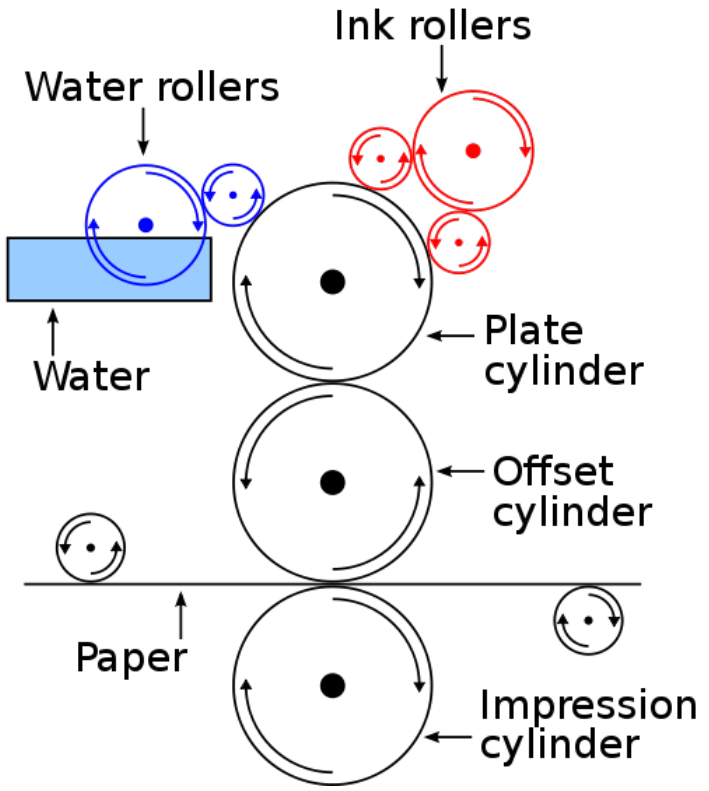
- Plate is a medium to transfer ink to a blank paper sheet
- Plate is divided into
  - **hydrophilic** regions accept water but repel ink
  - **hydrophobic** regions repel water and accept ink

lithographerinbrooklyn.blogspot.com

Offset Printing (1870s - now)

- Inked image is transferred (offset) from a plate to rubber blanket, then to the printing surface
- High-quality, high-speed, low-cost for high-volume runs
- Requires fabricating plates

solopress



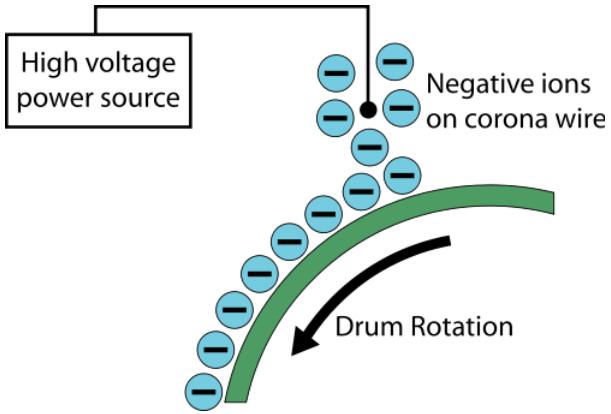
Digital Printing

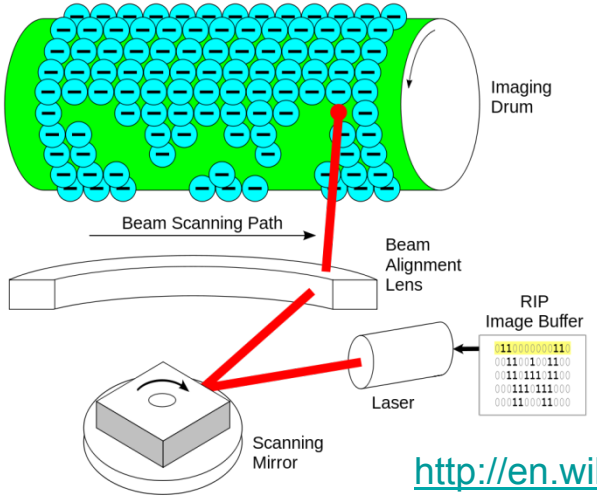
- Printing from a digital image directly to a physical medium
- No printing plates
  - Faster, less expensive for a single copy or short runs
- Most popular methods
  - Inkjet printers
  - Laser printers

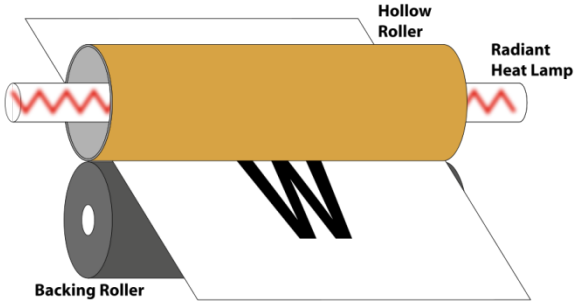
digitalprintingmchine.blogspot.com

Laser Printing

- **Charging** – negative charge is applied to a photoreceptor (drum)
- **Exposing** – laser beam neutralizes the charge
- **Developing** – toner particles are attracted to areas touched by the laser
- **Transferring** – photoreceptor is pressed against the paper
- **Fusing** – heat and pressure bond the plastic powder to the paper
- **Cleaning** – excess toner is removed from the photoreceptor





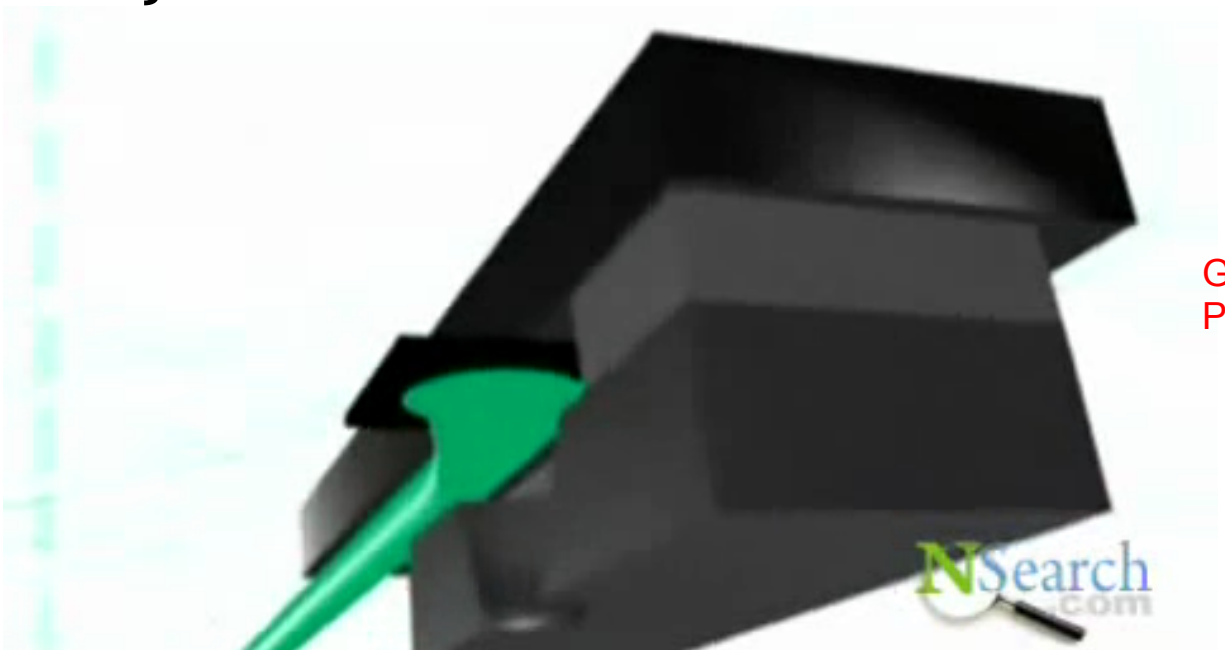


http://en.wikipedia.org/wiki/Laser_printers

Laser Printing

Inkjet Printing - Thermal (Canon, HP, Lexmark)

- Use a heating element in an ink-filled chamber behind each nozzle
- A pulse of current is passed through the heating element
- Vaporized ink in the chamber leads to a pressure change
- A droplet is jetted

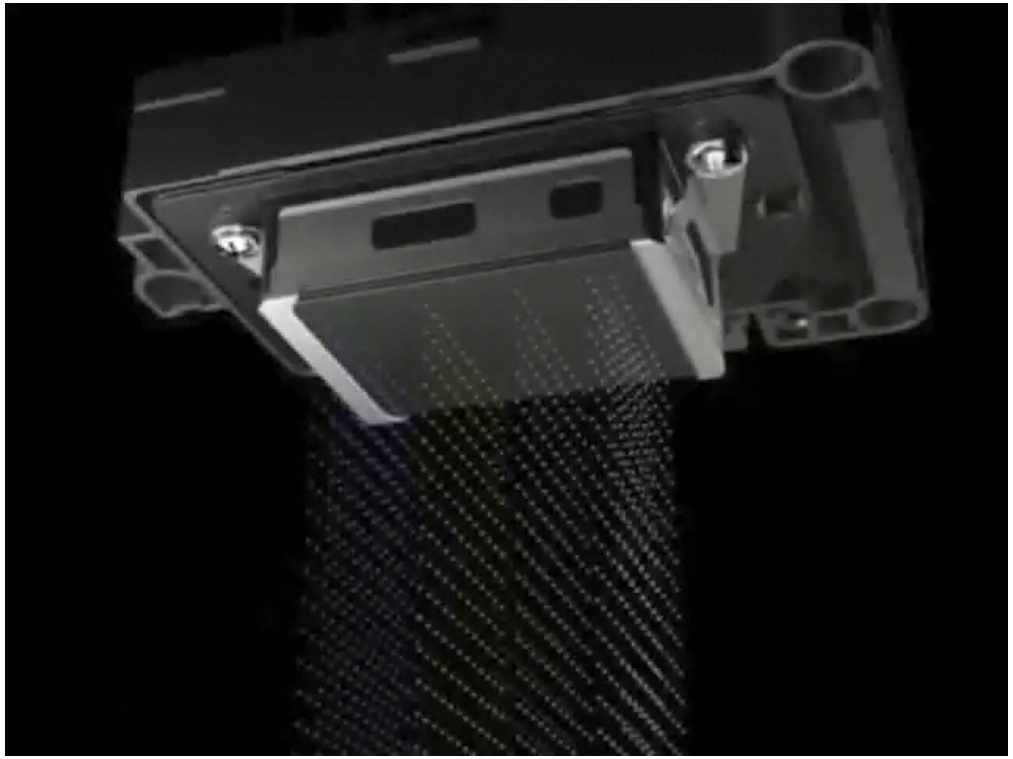


General principle:
Pressure to jet the ink

http://www.youtube.com/watch?v=V_RW4iD-3k0

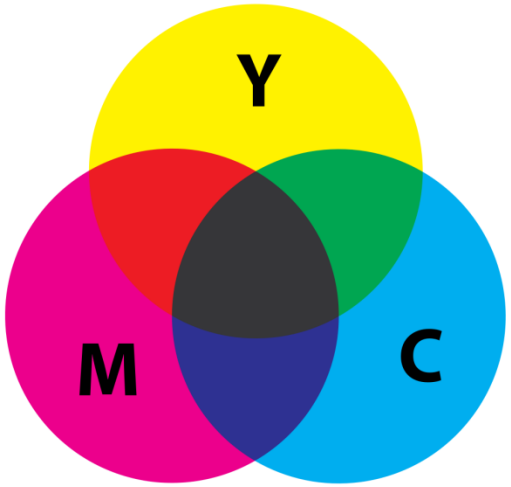
Inkjet Printing – Piezoelectric (Epson, Brother)

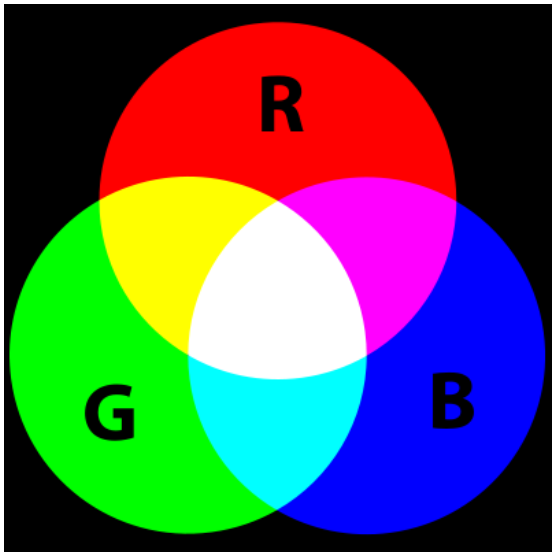
- Use piezoelectric material in an ink-filled chamber behind each nozzle
- Piezoelectric material changes shape when voltage is applied
- Generated pressure pulse ejects a droplet



Subtractive vs. Additive Color

- Subtractive color
  - We start with white light (e.g., white paper)
  - Colored inks act as filters that remove light
- Additive color
  - We start with no light (e.g., darkness)
  - Light sources of different wavelengths are added

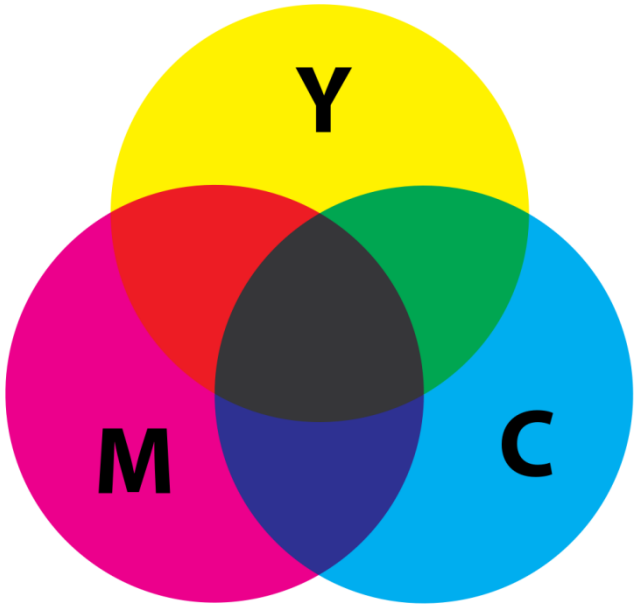




http://en.wikipedia.org/wiki/Additive_color & http://en.wikipedia.org/wiki/Subtractive_color

Color Printing Uses Subtractive Color Model

- CMYK color model
  - Cyan (C) – absorbs red, transparent to green and blue
  - Magenta (M) – absorbs green, transparent to red and blue
  - Yellow (Y) – absorbs blue, transparent to red and green
  - Black ink (K – key) – reduces unwanted color tints, improves sharpness, reduces cost



Cyan: 0,255, 255

Yellow: ?, ?, ?

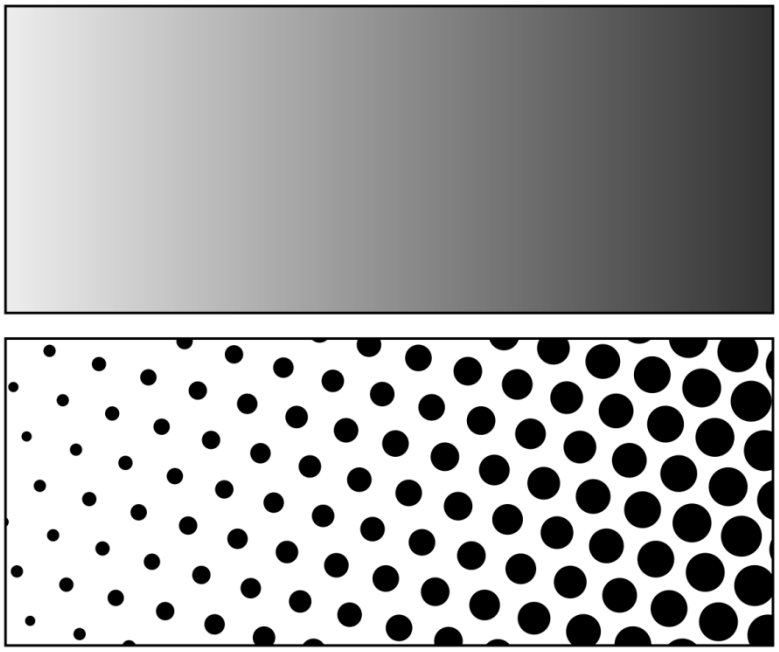
Cyan mix Yellow:
?, ?, ?

Plan for Today

- 2D Printing
- Halftoning
- Digital Halftoning

Halftoning

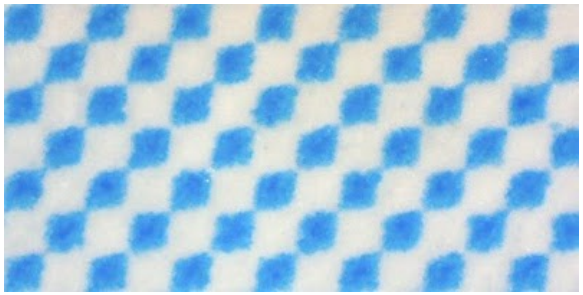
- A method to simulate a continuous tone through the use of dots that vary in size, shape, spacing
- E.g., continuous tone image contains infinite range of greys, the halftone process reduces it to an image with only one color of ink



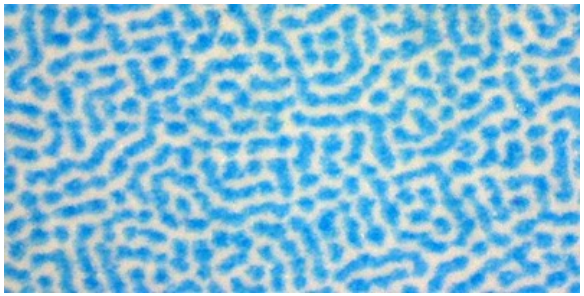
http://en.wikipedia.org/wiki/Halftone

Halftoning

- Traditional screening
  - Amplitude Modulation (AM) – regular grid of dots, dots vary in size
  - Frequency Modulation (FM) – variable dot density
- Dot shapes
  - Round, elliptical, square



AM screen (elliptical)



FM screen

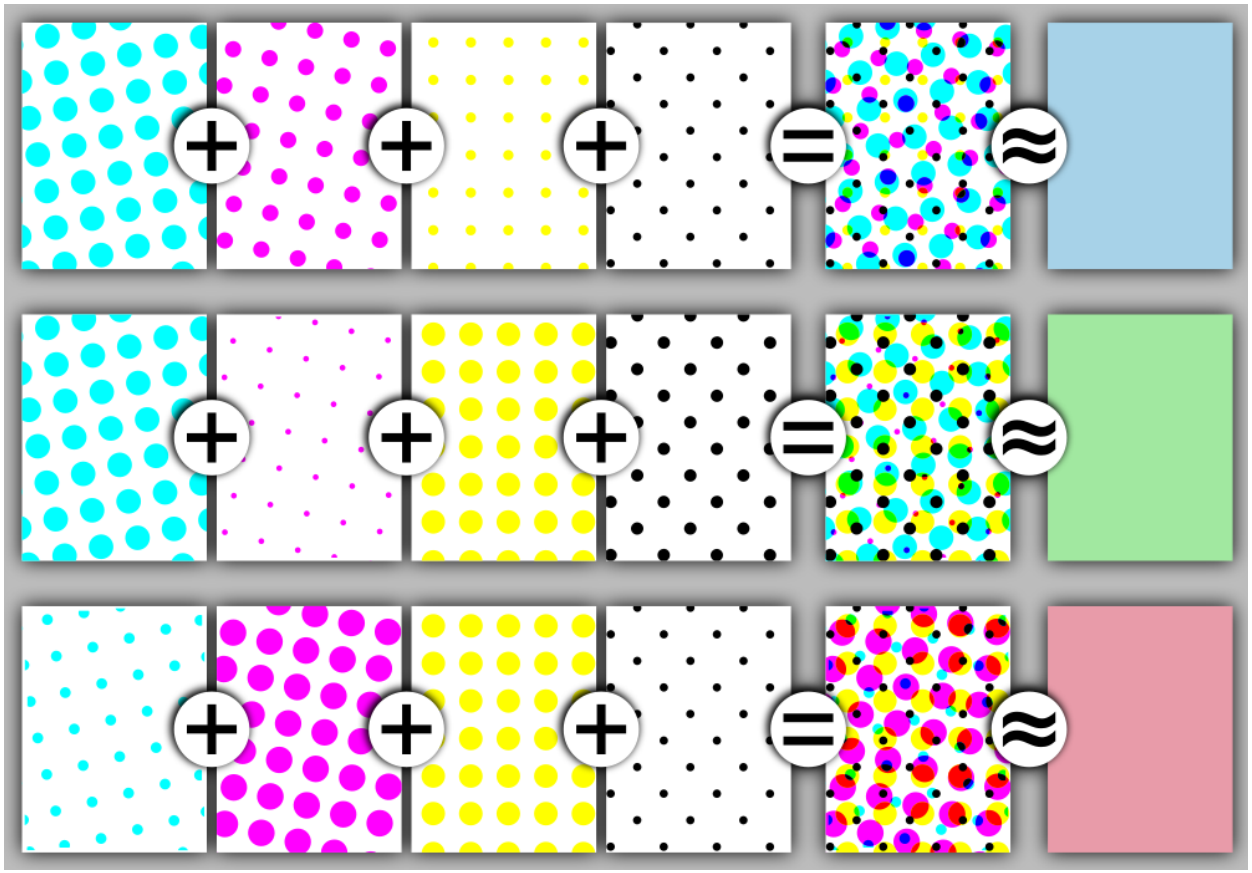
The first printed photo using a halftone, December 2, 1873.

http://the-print-guide.blogspot.com/2010_08_01_archive.html

http://en.wikipedia.org/wiki/Halftone

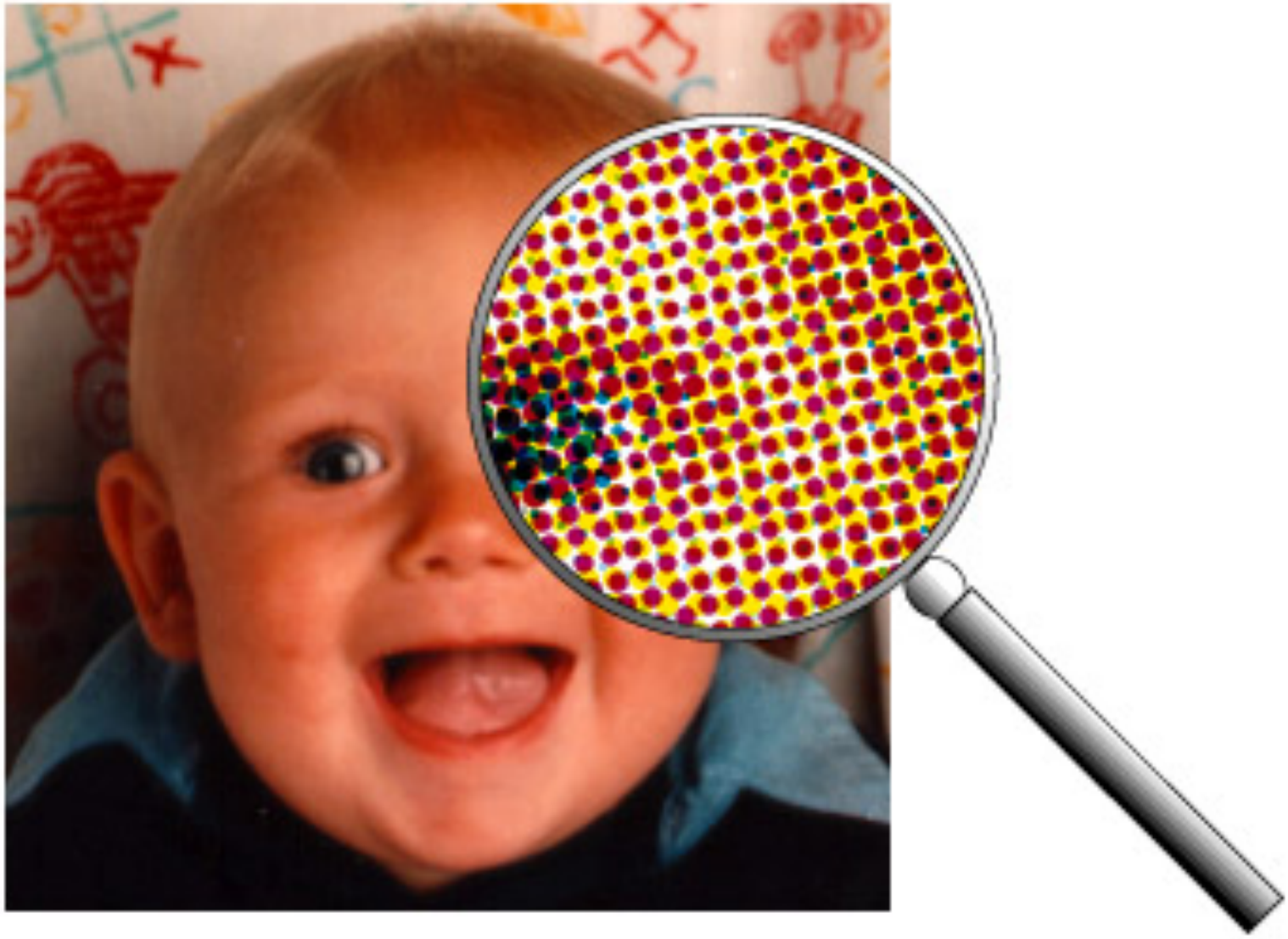
Color Halftoning

- Multiple screens (CMYK)
- Screens are rotated to reduce moiré pattern



http://en.wikipedia.org/wiki/Halftone

Color Halftoning

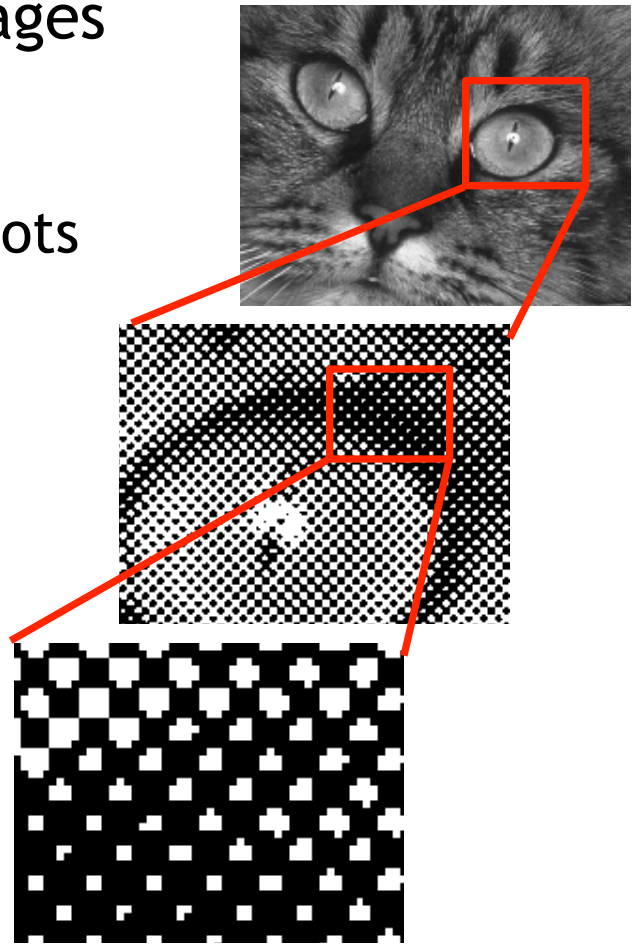


Plan for Today

- 2D Printing
- Halftoning
- Digital Halftoning
  - Point processes
  - Neighborhood processes
  - Global optimization

Digital Halftoning (since 1970s)

- To convert grayscale to bi-level images
- Used by all printers
  - Full color image -> CMY+Black ink dots
  - Grayscale image -> Black ink dots
- Digital world
  - Fixed pixel grid
  - [0, 255] -> { BLACK, WHITE }



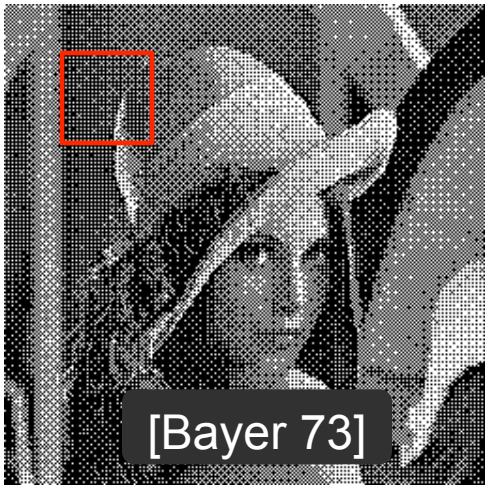
Classification

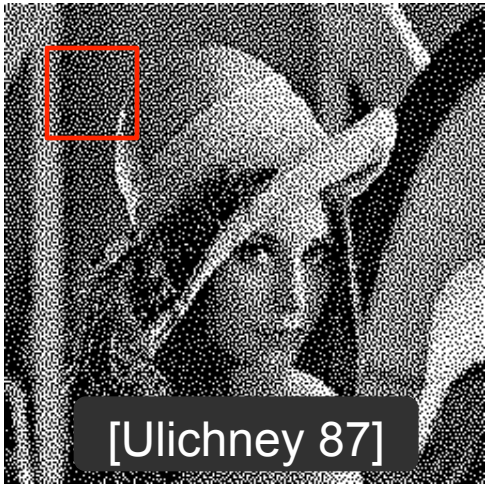
- General classification of halftoning [Model-based digital halftoning. Pappas et al. 03]
  - Class I: Point processes
  - Class II: Neighborhood processes
  - Class III: Global optimization

- Methods from a same class generally share
  - Quality
  - Speed

Class I: Point Processes

- One pixel at a time
  - Compare to threshold mask
- Speed
  - Ignore pixel interactions
  - Class I ⇨ **very fast**
- Quality
  - Regular patterns
  - Noise
  - Class I ⇨ **poor overall quality**





Thresholding

Original image

Simple threshold

$$v(x,y) = trunc(\hat{v}(x,y) + n)$$

Errors are low spatial frequencies.

Thresholding with Random Modulation (Roberts 1961)

- First add noise
- Then quantize

$$\hat{v}(x,y) = trunc(K \times v(x,y) + noise(x,y))$$

$$0 \le noise < 1$$

- Pre-compute pseudo-random function and store in table

http://www.packet.cc/files/pic-code-noise.html

Thresholding

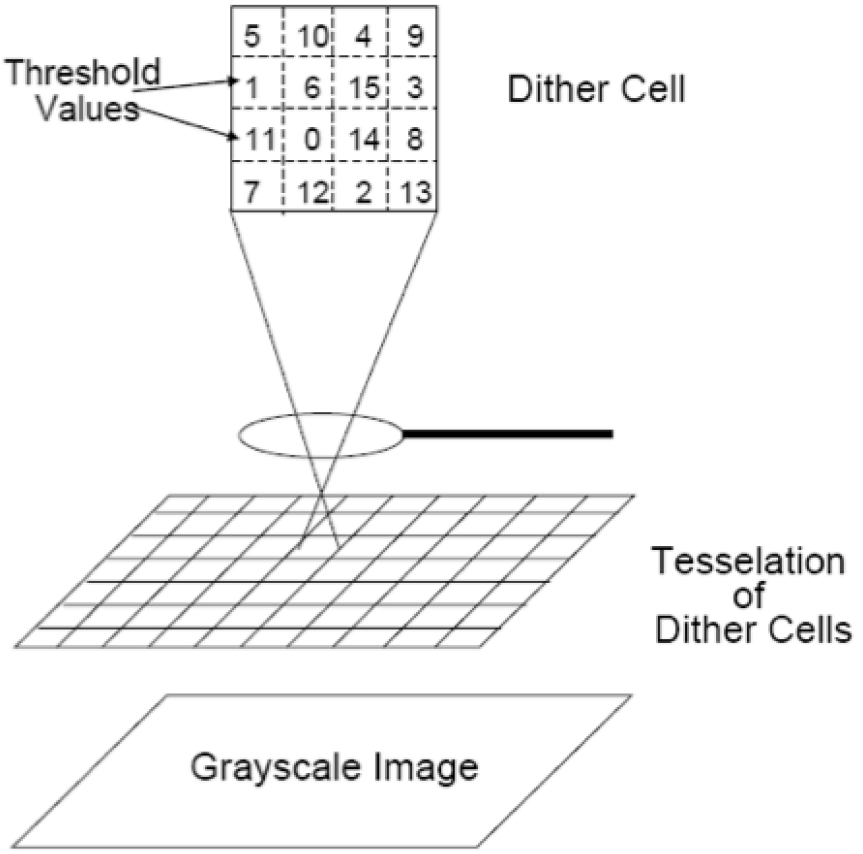
Thresholding + Noise

Dithering

- Each pixel produces a *quantization* error
- The quality of the result may be improved by adjusting the threshold locally, so that adjacent pixels in small areas are quantized with different thresholds.
- This reduces the **average** local quantization error. Matrices of these threshold are called **dither** matrices

Dithering

- Every pixel in a region is thresholded using a different threshold value.

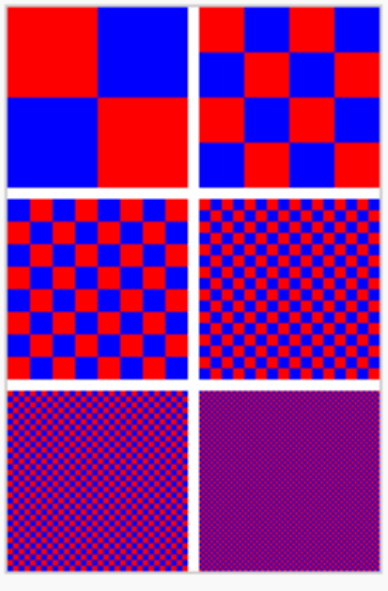


Thresholding + Noise

Dithering

Dithering

- Human visual system integrates information
- The human perception system do not have equal response to all spatial frequencies.
- As the spatial frequencies become higher and higher, our ability to perceive the pattern will be lower and lower.



Dithering

- Break the image into small blocks
- Define a *threshold matrix*
  - Use a different threshold for each pixel of the block
  - Compare each pixel to its own threshold
- The thresholds can be clustered
- The thresholds can be dispersed

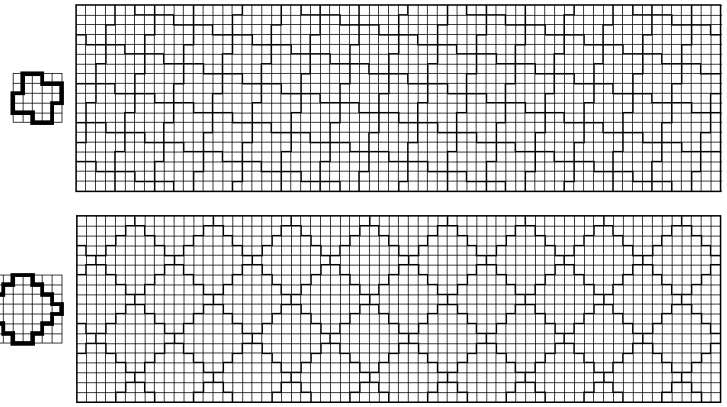
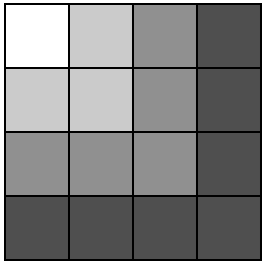


Image block

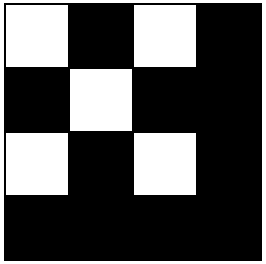


$$\begin{bmatrix} 1 & 0.75 & 0.5 & 0.25 \\ 0.75 & 0.75 & 0.5 & 0.25 \\ 0.5 & 0.5 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

Threshold matrix

$$\frac{1}{16}\begin{bmatrix} 2 & 16 & 3 & 13 \\ 14 & 6 & 11 & 7 \\ 4 & 10 & 1 & 15 \\ 12 & 8 & 9 & 5 \end{bmatrix}$$

Result



$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Dithering Algorithm

```
For all X
   For all Y
      v = getImageValue(x,y)
      i = x mod 3
      j = y mod 3
      if  v >= M[i,j] then
         setPixel(x,y, WHITE)
      else
         setPixel(x,y, BLACK)
```

The dithering matrix M (3x3)

3	**7**	**5**
6	**1**	**2**
9	**4**	**8**

Dithering Example

3	7	5
6	1	2
9	4	8

8	7	5	1	6	2	4	5	3	2
6	8	3	7	2	2	5	4	3	2
9	4	8	3	7	4	5	2	6	4
3	8	9	7	7	2	2	1	3	2
6	9	2	9	7	4	3	2	2	4
9	4	8	8	8	4	4	8	4	4

3	7	5
6	1	2
9	4	8

Dithering matrix

2	1	3
3	2	2
4	8	4

Image

?

Binary image

Dithering Example

3	7	5
6	1	2
9	4	8

8	7	5	1	6	2	4	5	3	2
6	8	3	7	2	2	5	4	3	2
9	4	8	3	7	4	5	2	6	4
3	8	9	7	7	2	2	1	3	2
6	9	2	9	7	4	3	2	2	4
9	4	8	8	8	4	4	8	4	4

3	7	5
6	1	2
9	4	8

Dithering matrix

2	1	3
3	2	2
4	8	4

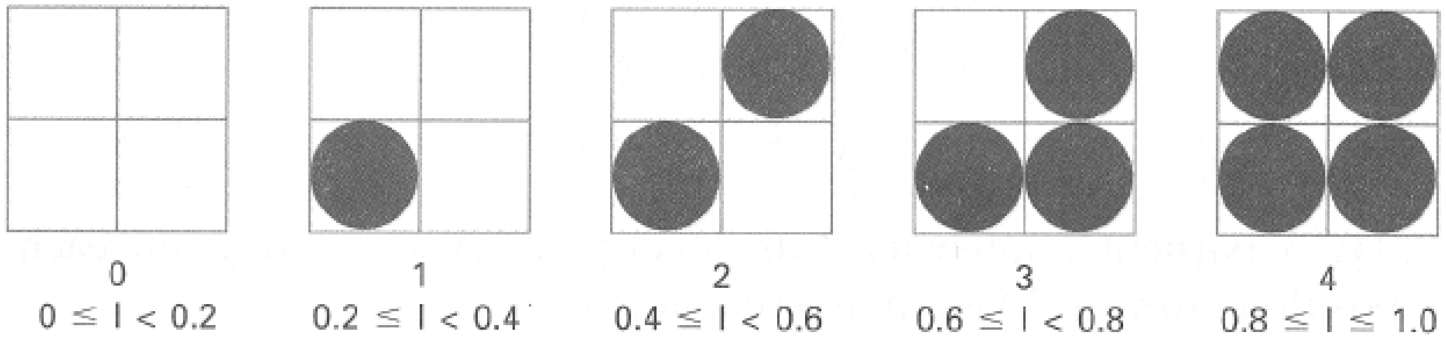
Image

0	0	0
0	1	1
0	1	0

Binary image

Clustered-dot Dithering

- Values in a threshold matrix are set to produce a single cell
  - User cluster of pixels to represent intensity
- The cell size depends on the input value
  - Trade spatial resolution for intensity resolution

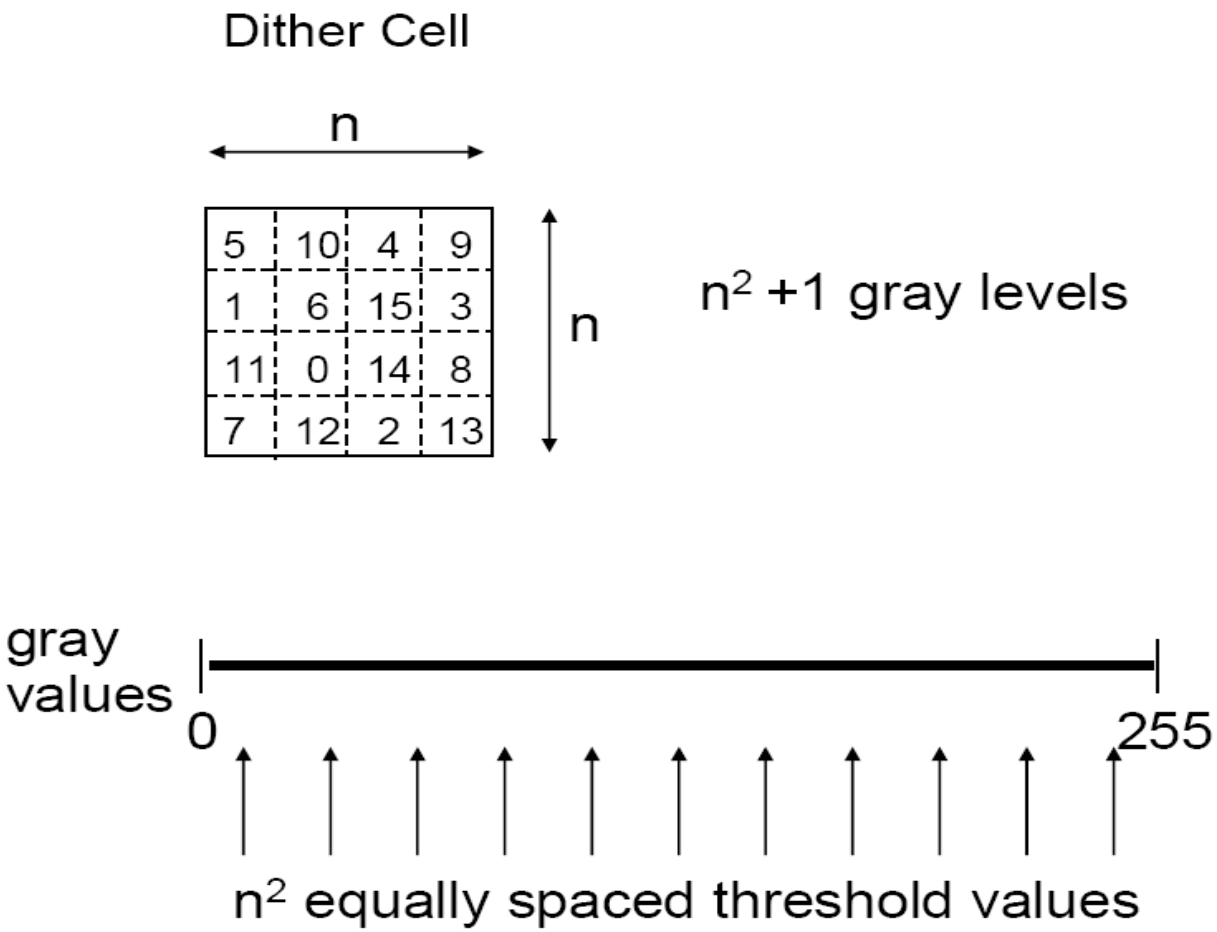


Clustered-dot Dithering

- How many intensities (gray scale values) in a n x n cluster?

Clustered-dot Dithering

- How many intensities (gray scale values) in a n x n cluster?

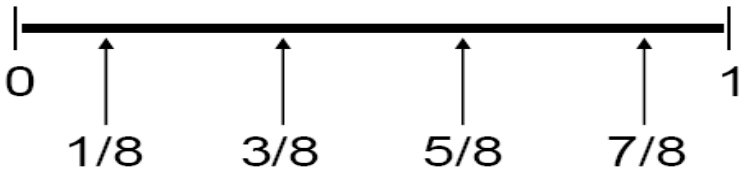


Clustered-dot Dithering

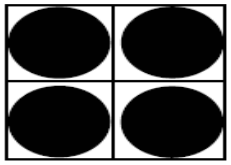
- Example:

dither cell

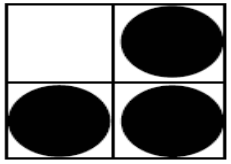
1/8	5/8
7/8	3/8



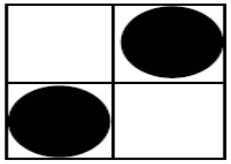
gray = 0...0.125



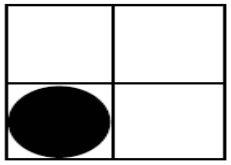
gray = 0.125...0.375



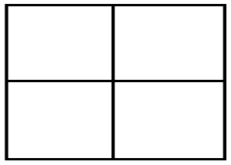
gray = 0.375...0.625



gray = 0.625...0.875

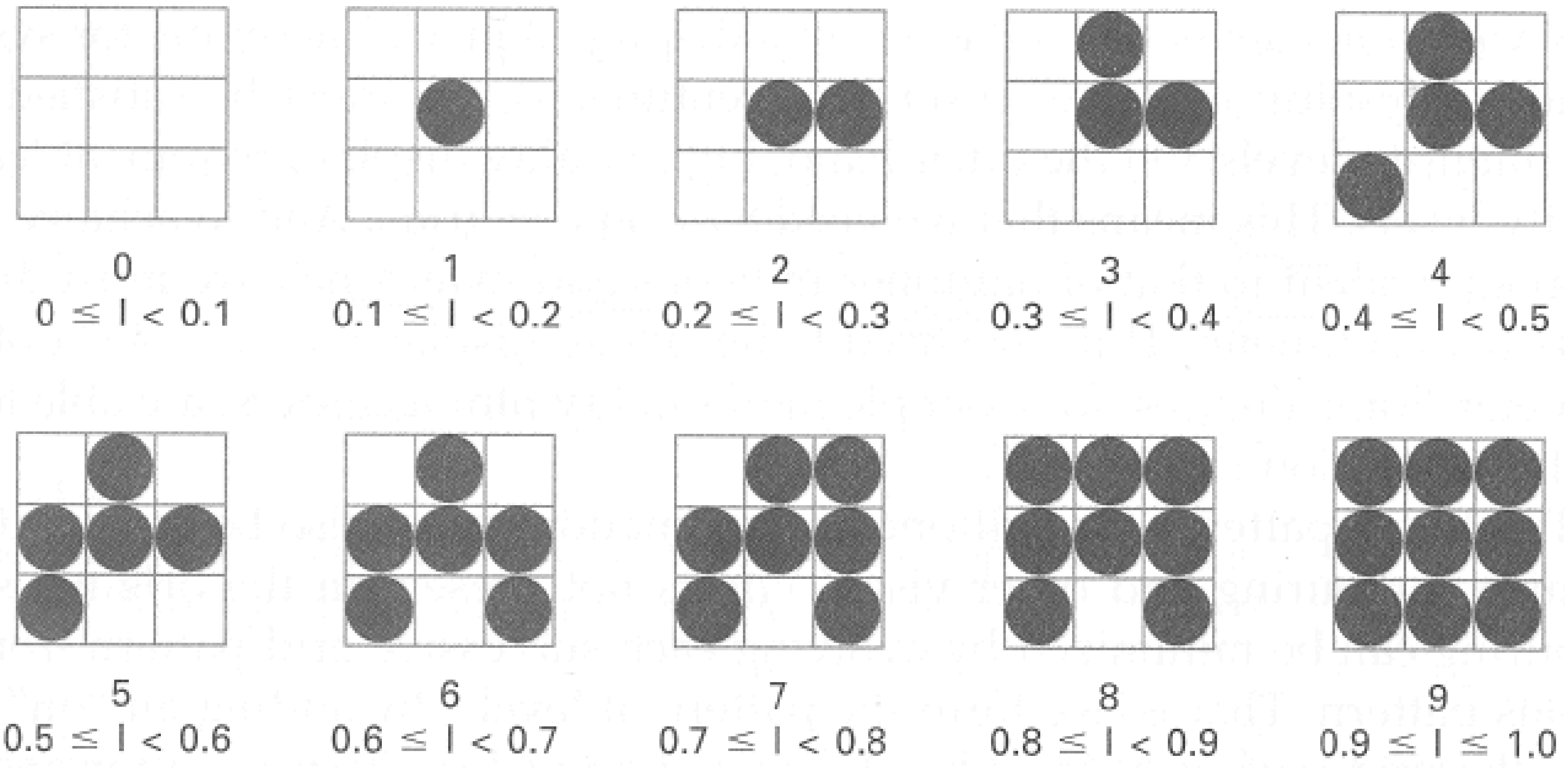


gray = 0.875...1.0



Clustered-dot Dithering

- Example:



Black and white dots here are arbitrary,
e.g., it can represent the intensity level or the order of turning on/off pixels

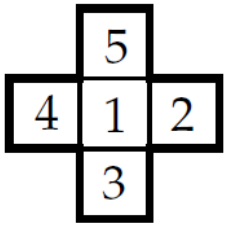
Clustered-dot Dithering

- If the consecutive thresholds are located in spatial proximity, it is called clustered-dot Dithering
- AM or FM halftoning?

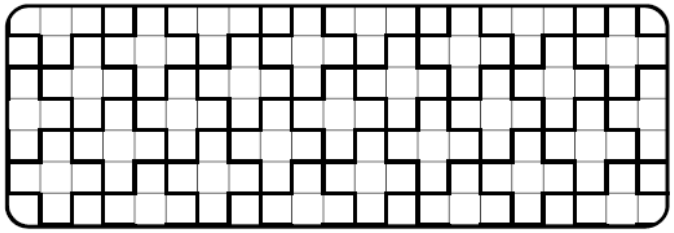
62	57	48	36	37	49	58	63
56	47	35	21	22	38	50	59
46	34	20	10	11	23	39	51
33	19	9	3	0	4	12	24
32	18	8	2	1	5	13	25
45	31	17	7	6	14	26	40
55	44	30	16	15	27	41	52
61	54	43	29	28	42	53	60

Clustered-dot Dithering

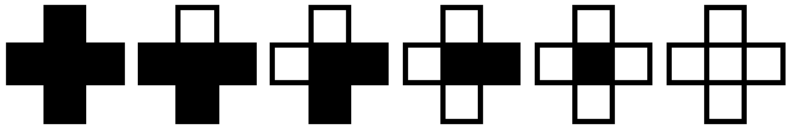
- Values in a threshold matrix are set to produce a single cell
  - User cluster of pixels to represent intensity
- The cell size depends on the input value
  - Trade spatial resolution for intensity resolution
- Properties
  - Dot shape, dot center of gravity motion, dot fusion, border to area ratio



Matrix with 5 values



The corresponding tiling



Micro dot selection for all six possible tones

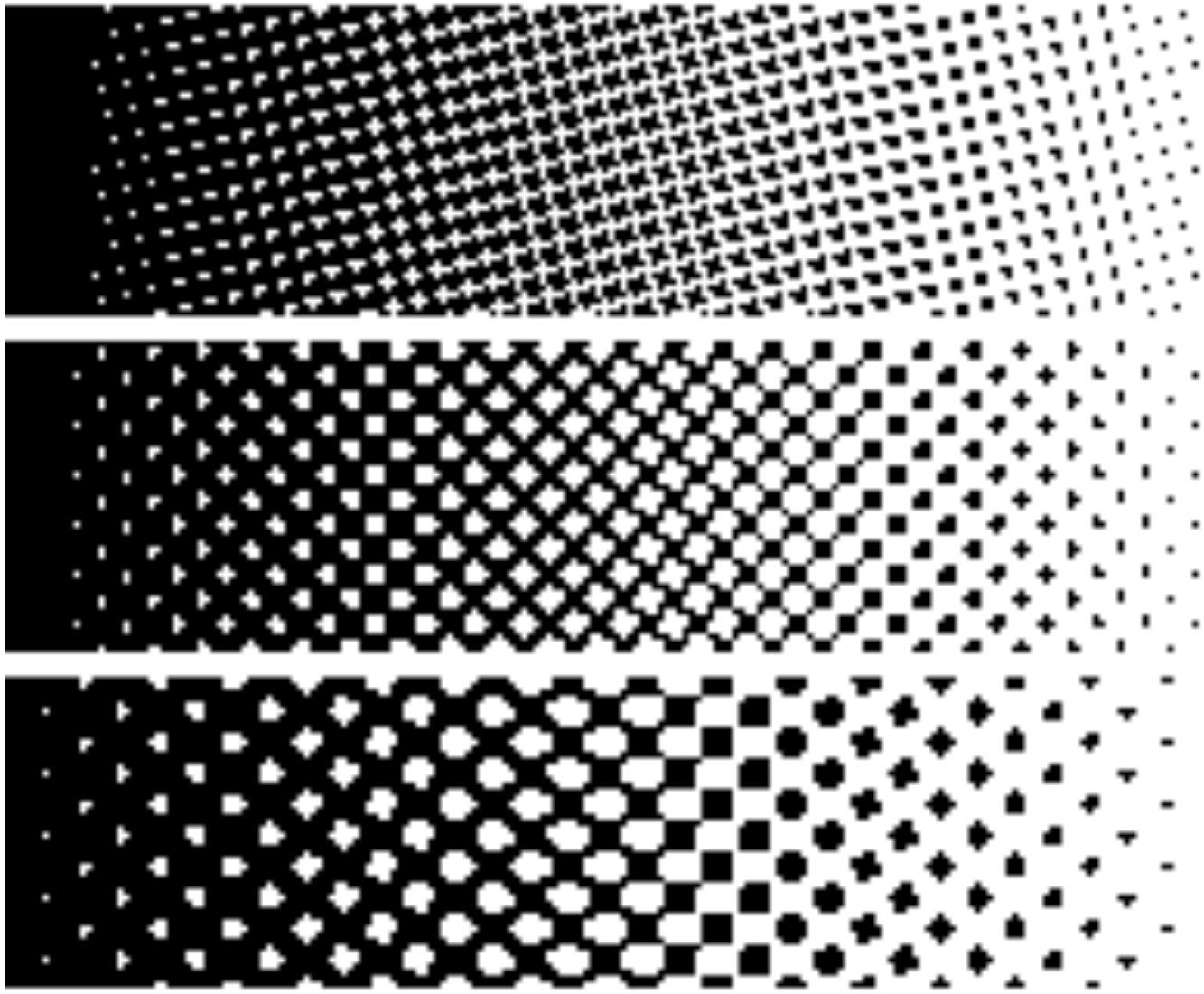
Clustered-dot Dithering Ramps

Matrix Size

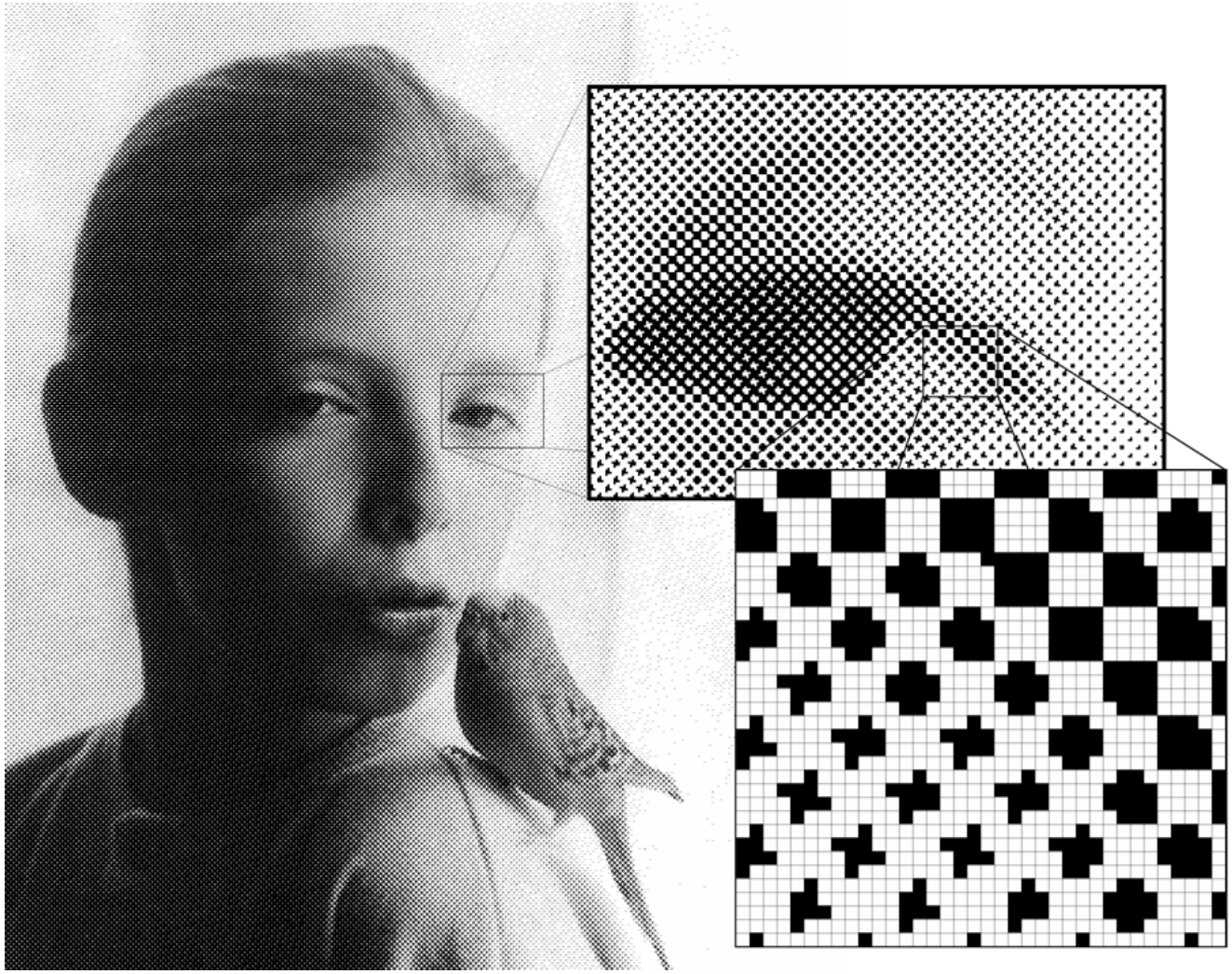
17

32

60



Clustered-dot Dithering Example



Clustered-dot Dithering - Supercells

- ~100 levels required for printing -> 10 x 10 matrix
- Supercell concatenates several threshold matrices
- Each threshold value is scaled/reassigned

base matrix M

4	8	1
7	9	5
3	6	2

concatenation of 9 copies of M

4	8	1	4	8	1	4	8	1
7	9	5	7	9	5	7	9	5
3	6	2	3	6	2	3	6	2
4	8	1	4	8	1	4	8	1
7	9	5	7	9	5	7	9	5
3	6	2	3	6	2	3	6	2
4	8	1	4	8	1	4	8	1
7	9	5	7	9	5	7	9	5
3	6	2	3	6	2	3	6	2

M_s, after scaling threshold values

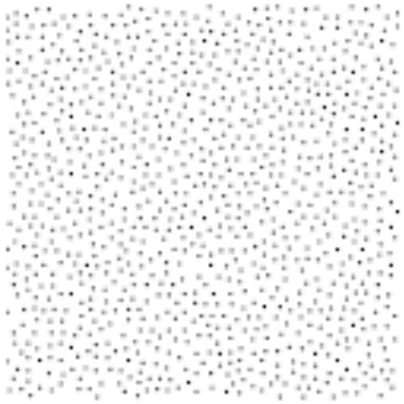
28	64	1	35	71	8	30	66	3
55	73	37	62	80	44	57	75	39
19	46	10	26	53	17	21	48	12
31	67	4	29	65	2	33	69	6
58	76	40	56	74	38	60	78	42
22	49	13	20	47	11	24	51	15
34	70	7	32	68	5	36	72	9
61	79	43	59	77	41	63	81	45
25	52	16	23	50	14	27	54	18

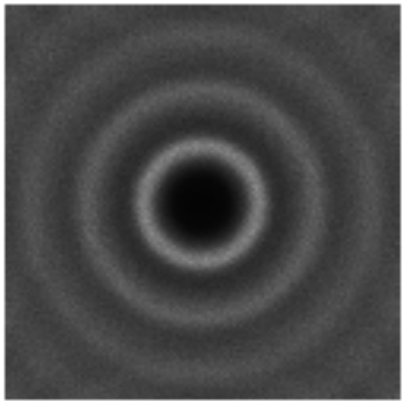
Dispersed-dot Dithering

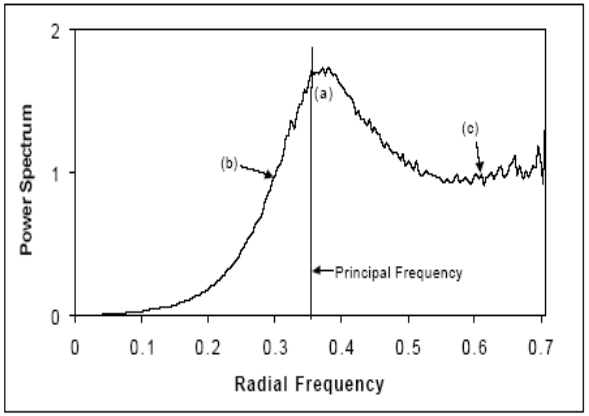
- Frequency Modulation (FM) halftoning
- The quality defined by the pattern of micro dots **for each** possible input constant value
- Each pattern should be as evenly dispersed as possible
  - **blue noise distribution**
- The distribution of threshold values inside the matrix is crucial to obtain a good distribution of micro dots

Blue Noise

- A pleasant distribution of points
  - I Points are away from each other with a nearly regular distance
  - II No alignment
  - In frequency domain: a preferred frequency, no preferred direction
- Blue noise (in frequency domain)
  - Radially symmetric
  - Null values for low frequencies, strong annulus with the frequency corresponding to the preferred distance period, white noise for high frequencies





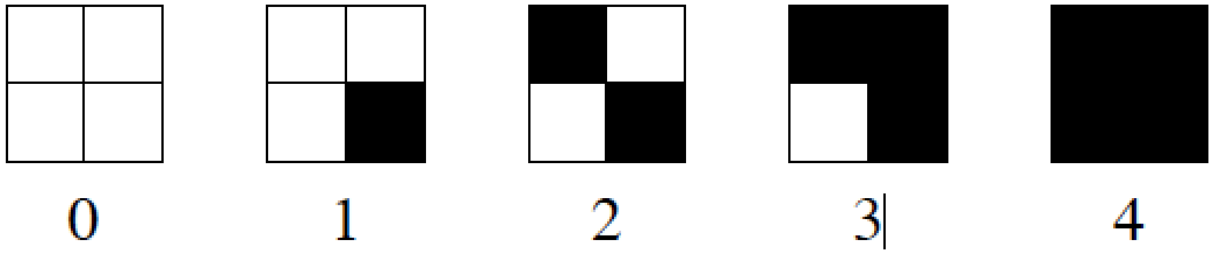


Bayer Ordered Dither Patterns

- The threshold matrix can be represented by an “index matrix”
  - The index matrix determines the order in which dots are “turned on” (change from white to black) as the images becomes darker

$$I_2(i,j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

  - Pixels are turned on (white to dark) in the following order:



Bayer Ordered Dither Patterns

- Bayer's optimum index matrix (1973) can be defined recursively

$$I_2(i,j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

$$I_{2n} = \begin{bmatrix} 4 * I_n + 1 & 4 * I_n + 2 \\ 4 * I_n + 3 & 4 * I_n \end{bmatrix}$$

- Examples

1	2
3	0

2×2

5	9	6	10
13	1	14	2
7	11	4	8
15	3	12	0

4×4

21	37	25	41	22	38	26	42
53	5	57	9	54	6	58	10
29	45	17	33	30	46	18	34
61	13	49	1	62	14	50	2
23	39	27	43	20	36	24	40
55	7	59	11	52	4	56	8
31	47	19	35	28	44	16	32
63	15	51	3	60	12	48	0

8×8

Bayer Ordered Dither Patterns

B_4

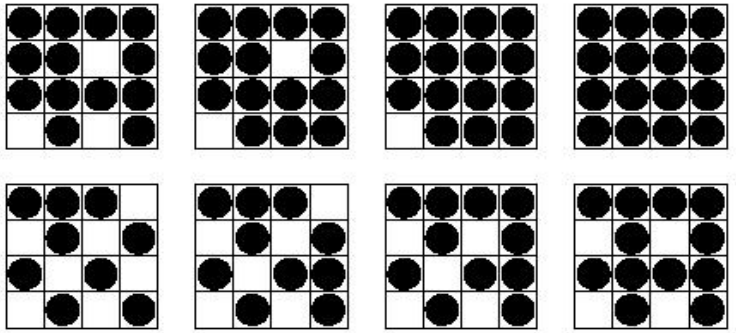
0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

B_3

3	6	2
7	0	5
4	8	1

B_2

0	2
3	1



Exercise

Bayer Ordered Dither Patterns

B_4

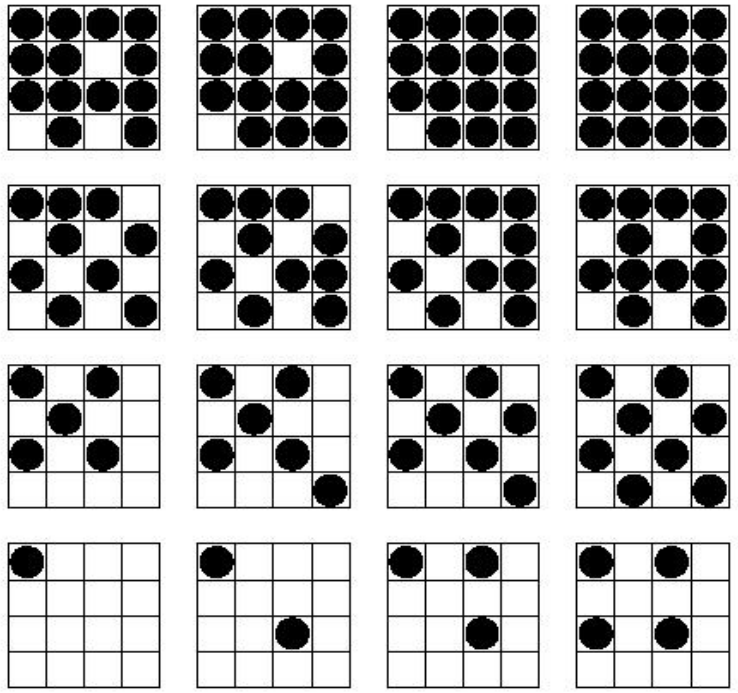
0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

B_3

3	6	2
7	0	5
4	8	1

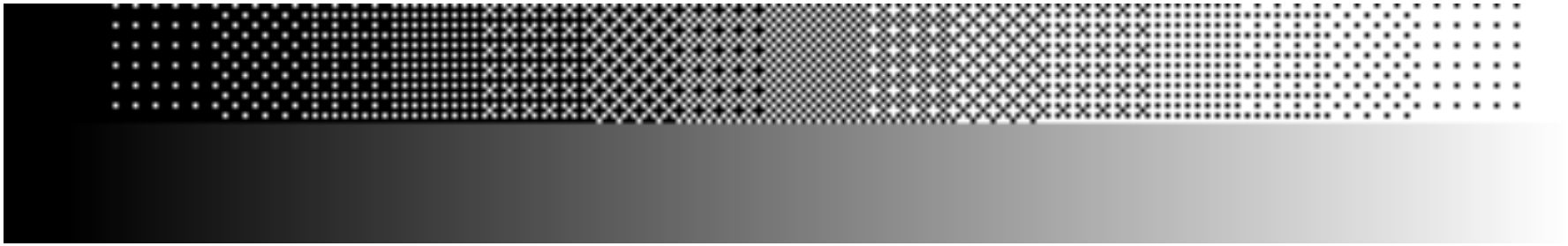
B_2

0	2
3	1



Bayer Ordered Dither Patterns

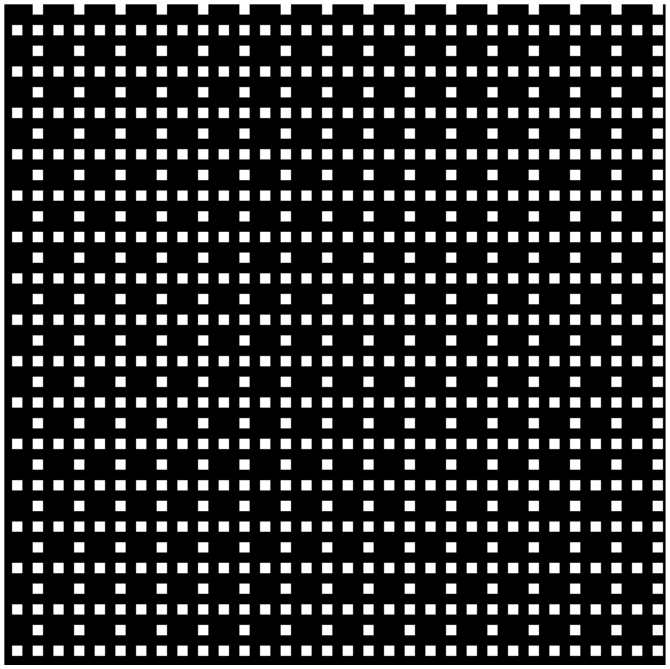
- Characteristic 15 patterns of the 4x4 ordered dithering matrix
  - Thresholds should be distributed as uniformly as possible compared with clustered dithering



Using multiple 4x4 dithering matrix to represent the bottom row (continuous color)

Bayer Ordered Dither Patterns

- Optimal in terms of point to point distance, but produce strong alignments
  - These alignments can be seen in Fourier domain as strong values for some frequencies



Distribution for a Bayer matrix of size 4x4

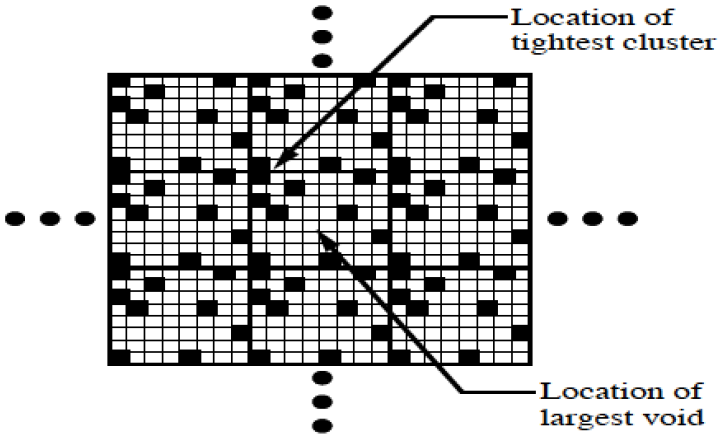
Fourier Amplitude Spectrum

Void-and-Cluster & Blue Noise Mask Dithering

- [Ulichney, 1993]
- Breaks Bayer alignments (periodic pattern)
- Produces dot patterns that come close to blue noise
- Goal:
  - Determine threshold values of a rectangular matrix that produces blue noise pattern for each tone level
- Based entirely in the spatial domain

Void-and-Cluster & Blue Noise Mask Dithering

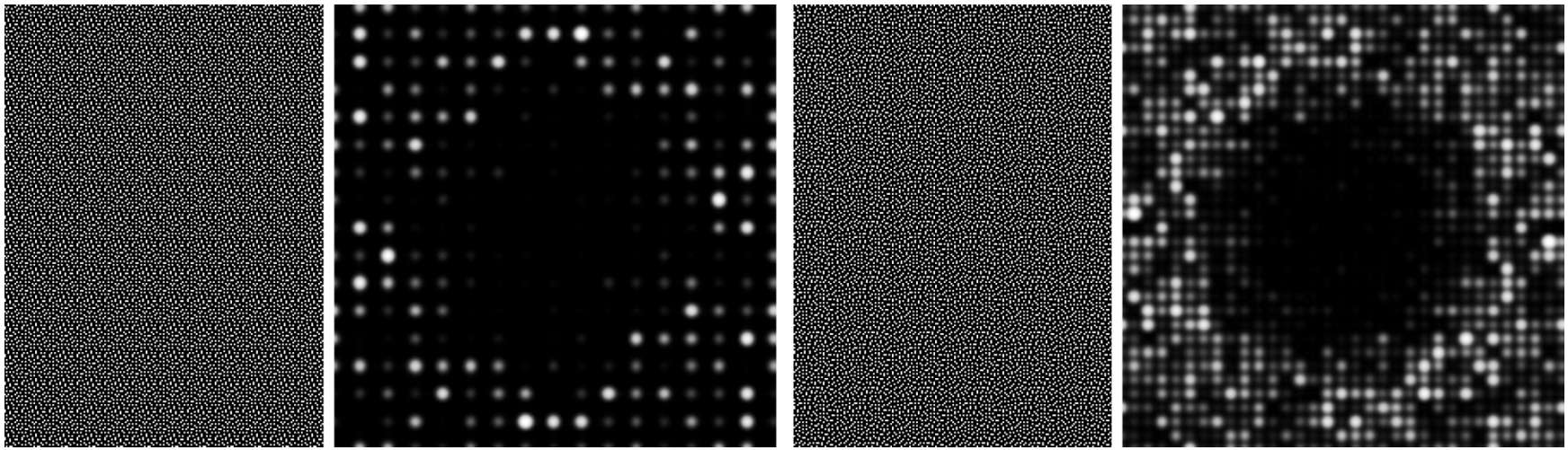
- Swapping pixels



c. swap hole closest to center of largest void with dot closest to center of tightest cluster.

Void-and-Cluster & Blue Noise Mask Dithering

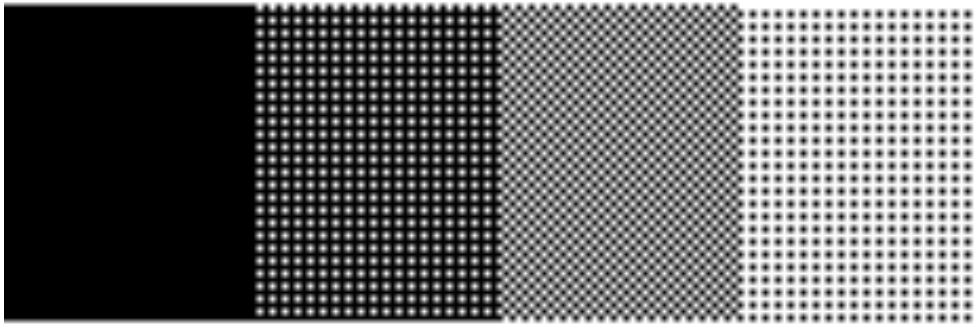
- Breaks Bayer alignments (periodic pattern)
- Produces dot patterns that come close to blue noise
- Goal:
  - Determine threshold values of a rectangular matrix that produces blue noise pattern for each tone level



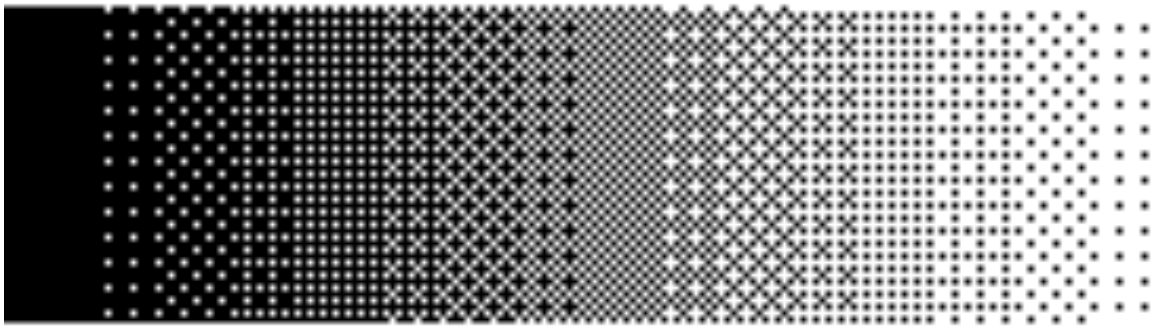
Void-and-cluster distribution and Fourier Amplitude Spectrum
Matrix size 16x16 and 32x32

Example

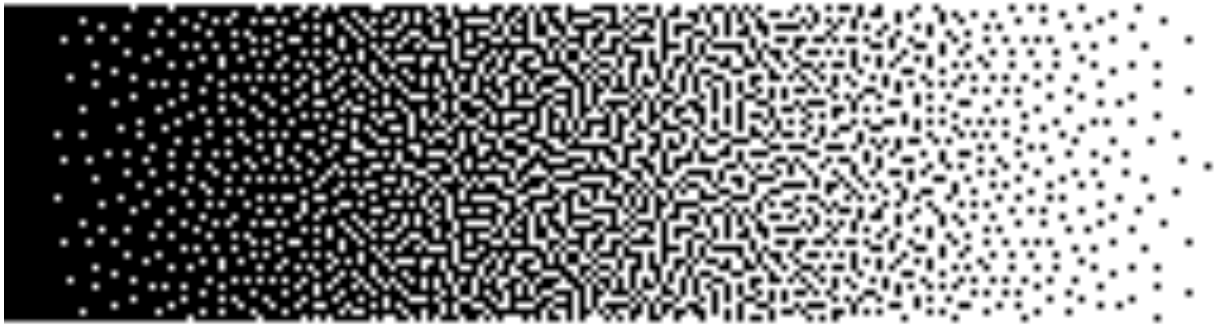
Bayer 2



Bayer 4

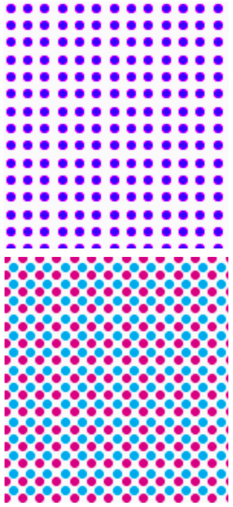


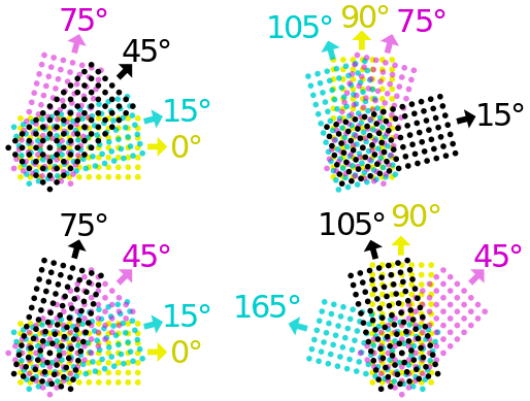
void-and-cluster 32



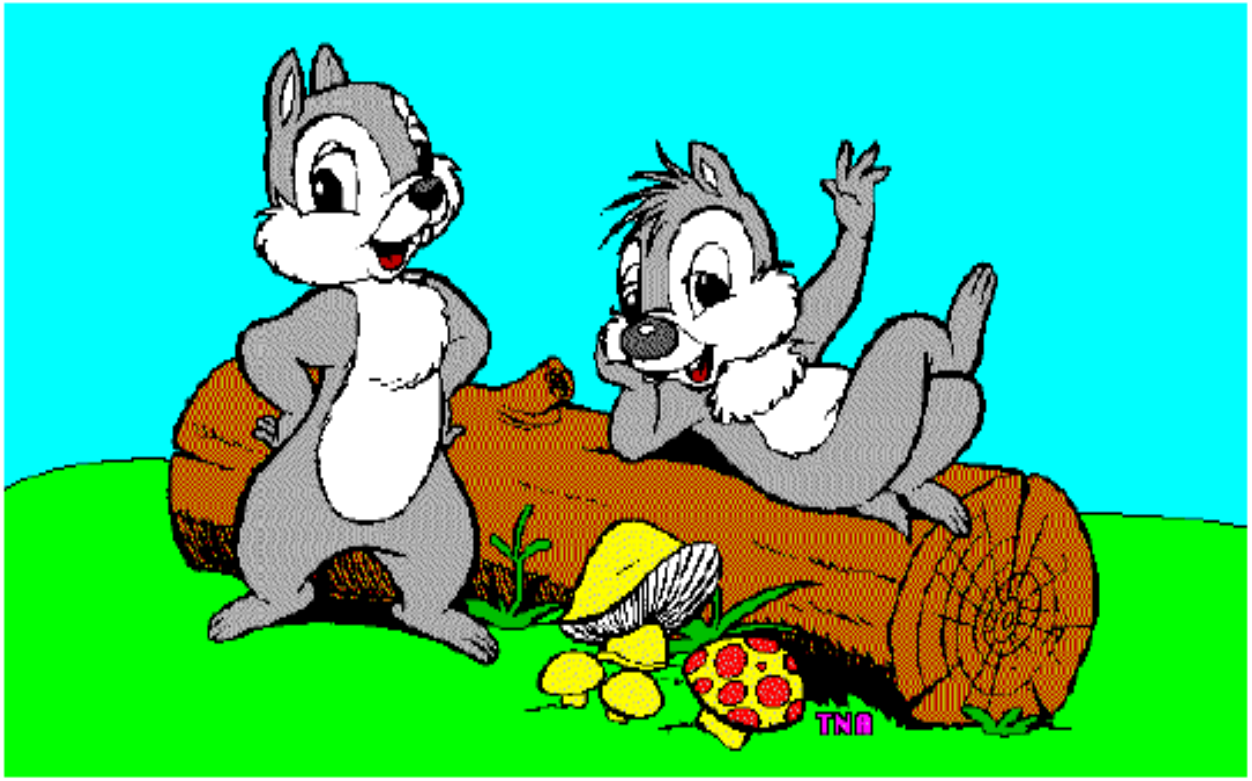
Clustered-dot Dithering - Color

- RGB converted to CMYK, each channel printed separately
- Dot-on-dot
  - Clusters centered at the same spatial position
  - Side-by-side drops produce different spectra than overlapping drops
  - Requires color calibration
- Dot-off-dot
  - 4 color matrices are shifted in xy to minimize overlap
- Rotated dot
  - CMYK layers have relative orientations



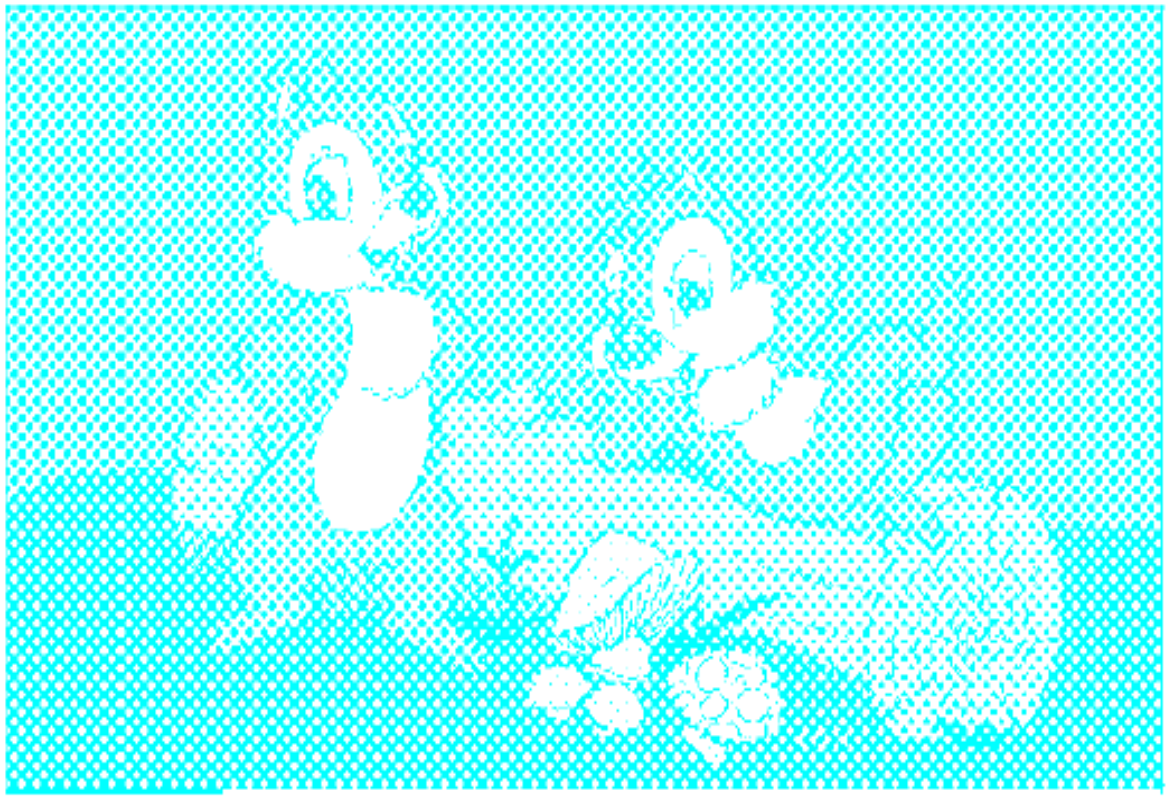


Rotated Dot Color Halftoning

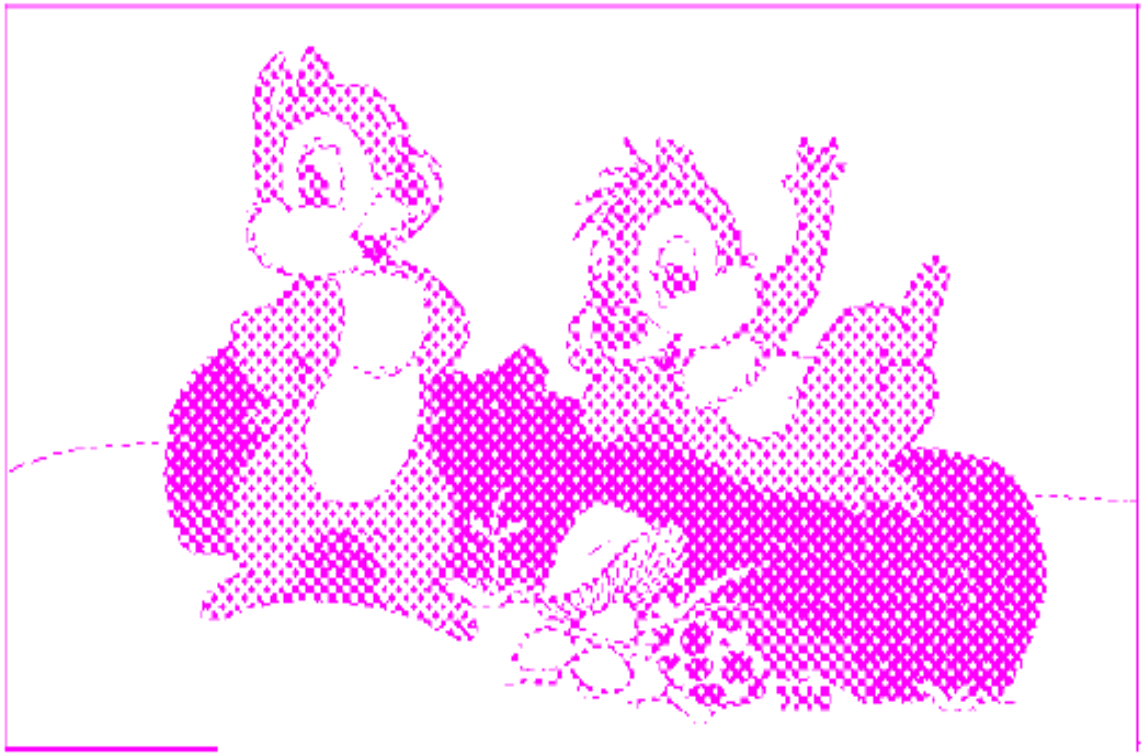


Rotated Dot Color Halftoning

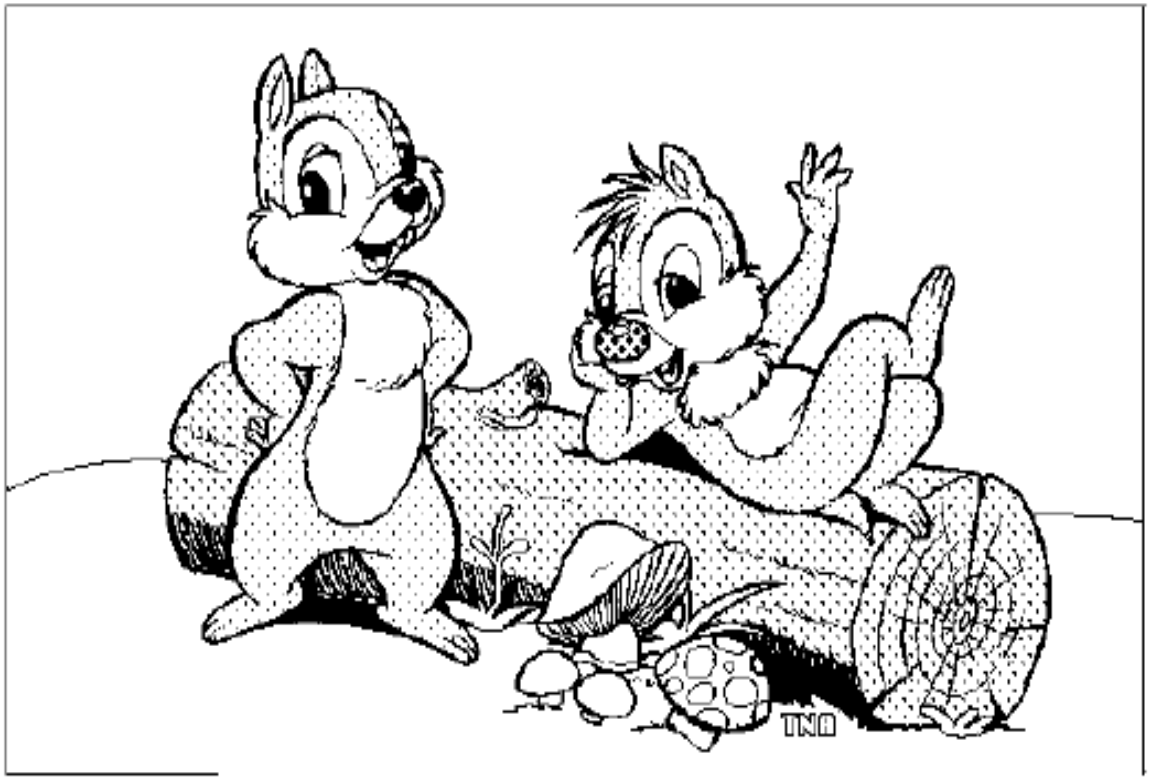
Rotated Dot Color Halftoning



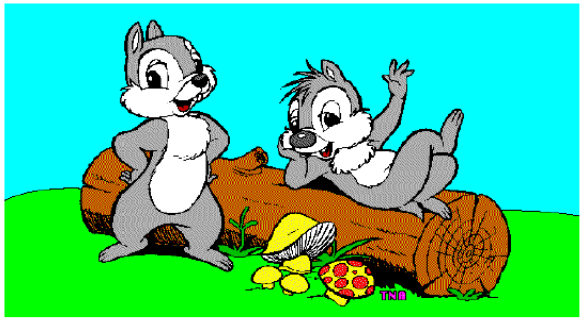
Rotated Dot Color Halftoning

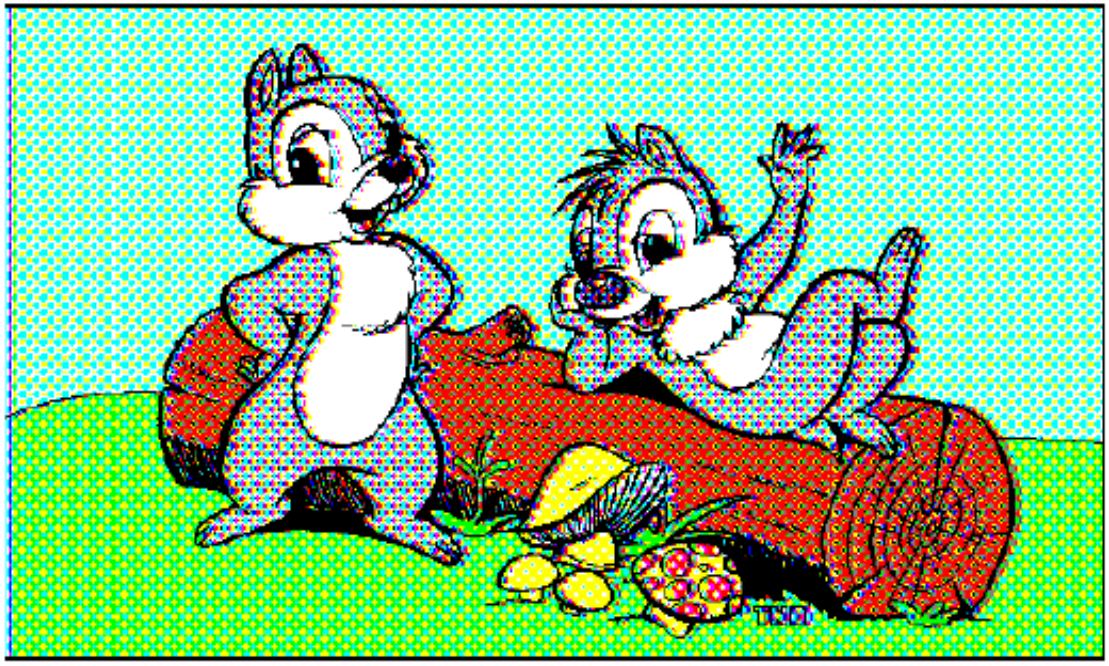


Rotated Dot Color Halftoning



Rotated Dot Color Halftoning





Class II: Neighborhood Processes

- Error diffusion
  - [Floyd-Steinberg 1976, Jarvis et al. 1976, Eschbach & Knox 1991, Shiau & Fan 1994, Marcu & Abe 1996, Ostromoukhov 2001, Zhou & Fang 2003, Hwang et al. 2004, Kwak et al. 2006]
- Each pixel depends on multiple pixels
  - Extra computations
  - Class II ⇨ **moderately fast**
- Weak uniformity over the image
  - Loss of detail
  - Tradeoff between good tones and sharp edges
  - Class II ⇨ **average overall quality**

Floyd-Steinberg Error Diffusion

- With this method, the average quantization error is reduced by propagating the error from each pixel to some of its neighbors in the scan order

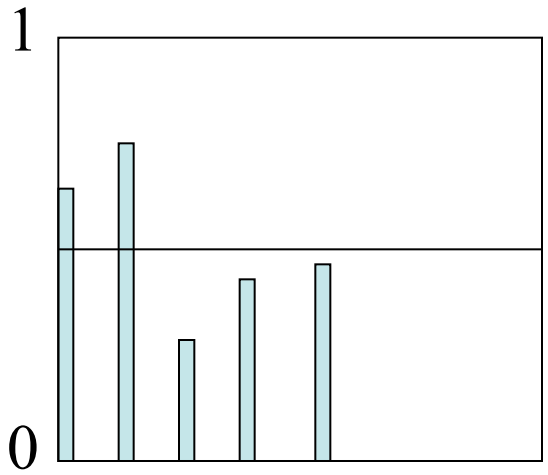
Floyd-Steinberg Error Diffusion

- With this method, the average quantization error is reduced by propagating the error from each pixel to some of its neighbors in the scan order

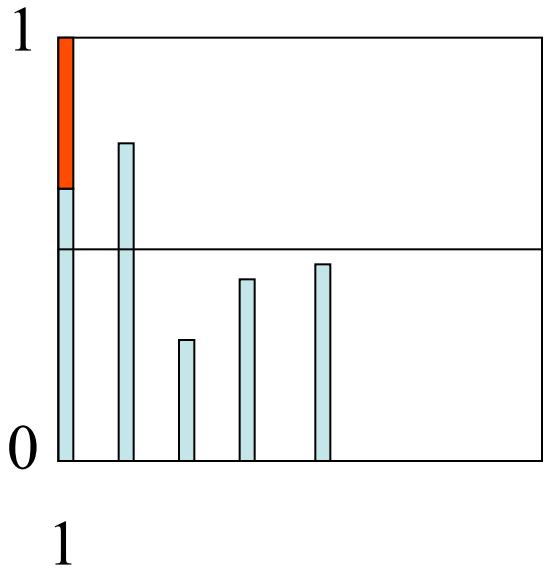
original image

original image after the application of Floyd-Steinberg dithering

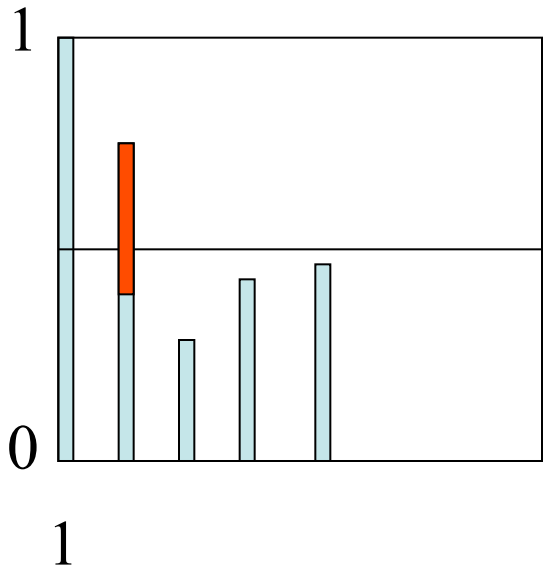
1D Error Diffusion



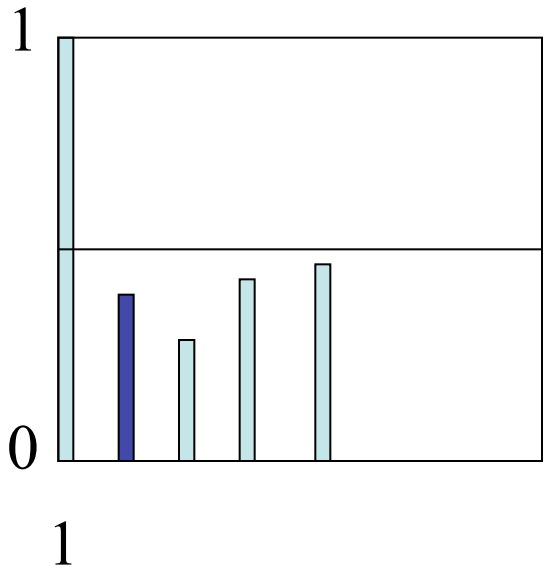
1D Error Diffusion



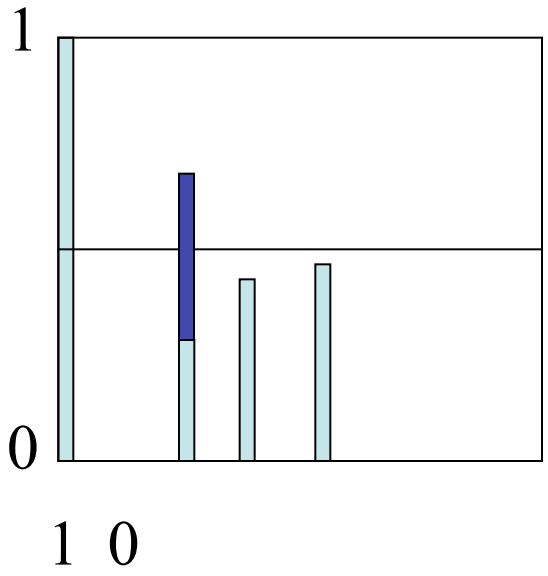
1D Error Diffusion



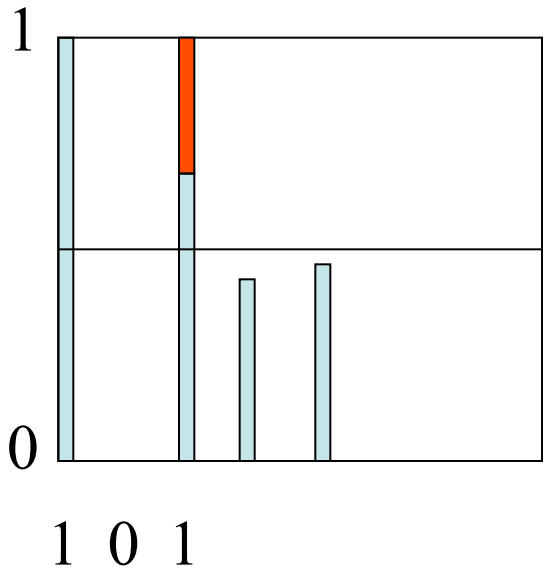
1D Error Diffusion



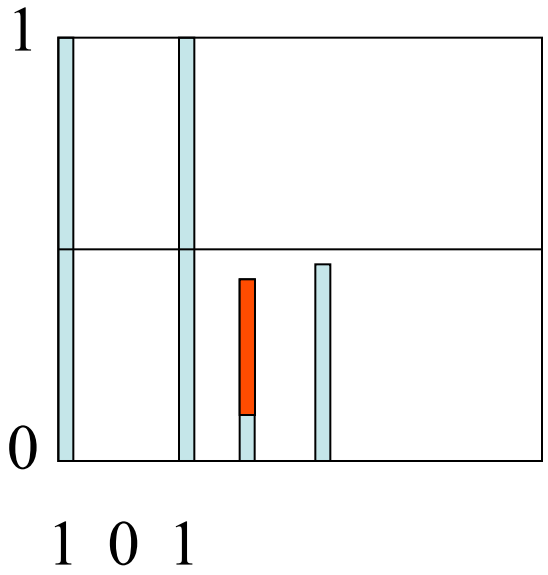
1D Error Diffusion



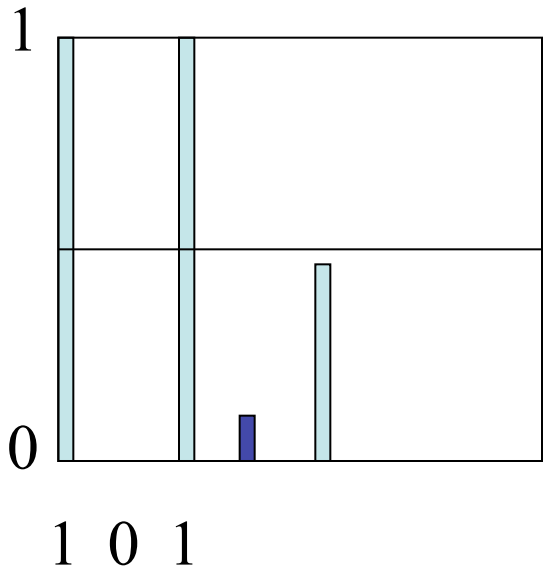
1D Error Diffusion



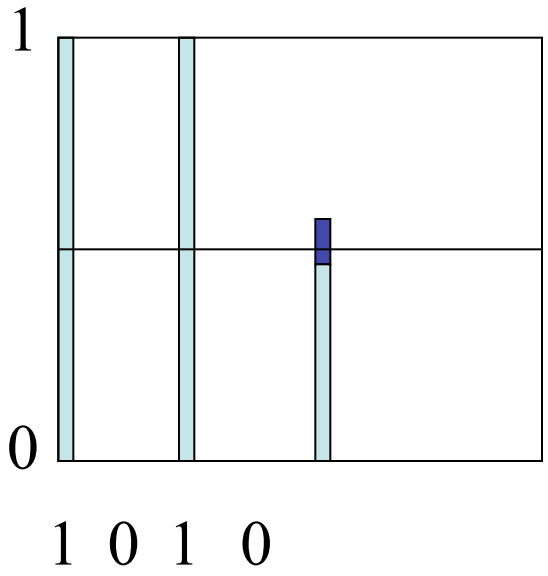
1D Error Diffusion



1D Error Diffusion

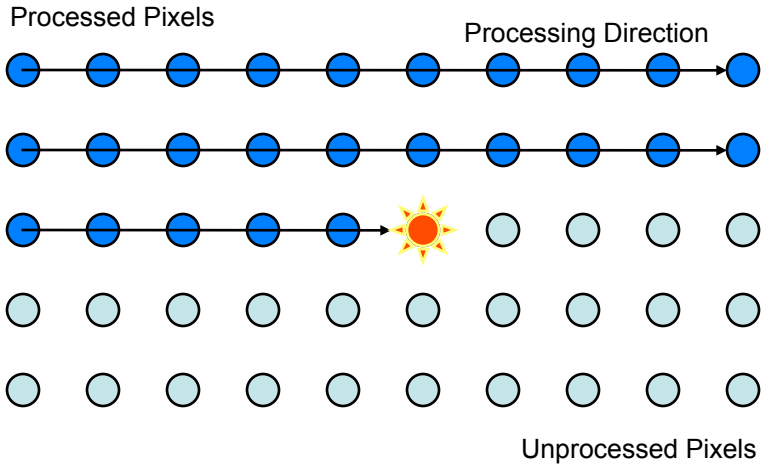


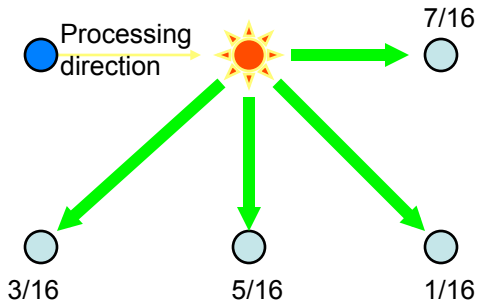
1D Error Diffusion



Floyd-Steinberg Error-Diffusion Algorithm (1975)

- Process pixels in order of scanlines
- Compare input(x,y) with threshold(x,y)=.5
- Distribute quantization error to unprocessed pixels





Note that the error propagation weights must sum to one

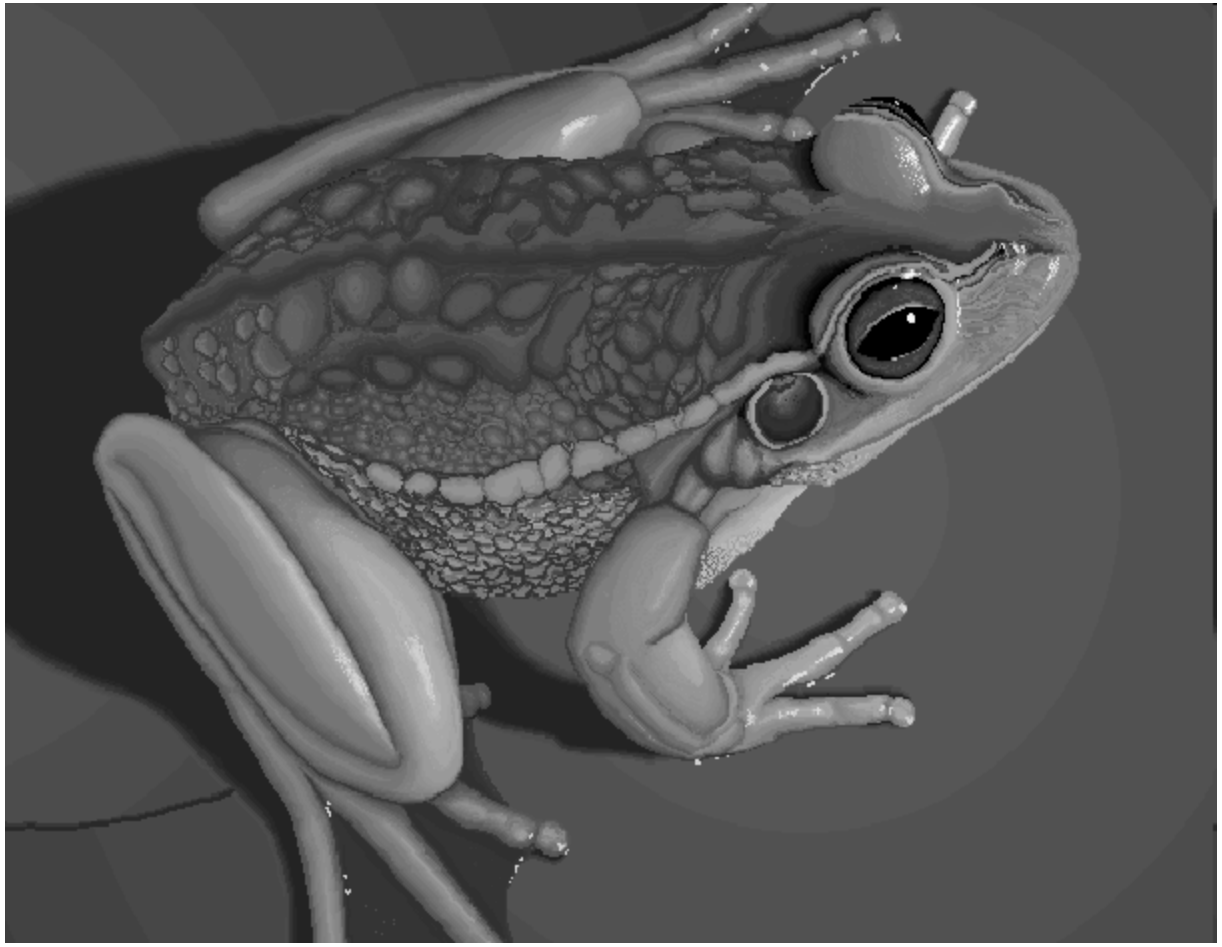
Error Diffusion Algorithm

```
Set AccErr[] to zero;
For each pixel in the image scanning from left to right:
   value= pixelValue(x,y) + AccErr[x,y];
   if (value > WHITE/2)
      setPixel(x,y, WHITE);
      Error = value - WHITE;
   else
      setPixel(x,y, BLACK);
      Error = value - BLACK;

   AccErr[x+1, y]    += 7/16 * Error;
   AccErr[x-1, y+1]  += 3/16 * Error;
   AccErr[x  , y+1]  += 5/16 * Error;
   AccErr[x+1, y+1]  += 1/16 * Error;
```

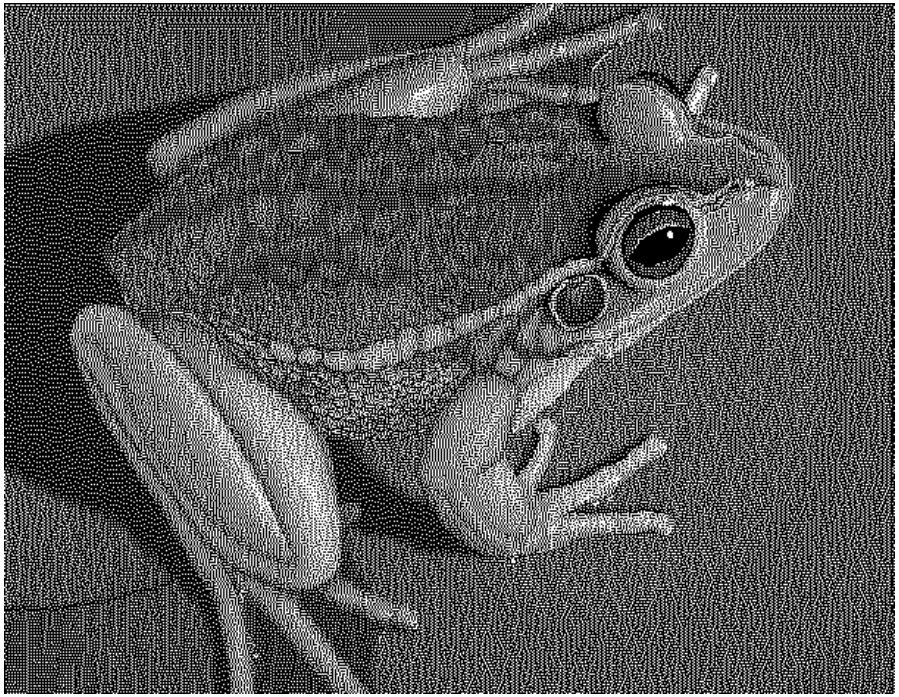
Examples

Original Picture

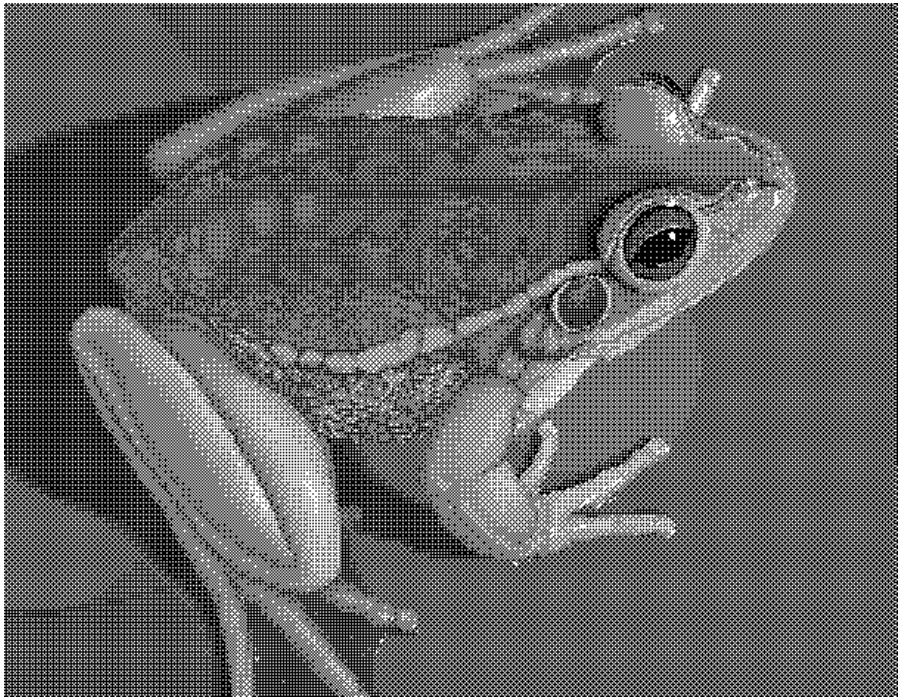


Examples

Error Diffusion

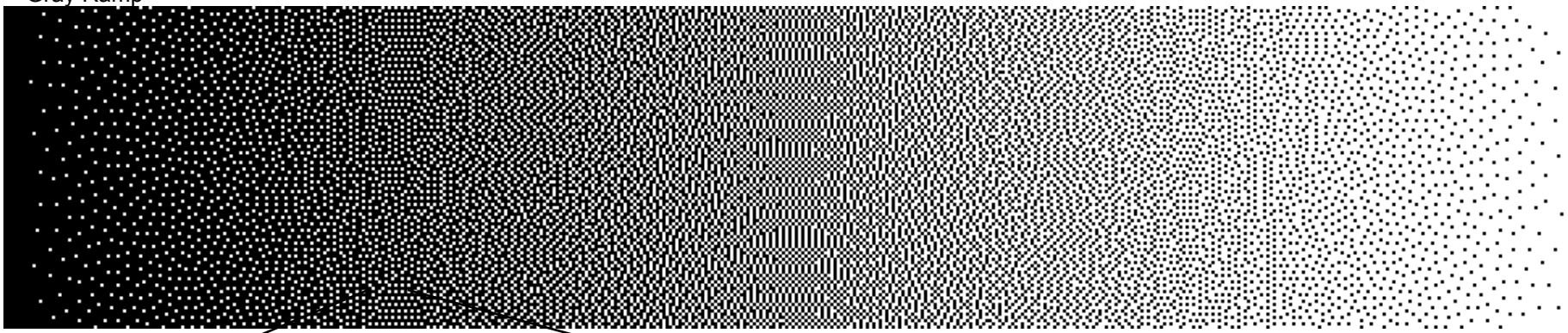


Dithering

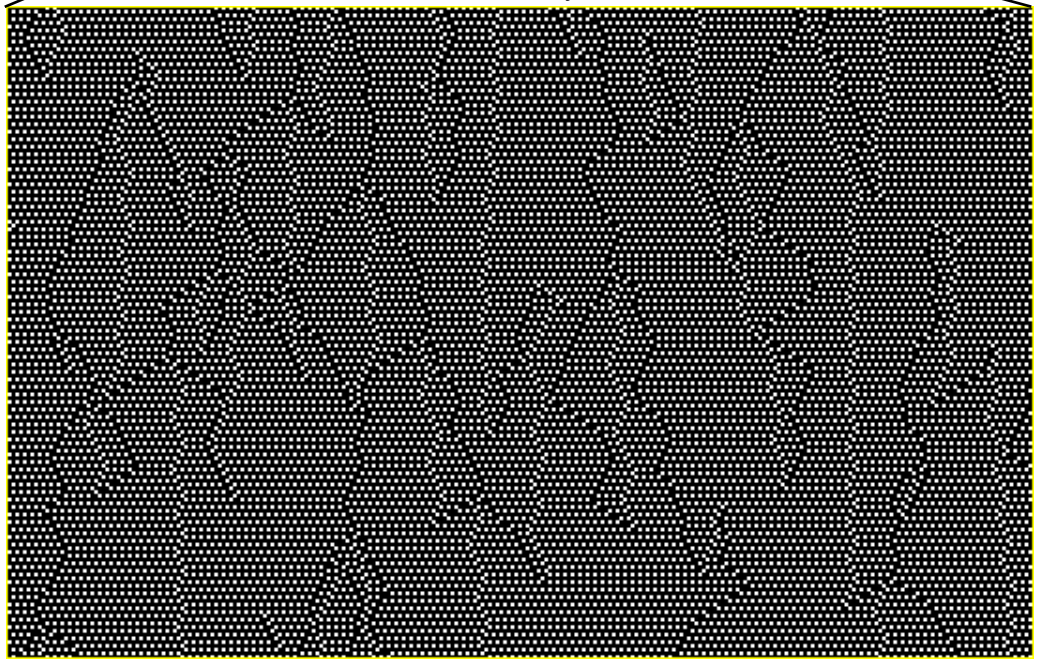


Floyd-Steinberg Error-Diffusion Algorithm

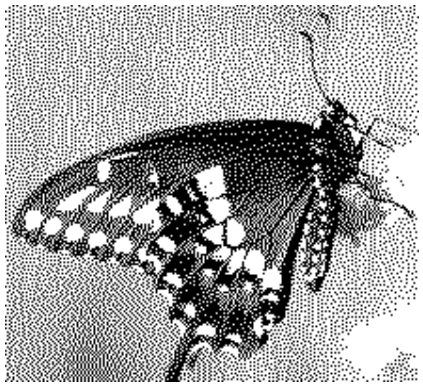
Gray Ramp



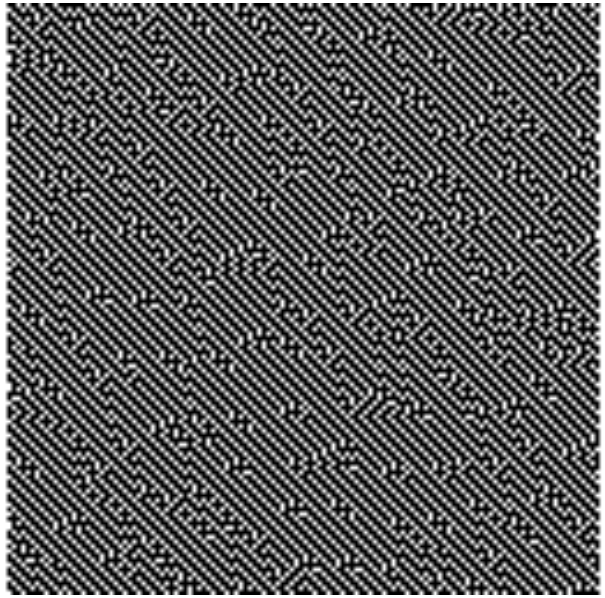
Intolerable Artifacts at Certain Intensity Levels, Out of Control



Sample Image

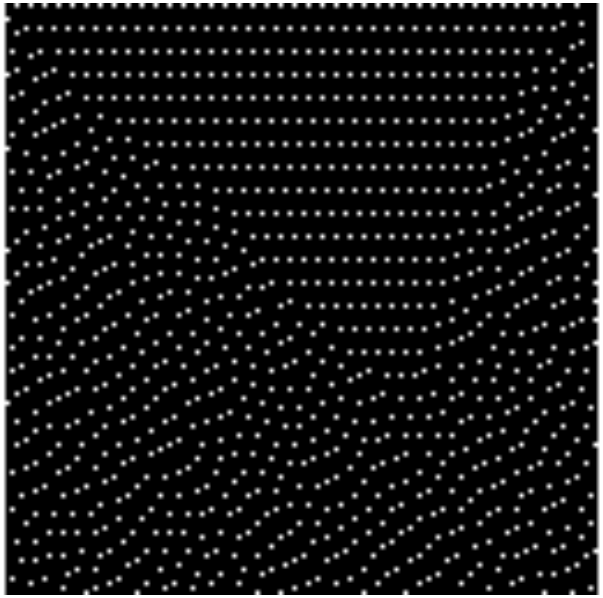


Floyd-Steinberg Error-Diffusion Artifacts



maze-like structures

worm-like alignments

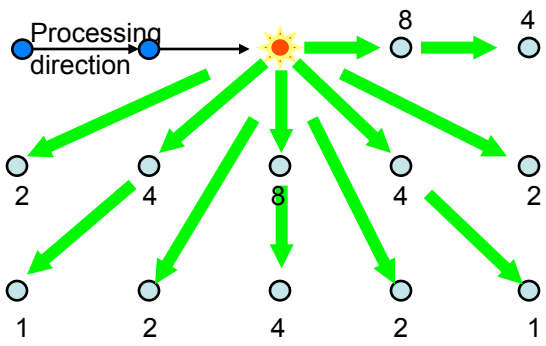


edges instabilities

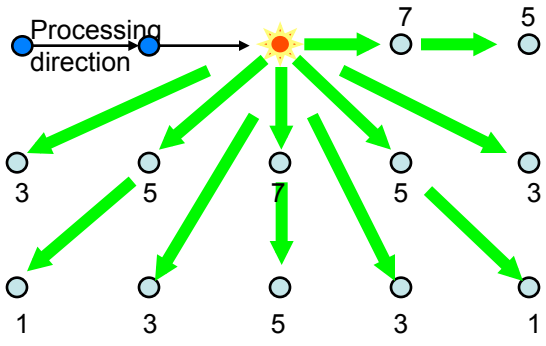
Sophistication of Error-Diffusion Algorithms

- Different error distribution coefficients
- Processing paths: serpentine, space-filling curves
- Threshold modulation: control of edge enhancement

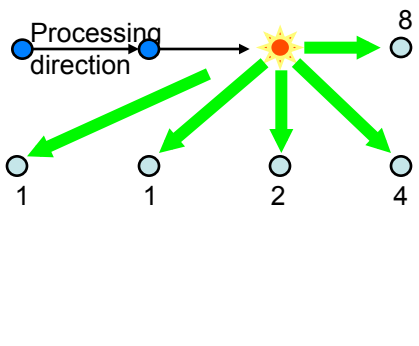
Stucki (1981)



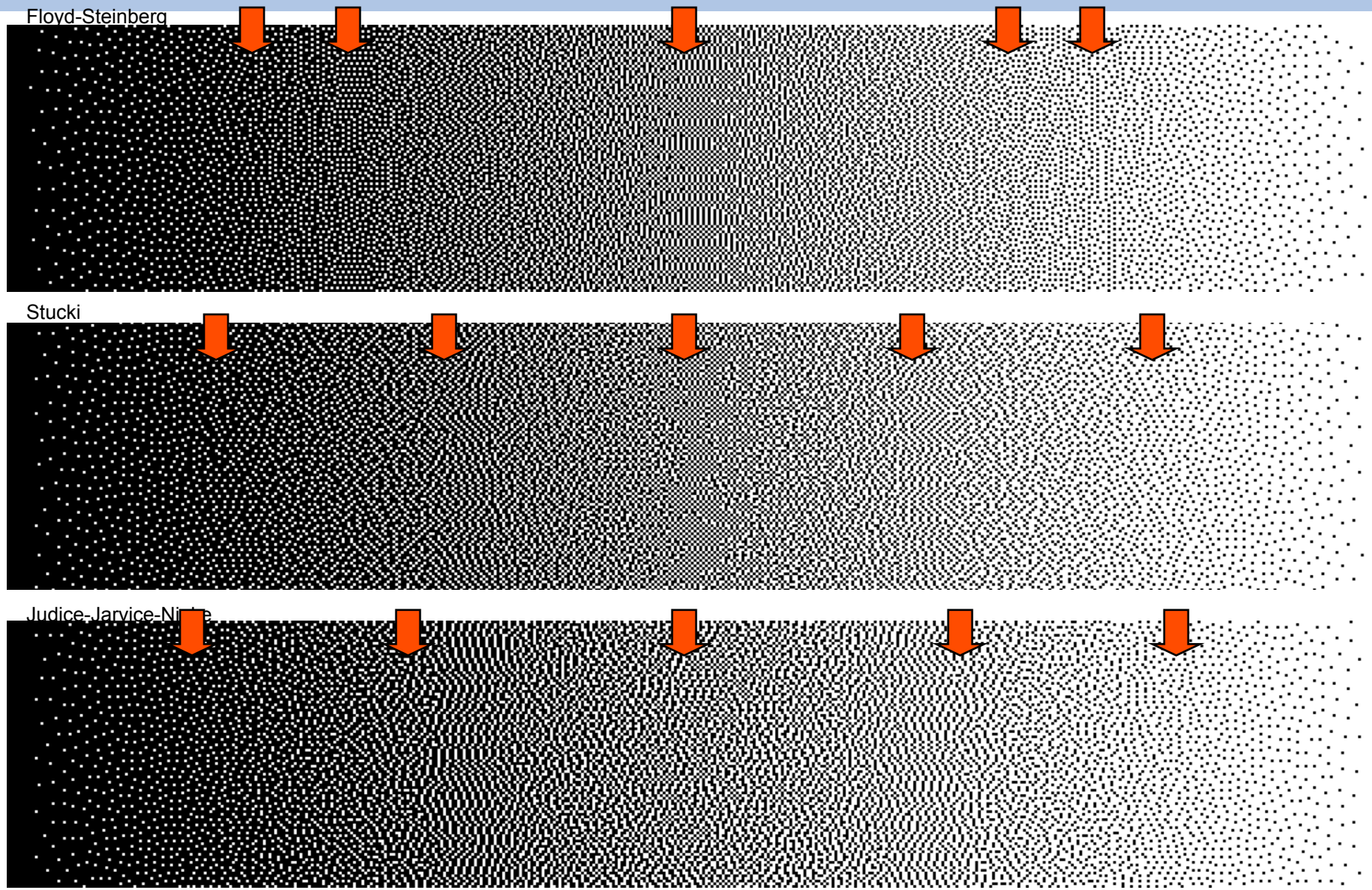
Judice-Jarvice-Ninke (1976)



Shiau-Fan (1996)



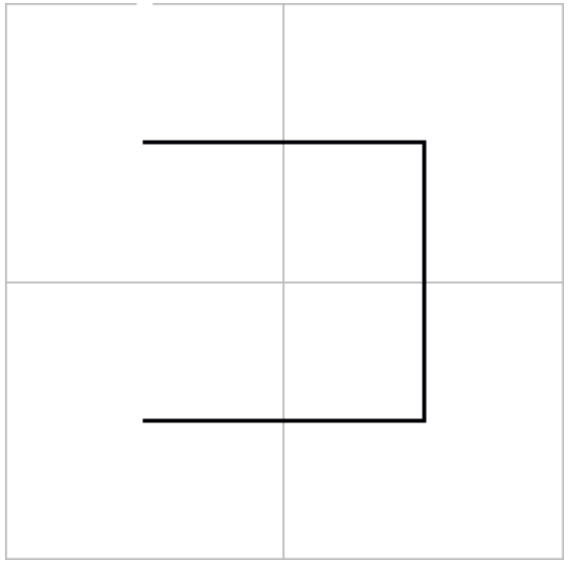
Different Error-Diffusion Algorithms

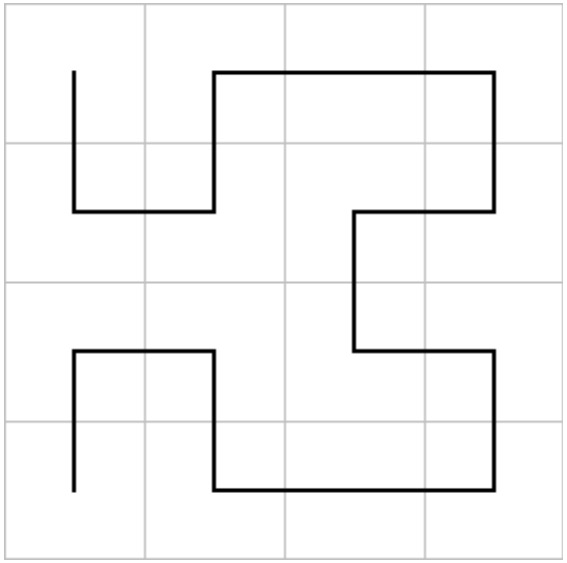


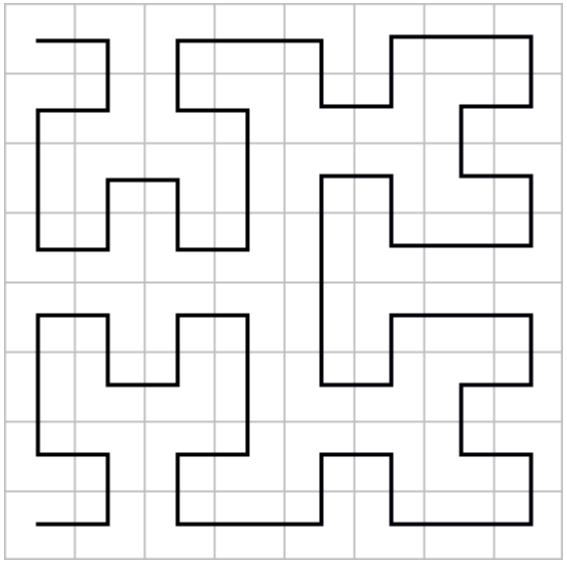
Space Filling Curves

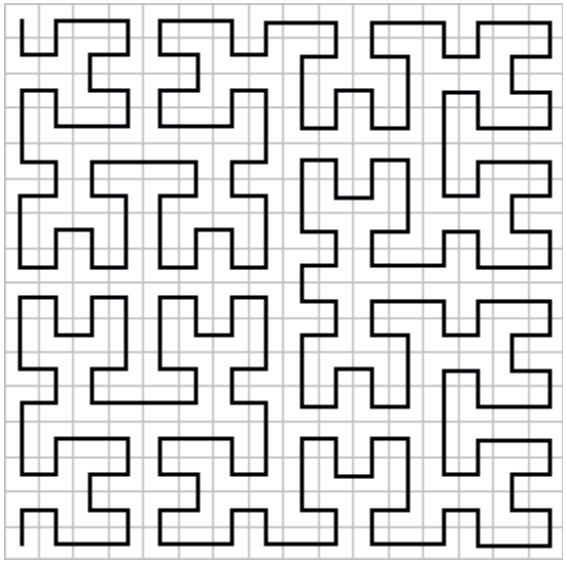
Hilbert curve

(1-4)



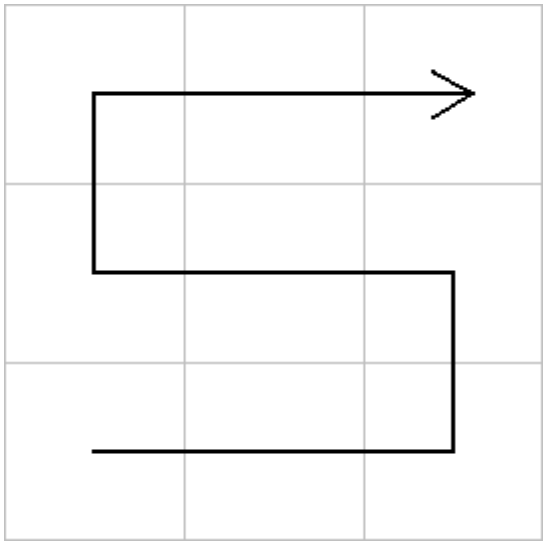


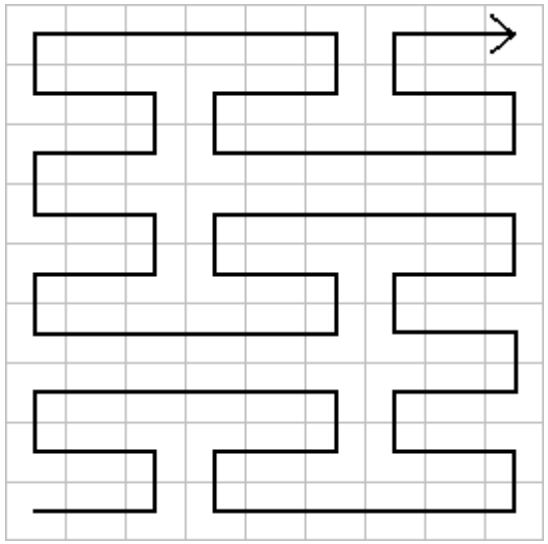


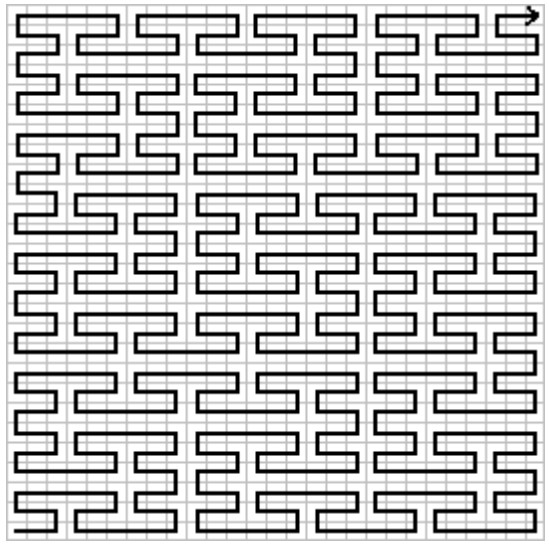


Space Filling Curves

- Peano curve







Space Filling Curves

break regularity in raster order,
but introduce textured noise

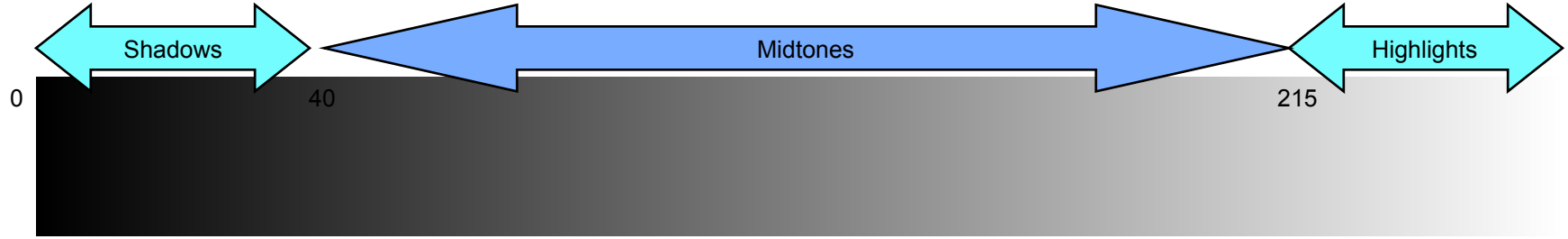
Left to right

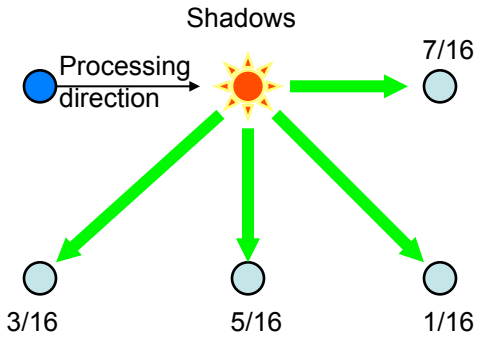
Hilbert Curve

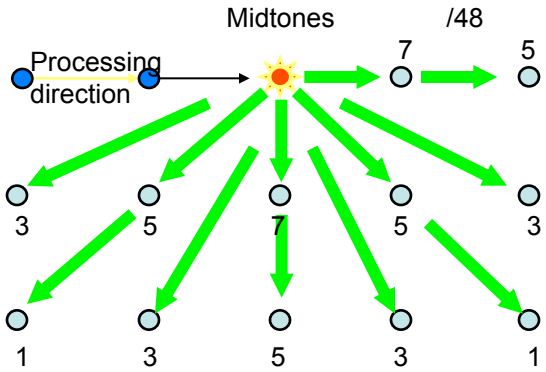
Peano Curve

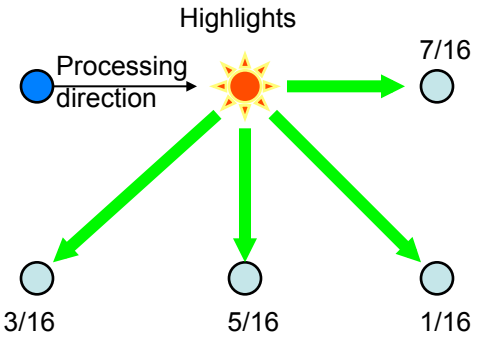
Tone-Dependent Weights

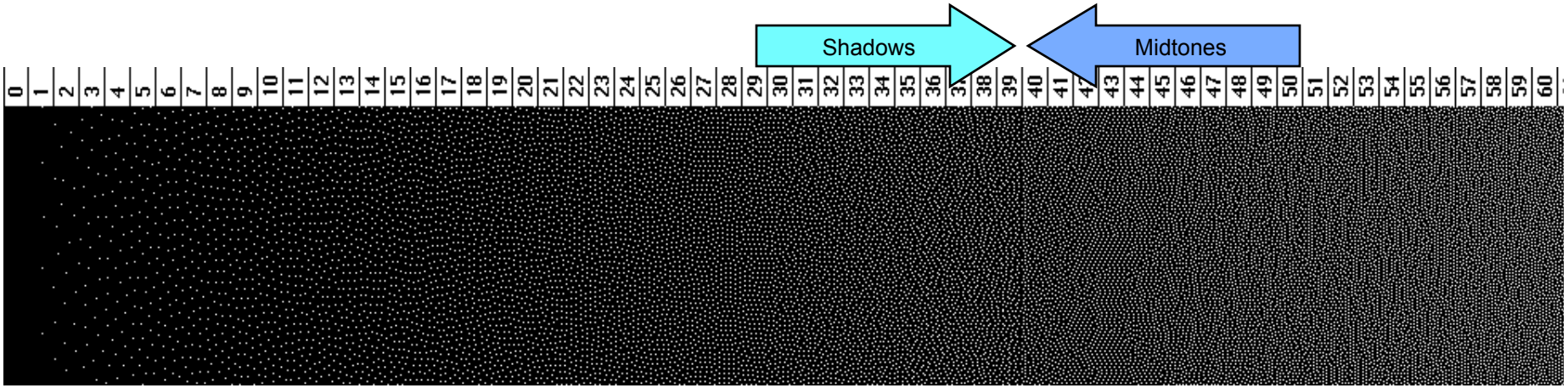
Eschbach





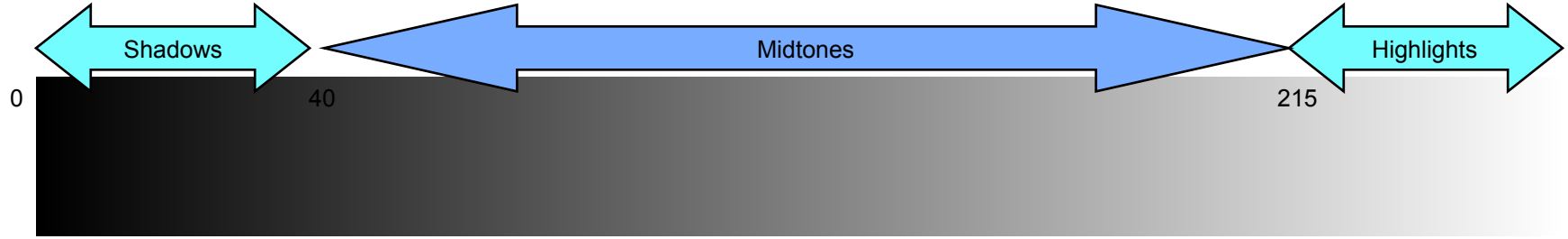


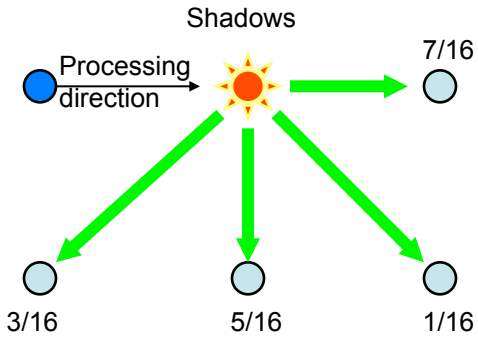


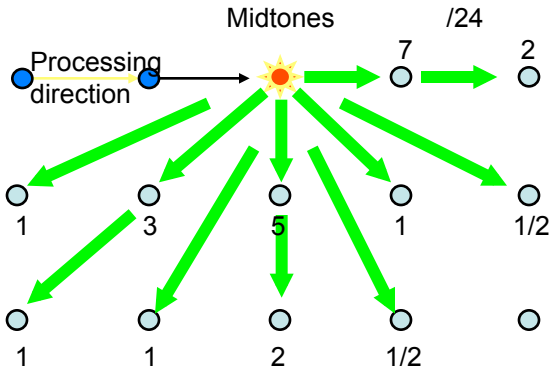


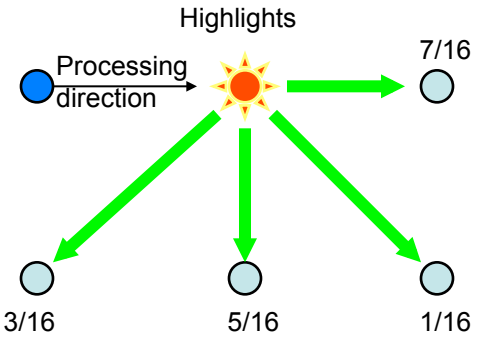
Tone-Dependent Weights

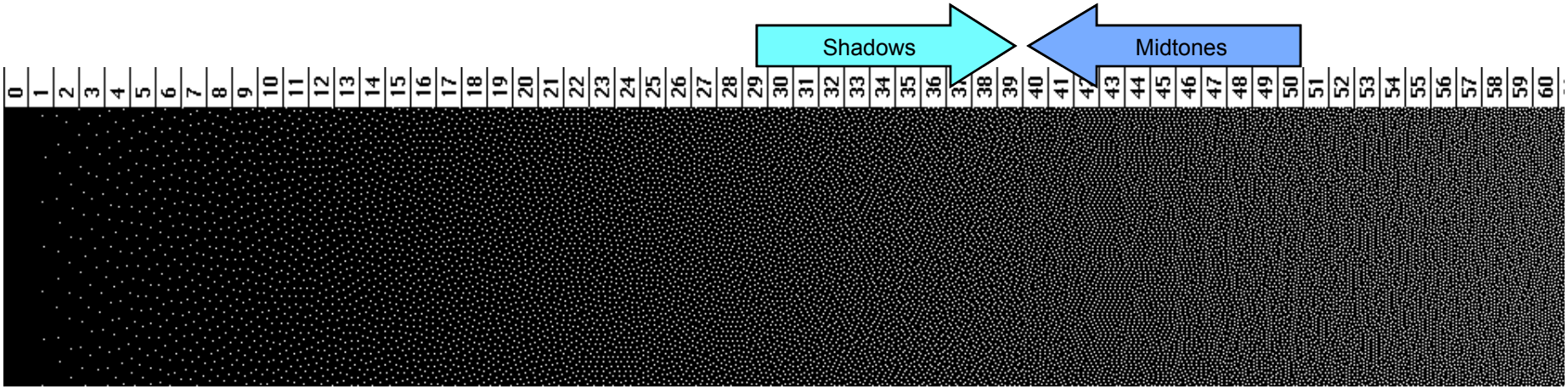
Eschbach









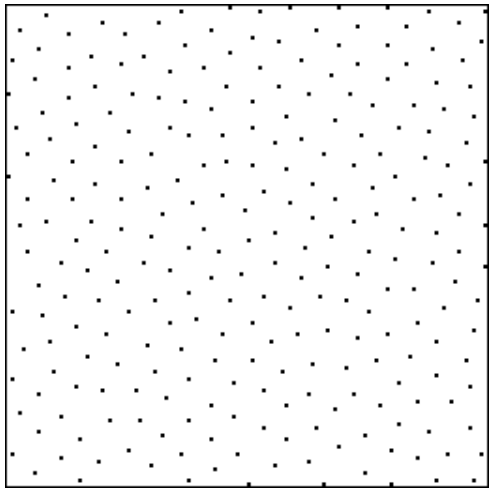


Class II: Neighborhood Processes

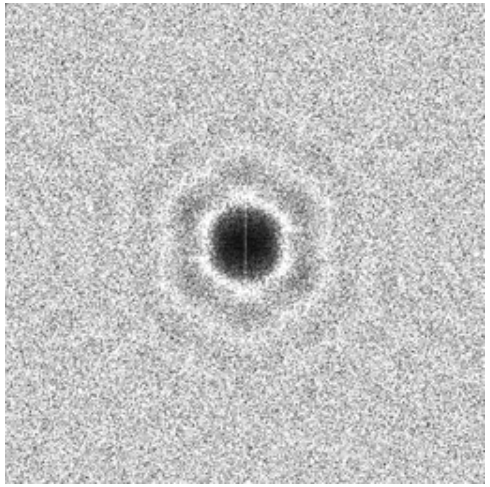
- Conclusion
  - Visual Artifacts can be improved Using Input-Dependent Weights
  - Correlation Between Matrices May Reduce Banding Effect
  - Weak Points of the Method
  - No Objective Criterion on Input-Dependent Weights
  - No Mechanism Proposed To Insure Correlation Between Weights

Objective Criterion of Visual Quality of Halftoning

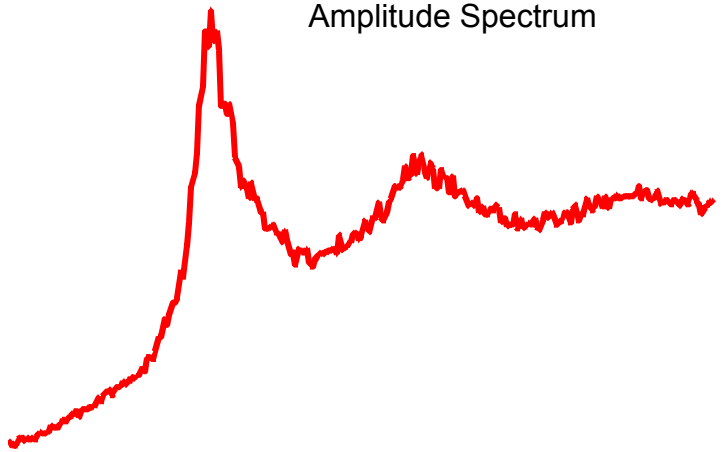
Image



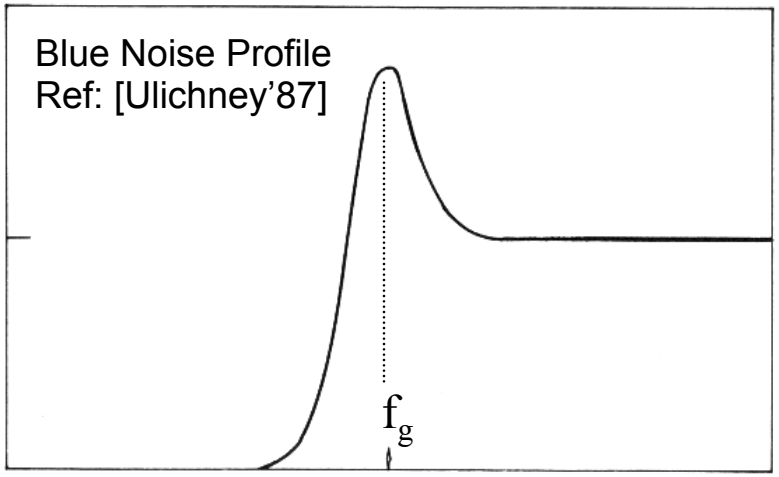
Fourier Amplitude Spectrum



Profile of Fourier Amplitude Spectrum



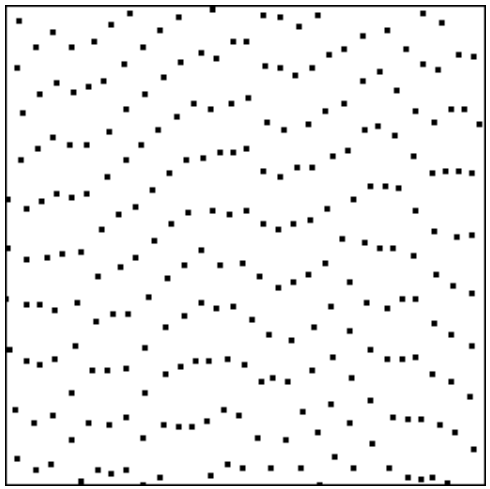
- Radial symmetry
- Characteristic blue noise profile

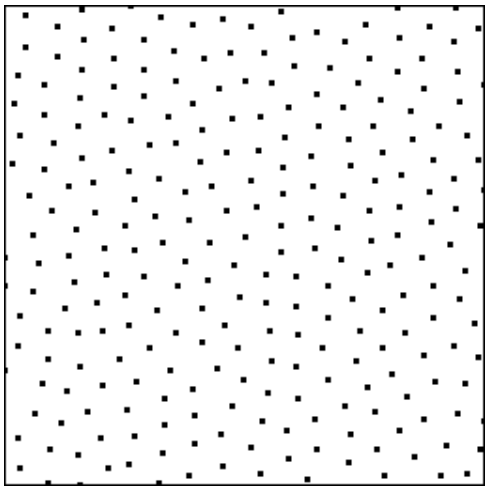


Objective Criterion of Visual Quality of Halftoning

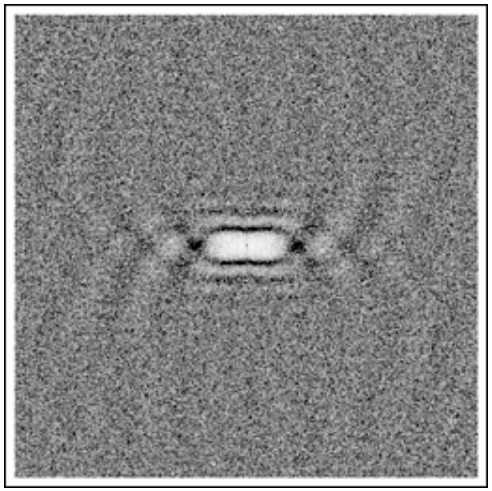
Bad Error-Diffusion
(Std. Floyd-Steinberg, scanlines)

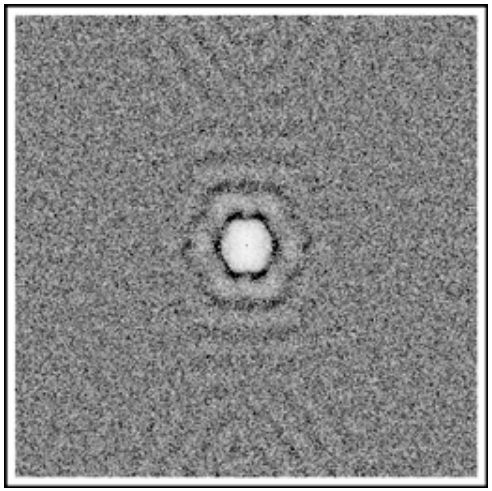
Good Error-Diffusion





Images



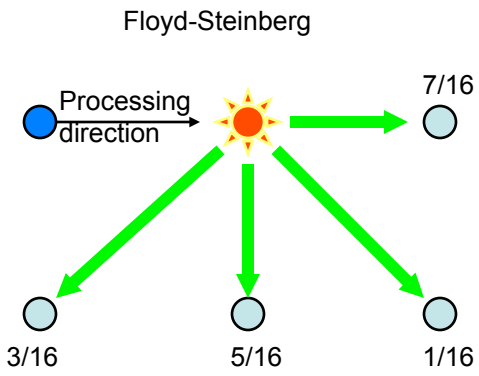


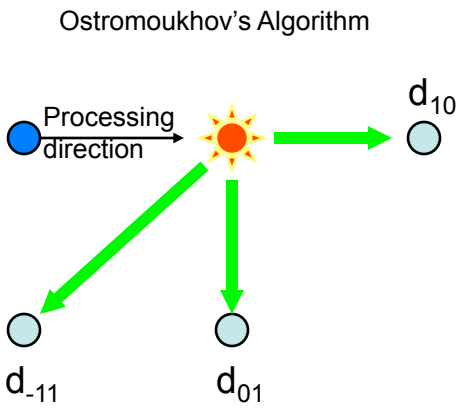
Fourier
Amplitude Spectra

Tone-Dependent Weights

Ostromoukhov 2001

- Simplified set of distribution coefficients $d^i=\{d_{10},d_{-11},d_{01}\}$ **for each intensity level *i***
- For each level, find $d^i=\{d_{10},d_{-11},d_{01}\}$ that approaches "Blue Noise", by minimizing the difference between given Fourier Amplitude Spectrum and ideal "Blue Noise Profile"



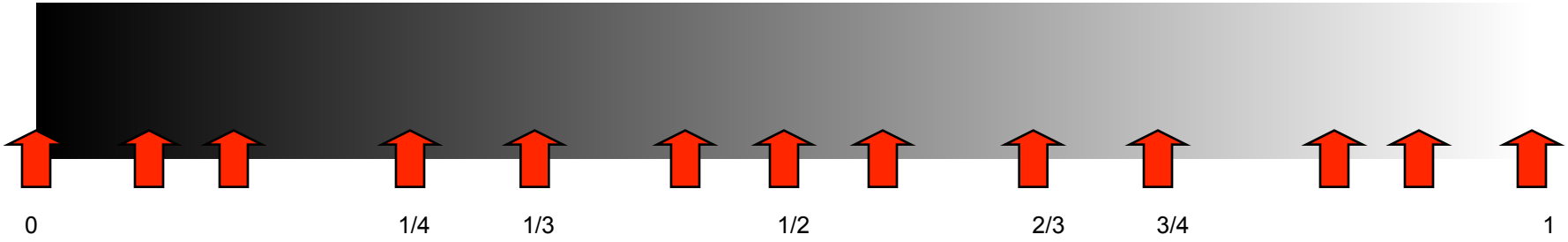


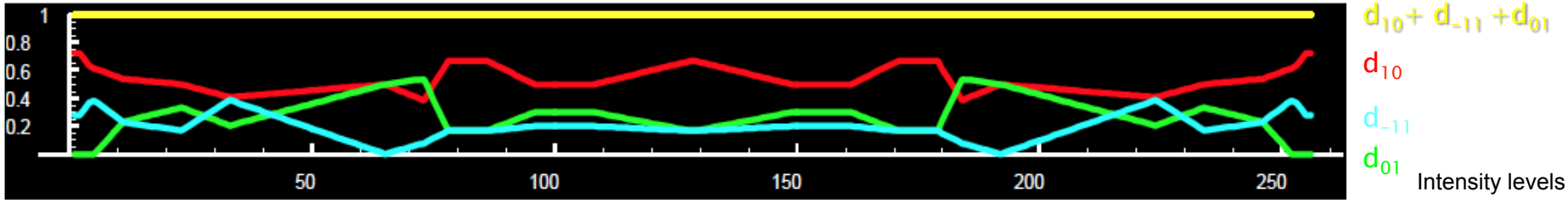
instead of presetting the values, compute the best values

Tone-Dependent Weights

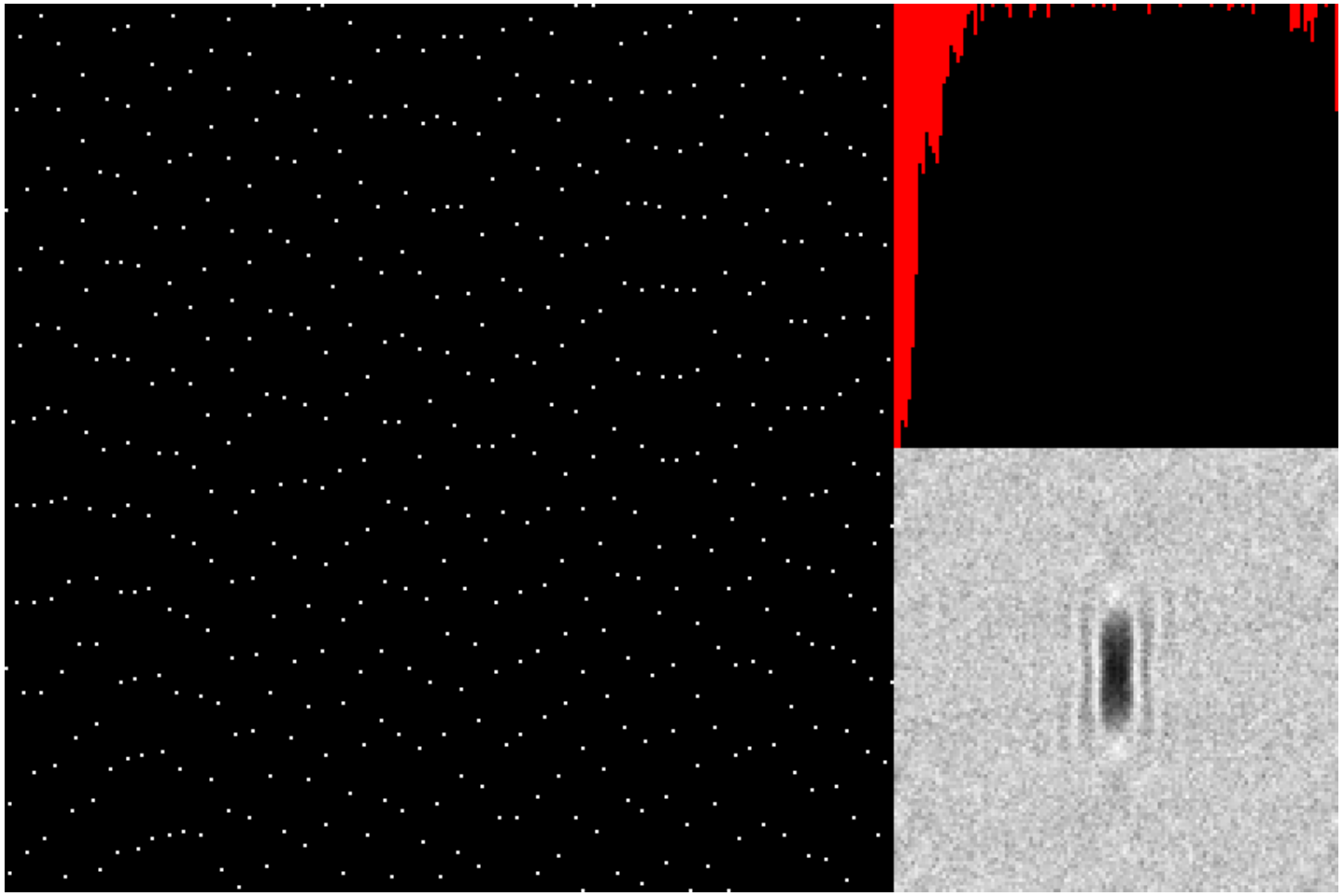
Ostromoukhov 2001

- Define distribution coefficient for a set of key levels
- Interpolate between coefficients





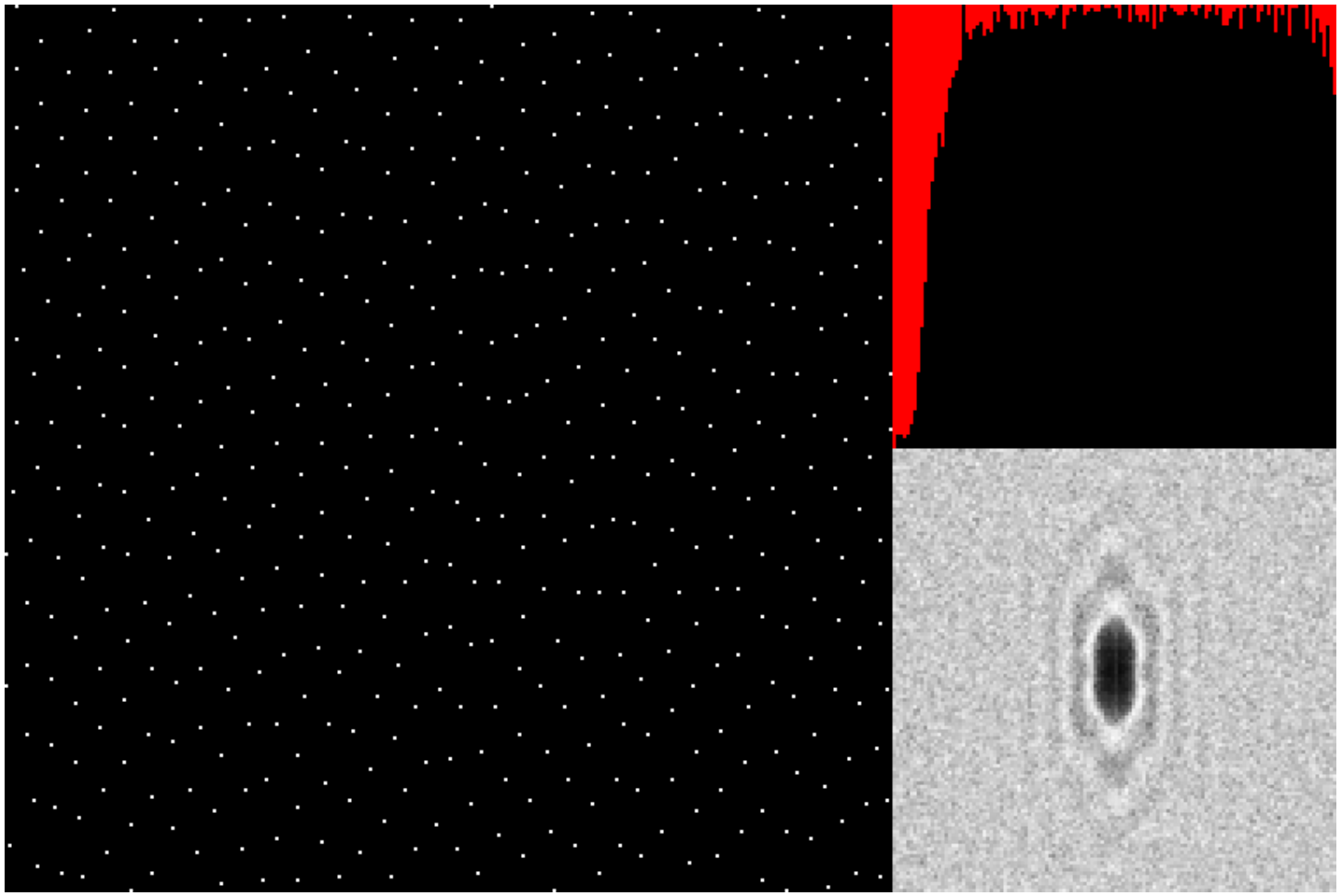
Computing Distribution Coefficients, Iteration # 1



$d_{10} = 0.333333$
$d_{-11} = 0.333333$
$d_{01} = 0.333333$

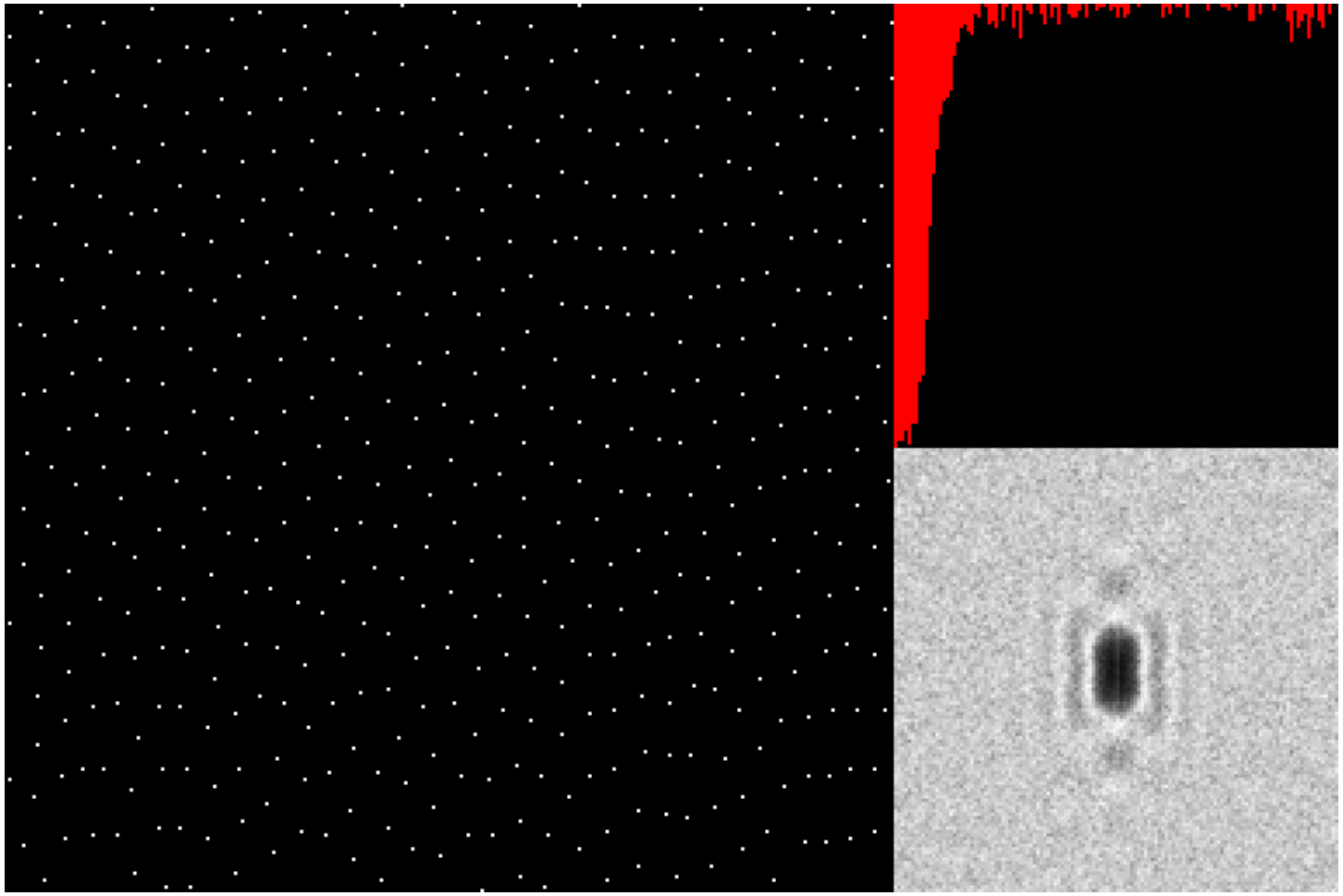
Here it is a particular intensity level (dark)

Computing Distribution Coefficients, Iteration # 3



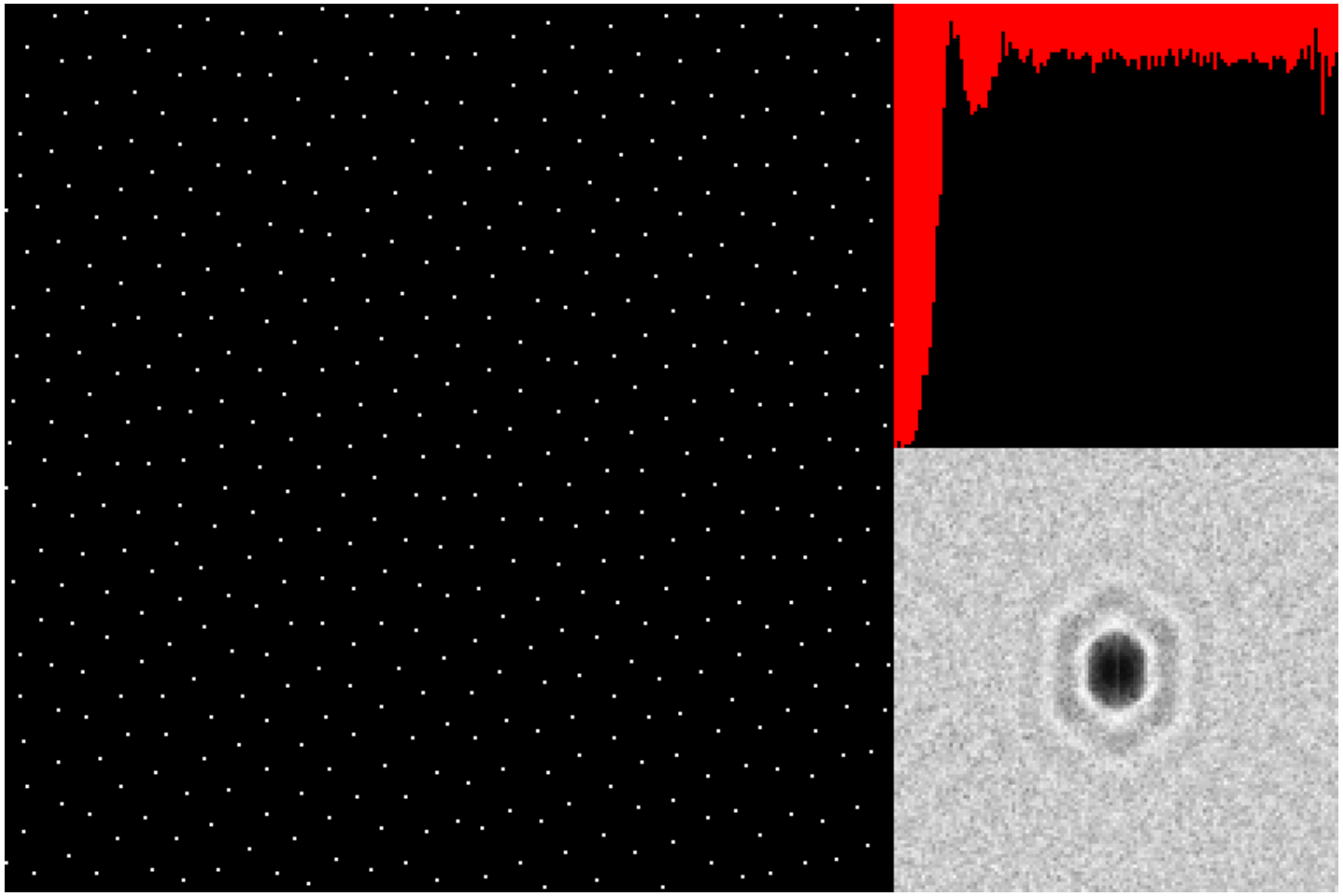
d_{10} = 0.445736
d_{-11} = 0.306202
d_{01} = 0.248062

Computing Distribution Coefficients, Iteration # 7



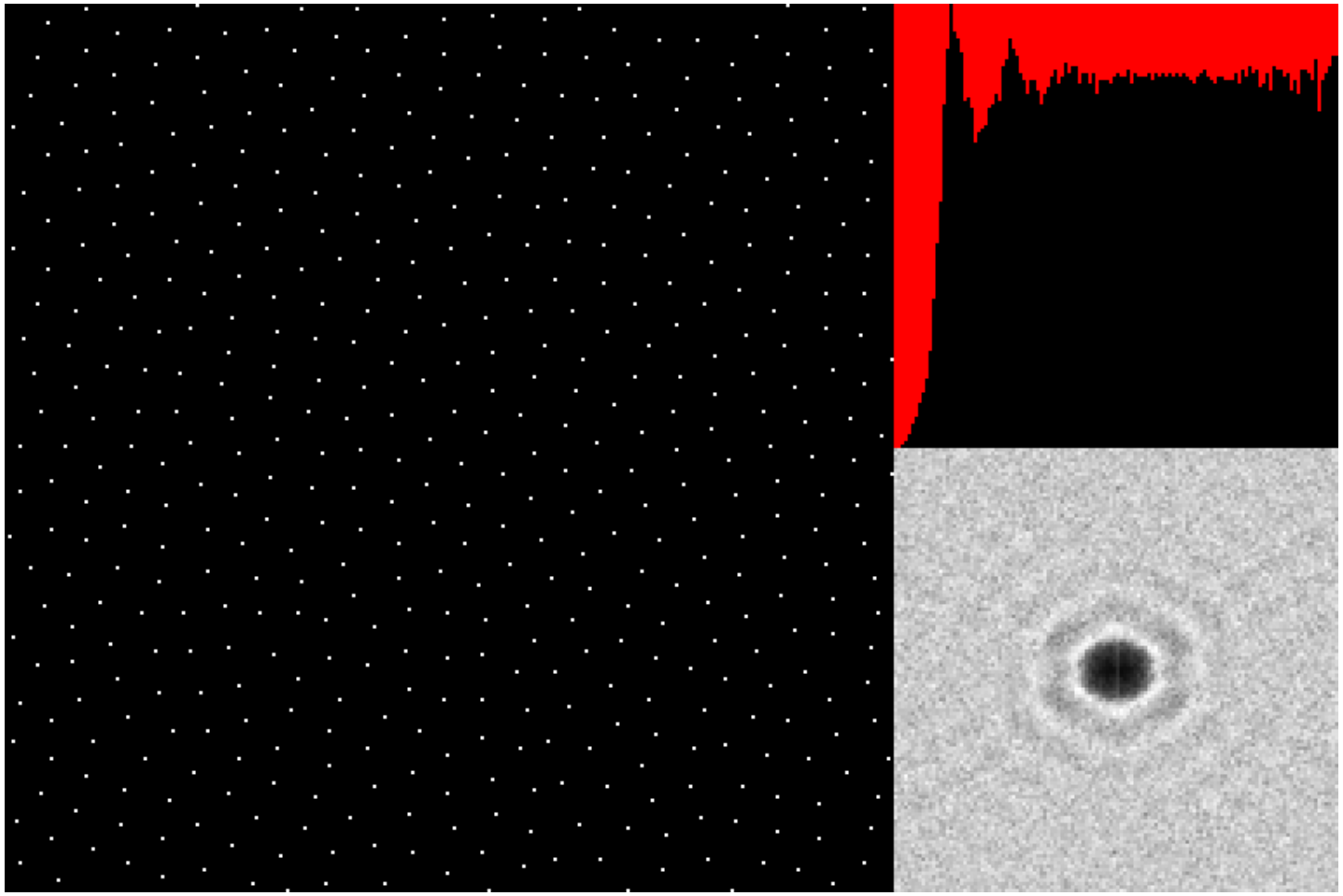
d_{10} = 0.527778
d_{-11} = 0.444445
d_{01} = 0.0277776

Computing Distribution Coefficients, Iteration # 12



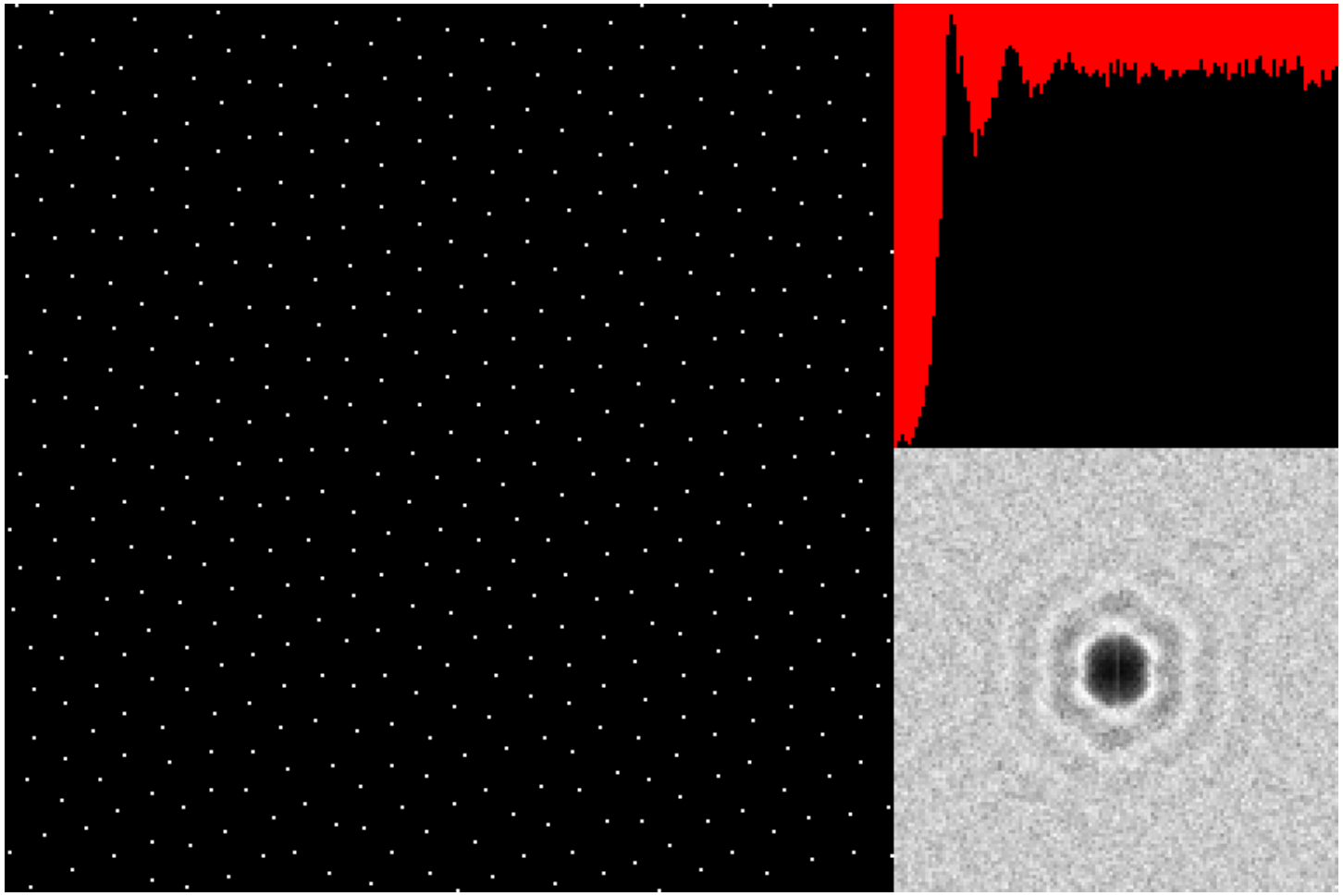
d_{10} = 0.637344
d_{-11} = 0.186967
d_{01} = 0.175689

Computing Distribution Coefficients, Iteration # 15



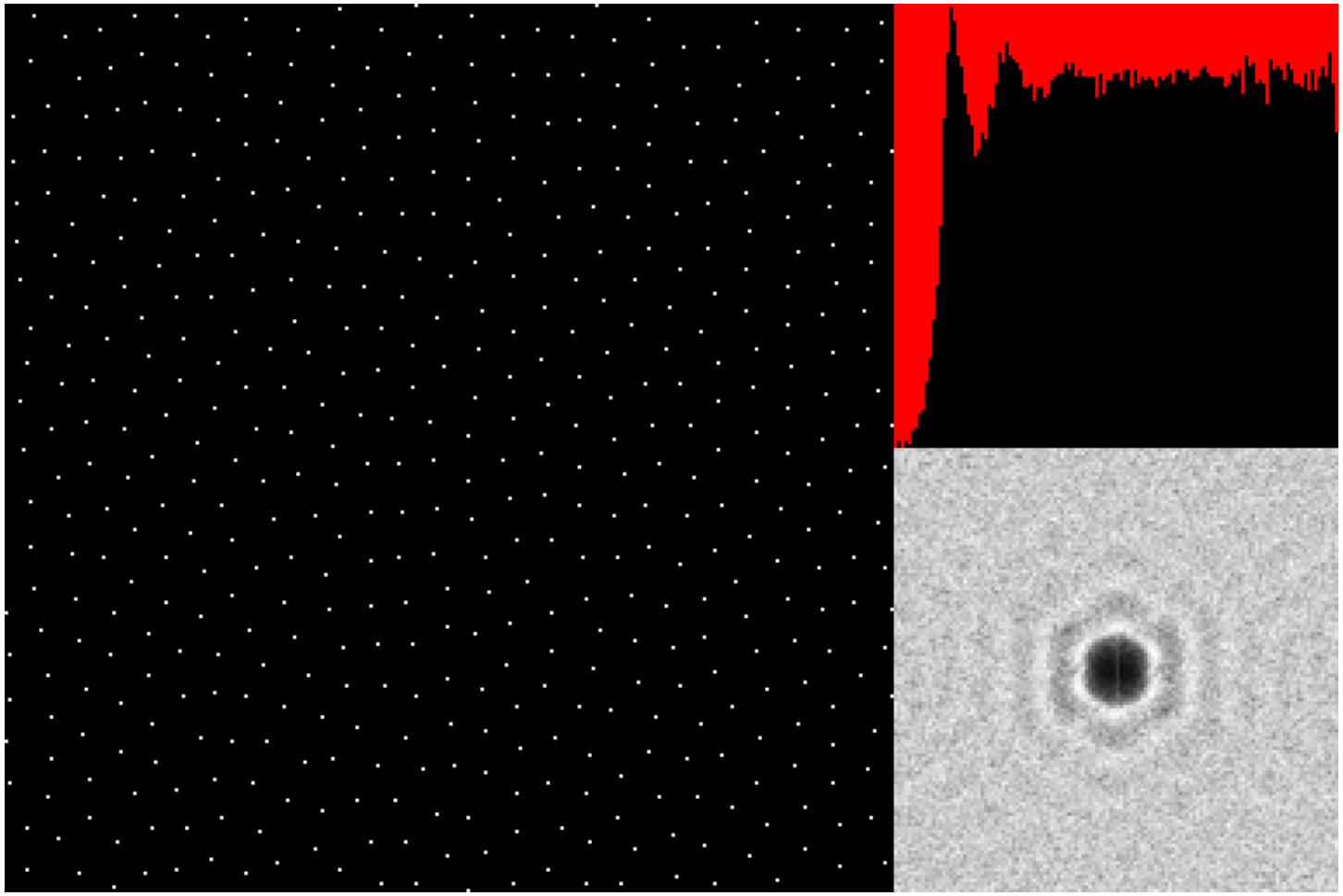
d_{10} = 0.630038
d_{-11} = 0.218744
d_{01} = 0.151219

Computing Distribution Coefficients, Iteration # 20



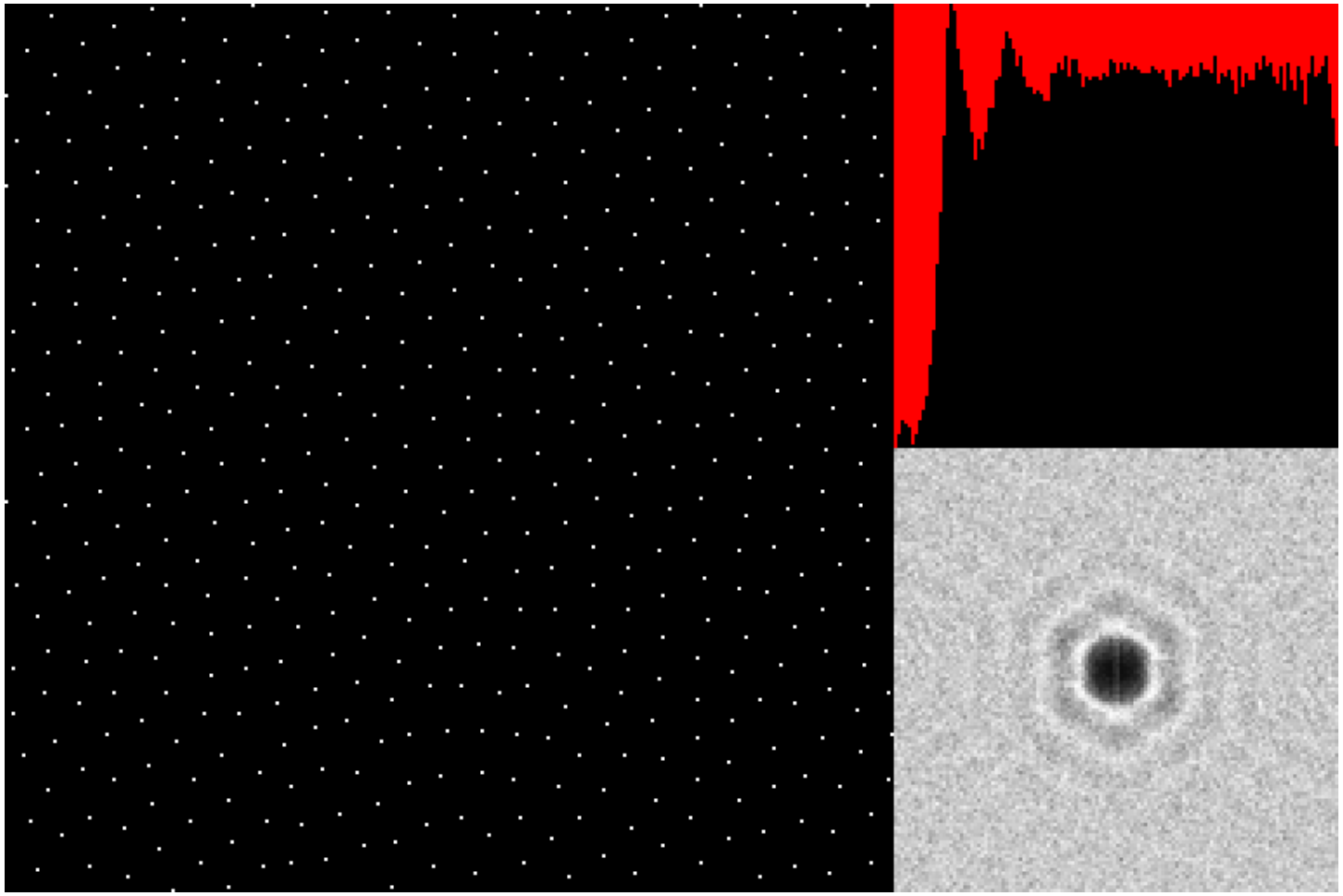
d_{10} = 0.673123
d_{-11} = 0.114484
d_{01} = 0.212393

Computing Distribution Coefficients, Iteration # 33



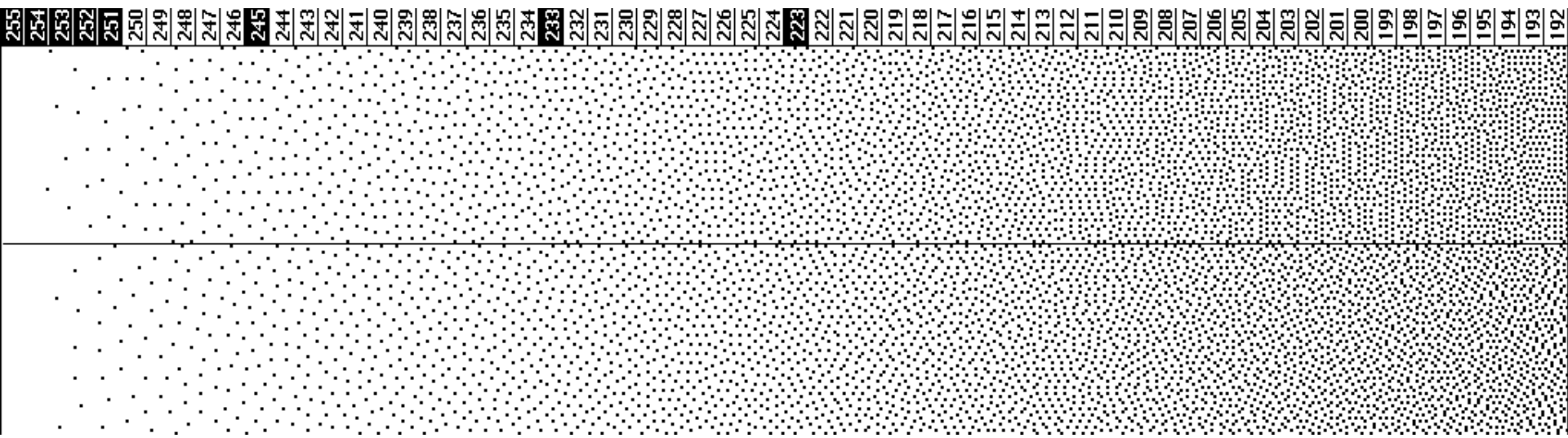
d_{10} = 0.664381
d_{-11} = 0.185621
d_{01} = 0.149998

Computing Distribution Coefficients, Iteration # 41



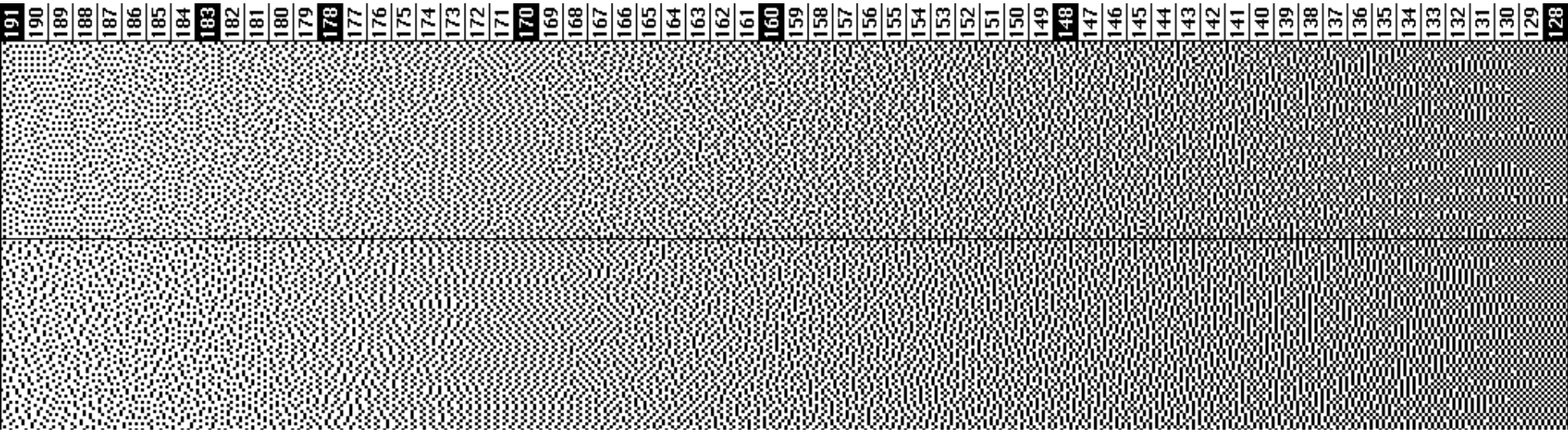
d_{10} = 0.655903
d_{-11} = 0.182528
d_{01} = 0.16157

Results



Top: Std. Floyd-Steinberg E-D
Middle: Ostromoukhov 2001

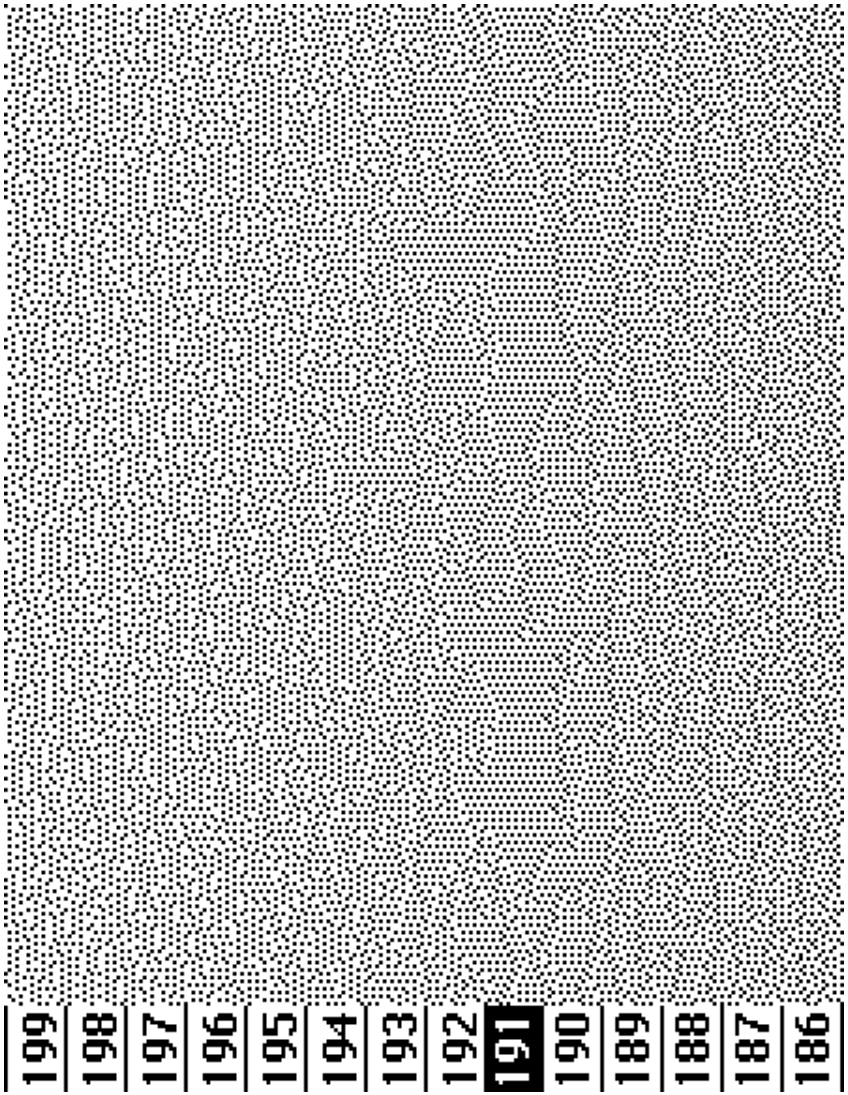
Results



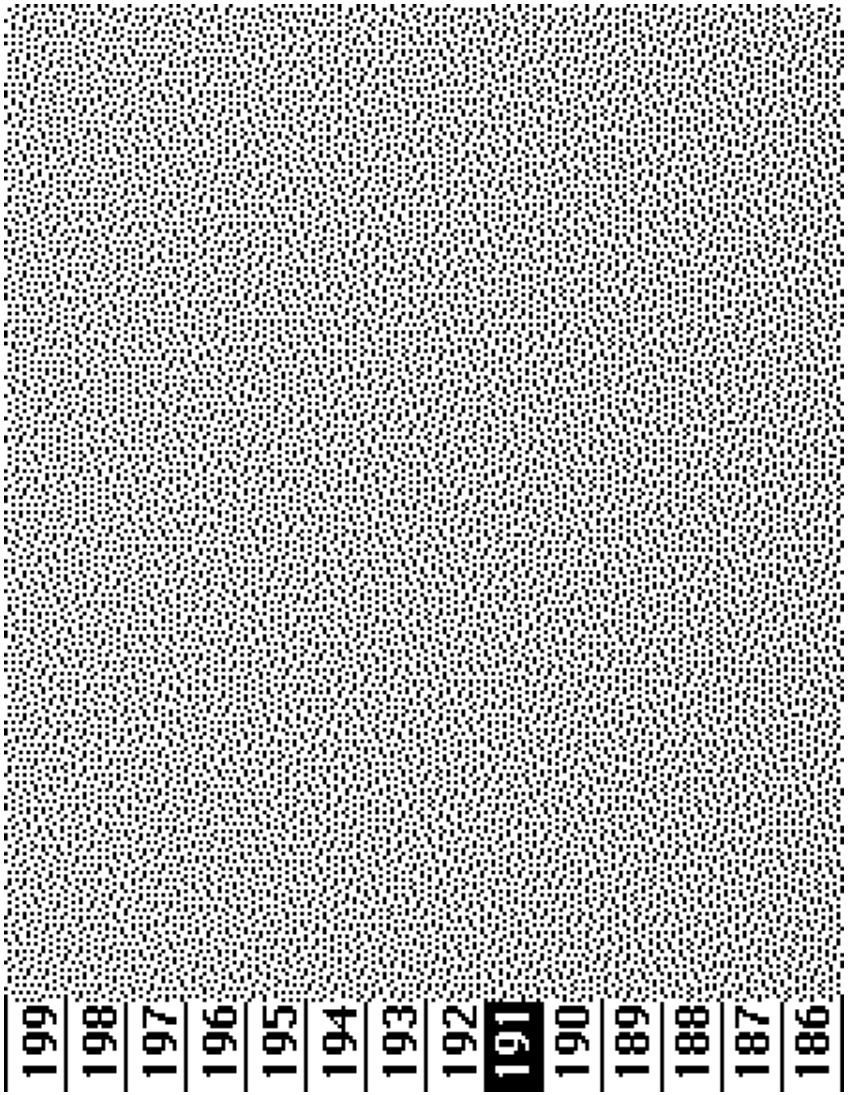
Top: Std. Floyd-Steinberg E-D
Middle: Ostromoukhov 2001

Results

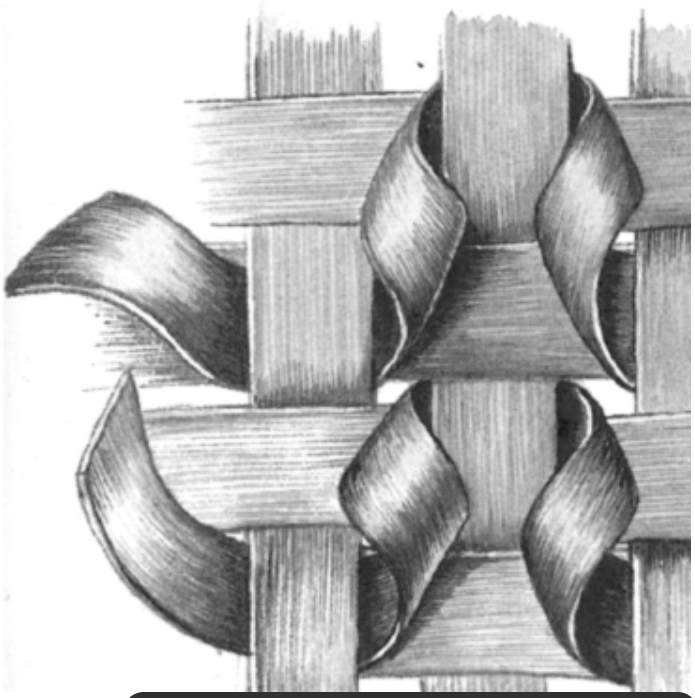
Std. Floyd-Steinberg E-D



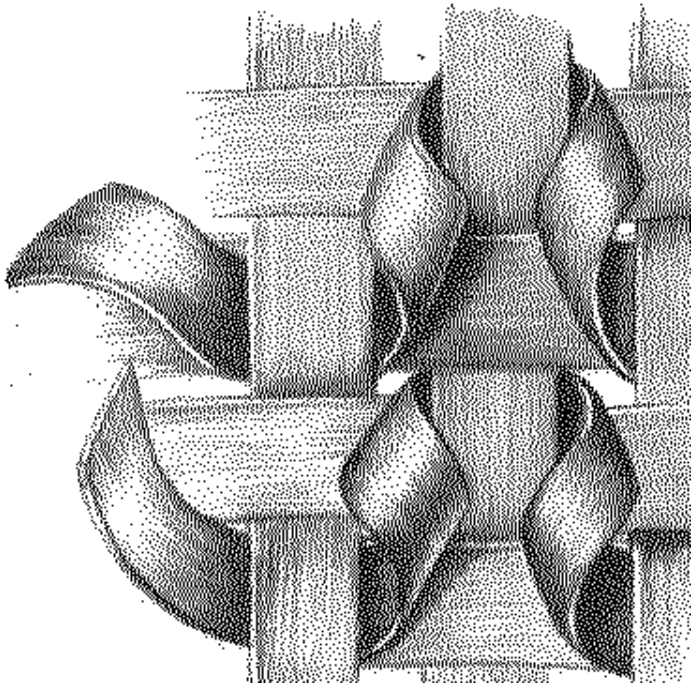
Ostromoukhov 2001



Results



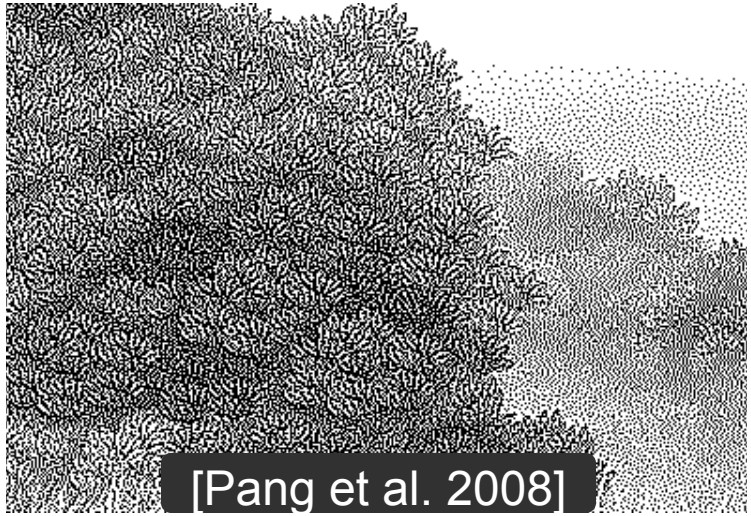
Original



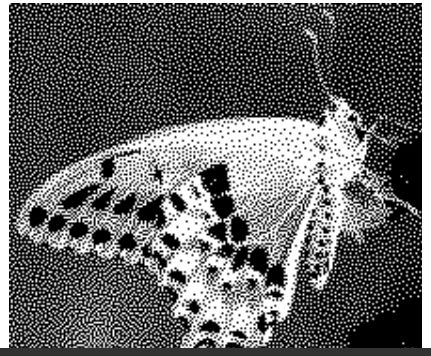
Error Diffusion
[Ostromoukhov 2001]

Class III: Global Optimization

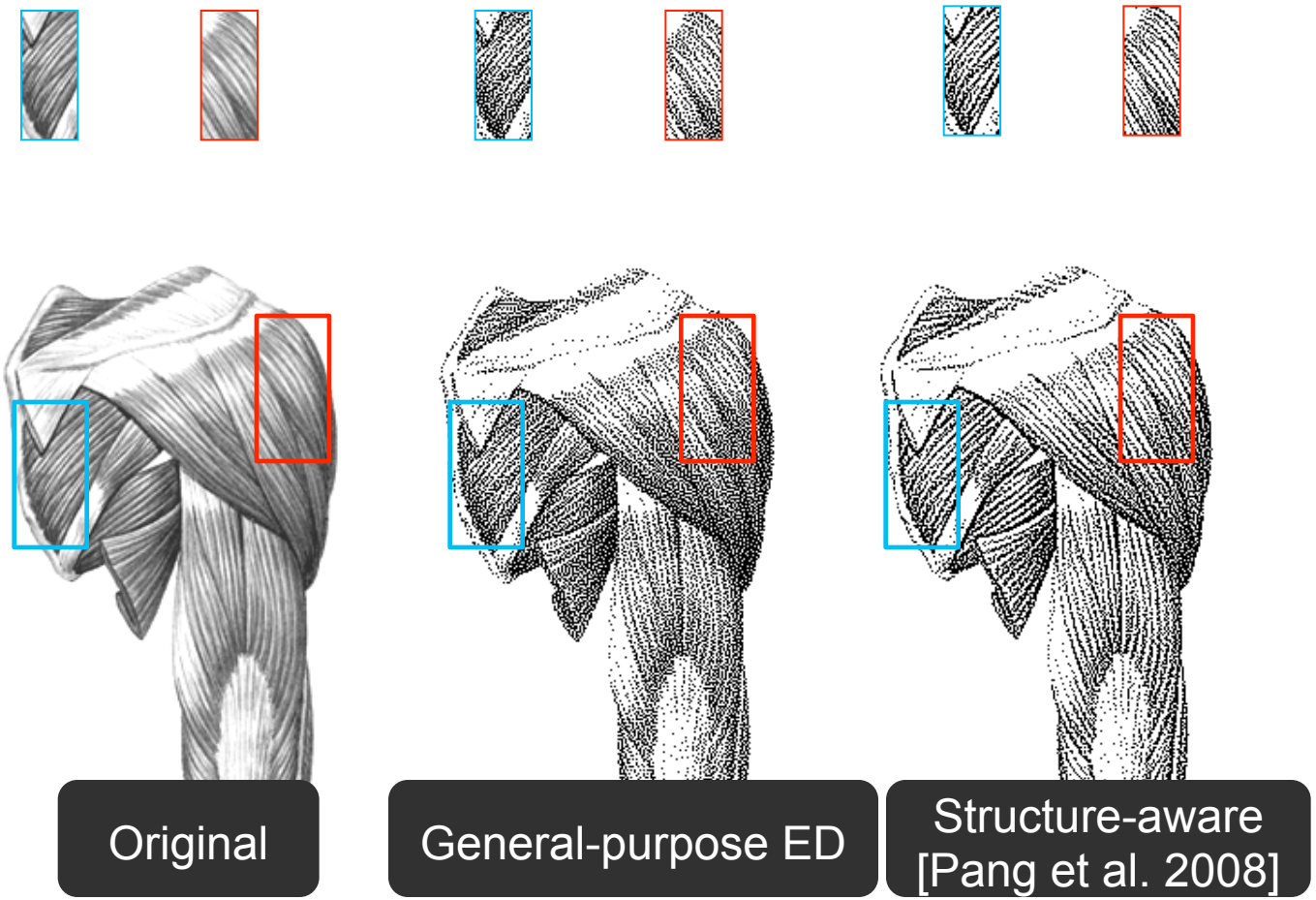
- Optimize whole image
  - Global objective function
  - Keep only good changes
  - Class III ⇨ **prohibitively slow**


[Pang et al. 2008]

- Results
  - No visible artifacts
  - Accurate tones & structures
  - Class III ⇨ **the reference for quality**


[Analoui & Allebach 1992]

Results



http://www.cse.cuhk.edu.hk/~ttwong/papers/structurehalftone/structurehalftone.html

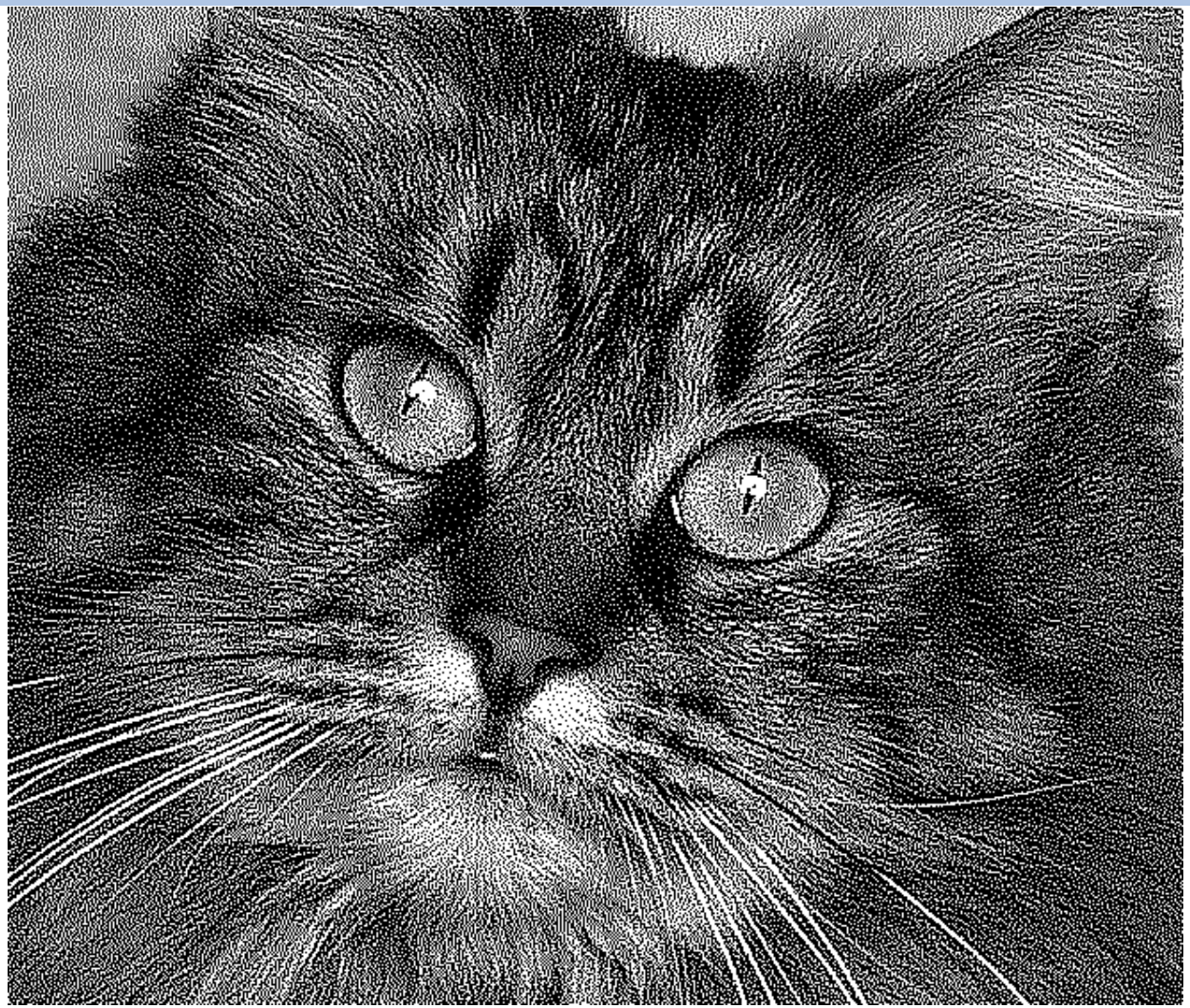
Results

Original

Results

General-purpose ED

Results



Structure-aware
[Pang et al. 2008]

Results


Structure-Aware Halftoning
Wai-Man Pang[1], Yingge Qu[1], Tien-Tsin Wong[1],
Daniel Cohen-Or[2], Pheng-Ann Heng[1]
[1]The Chinese University of Hong Kong
[2]Tel Aviv University

That’s All for Today!

More info:
- A Guide to Halftone Technologies
http://www.isisimaging.com/Isis_Library_files/Halftone_technology_guide.pdf