Dithering / digital halftoning

AIG 5/8/20

Kevin Ly

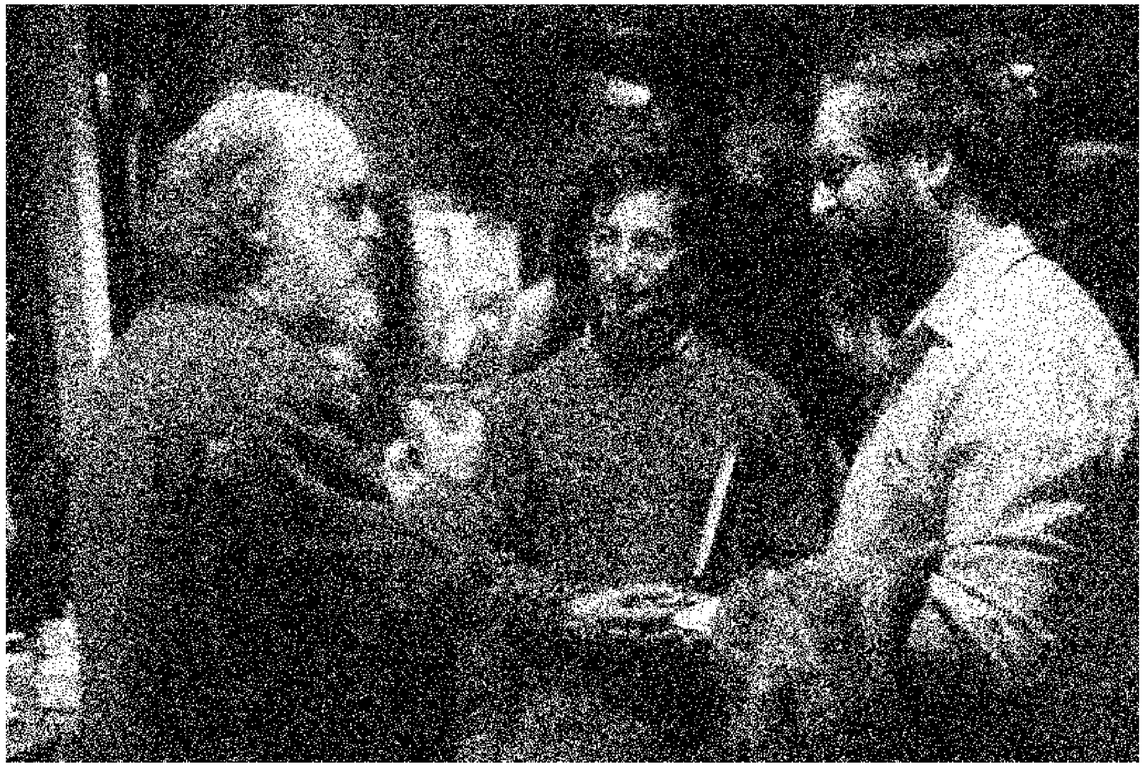
Itinerary

- Random dither
- Ordered dither
- Error-diffusion halftoning

all taken from DHALF.TXT

Random dither demo

Fixed threshold



Random dither

What just happened?

- In fixed thresholding, the “error” is concentrated: if a region is neither very dark nor very bright, rounding destroys it
- In random dithering, the “error” is spread out

By “error” I mean (BW image) - (grayscale image, normalized)

“The basic idea is to replace a sharp boundary between quantization levels by a fuzzy one, where the two levels are so intimately mixed together that the eye is fooled into seeing intermediate levels and perceives the transition as smooth rather than abrupt. (The everyday meaning of the word ‘dither’ — to act nervously or indecisively — is applied metaphorically to the wavering between two levels that appears to take place.)”

J. Gomes et al., *Image Processing for Computer Graphics*

Patterning and ordered dither

- Patterning: take a grayscale pixel and replace it with an *array* of 0s and 1s. The higher the value of the grayscale pixel, the more 1s, e.g.

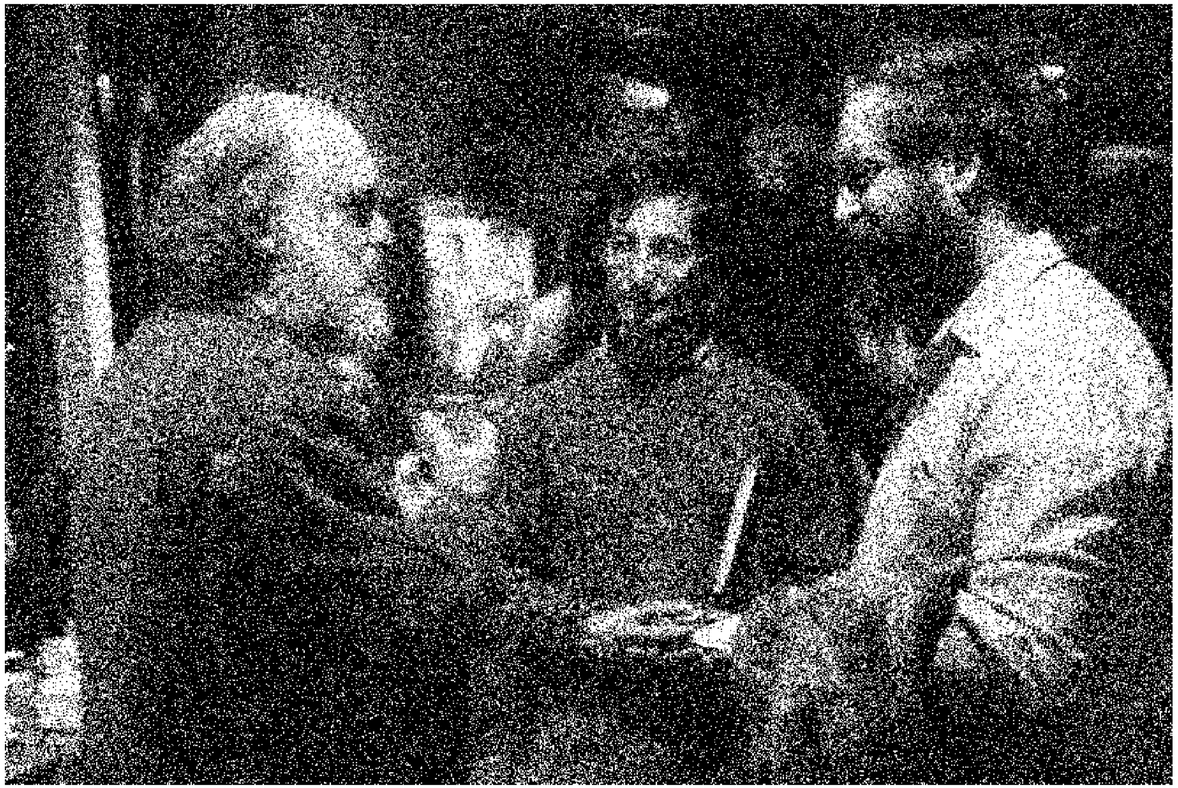
$$69 \to \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{matrix} \quad \text{or} \quad 213 \to \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

- Ordered dither: take a $N \times N$ cluster of gray scale pixels, rescale it, compare it against a (fixed) $N \times N$ pattern, and round according to this pattern. An "optimal" pattern for $N = 4$ is

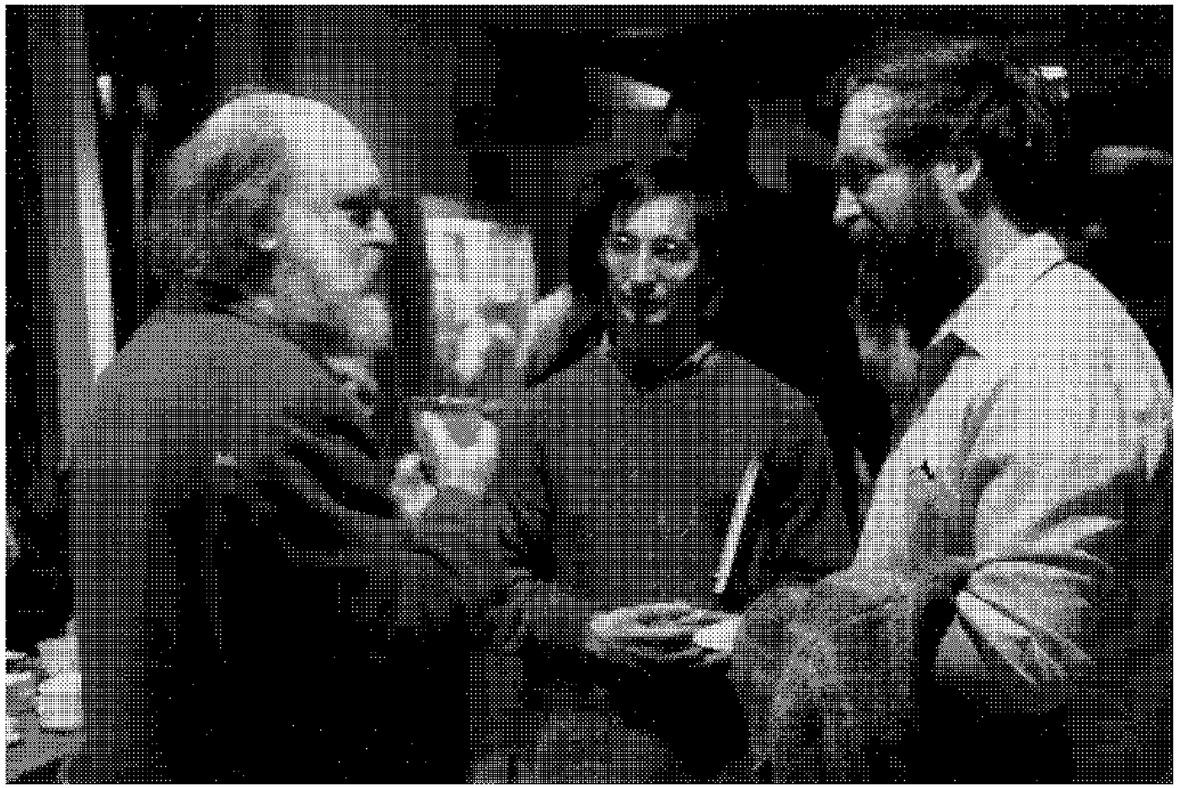
$$\begin{matrix} 1 & 9 & 3 & 11 \\ 13 & 5 & 15 & 7 \\ 4 & 12 & 2 & 10 \\ 16 & 8 & 14 & 6 \end{matrix}$$

(see Bayer (1973) or Ulichney (1987), or "Method of Recursive Tessellation")

Ordered dither demo



Random dither



Ordered dither

Error diffusion

- Go through each pixel in grayscale image and round it up to white or down to black, depending on which one is closer
- BUT: after rounding, compute the “error” and distribute it to some neighboring pixels, e.g.

$$\begin{matrix} 0.3 & 0.34 \\ 0.25 & 0.29 \end{matrix} \rightarrow \begin{matrix} 0 & 0.20875 \\ 0.15625 & 0.27125 \end{matrix}$$

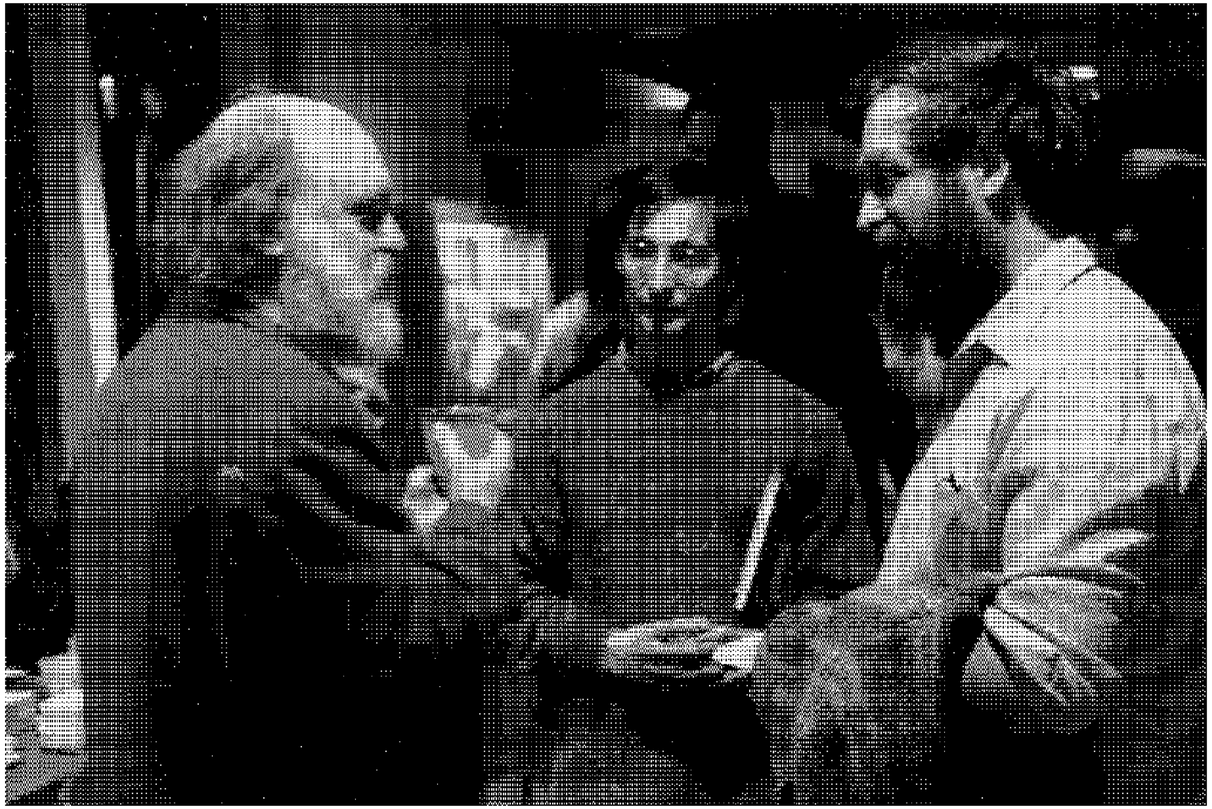
(done with Floyd-Steinberg filter)

$$\left(\frac{1}{16}\right) \begin{matrix} & * & 7 \\ 3 & 5 & 1 \end{matrix}$$

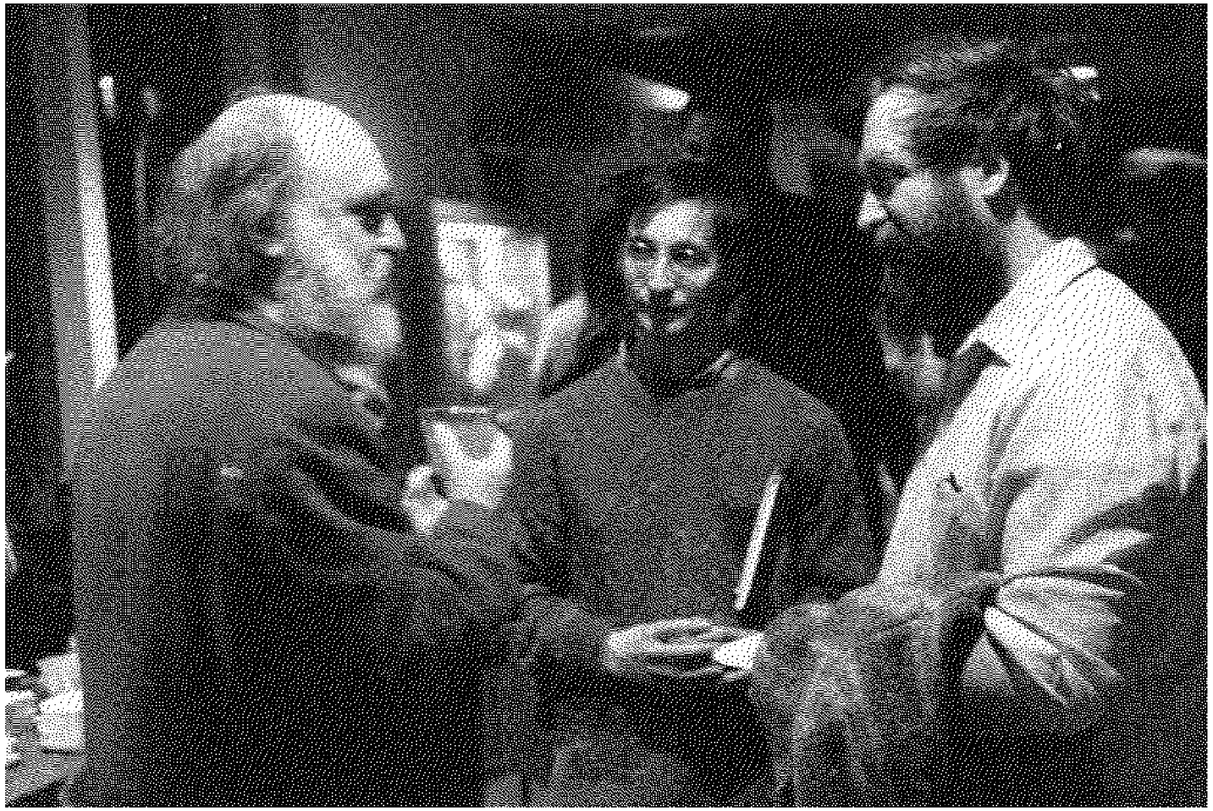
“The algorithm scans the image from left to right, top to bottom, quantizing pixel values one by one. Each time the quantization error is transferred to the neighboring pixels, while not affecting the pixels that have already been quantized. Hence, if a number of pixels have been rounded downwards, it becomes more likely that the next pixel is rounded upwards, such that on average, the quantization error is close to zero.”

Wikipedia

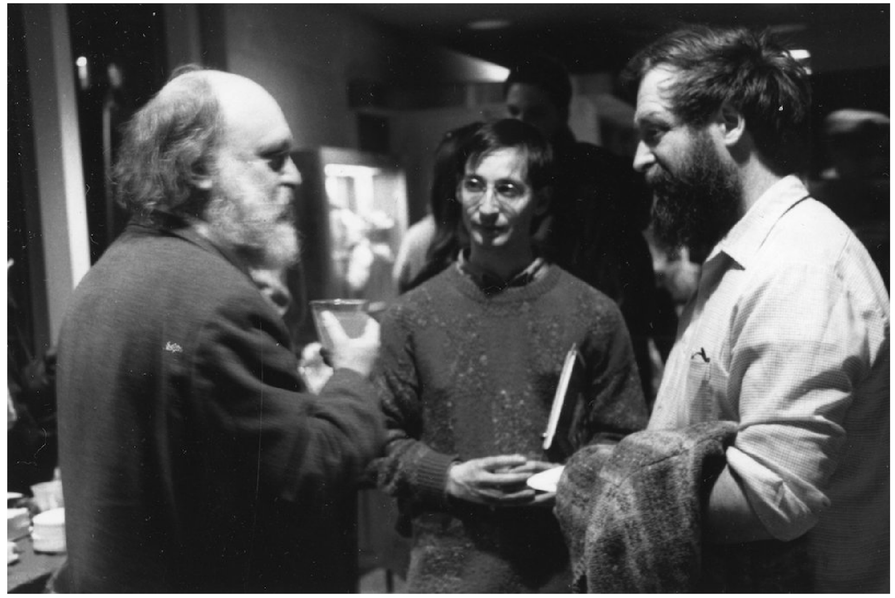
Floyd-Steinberg filter demo

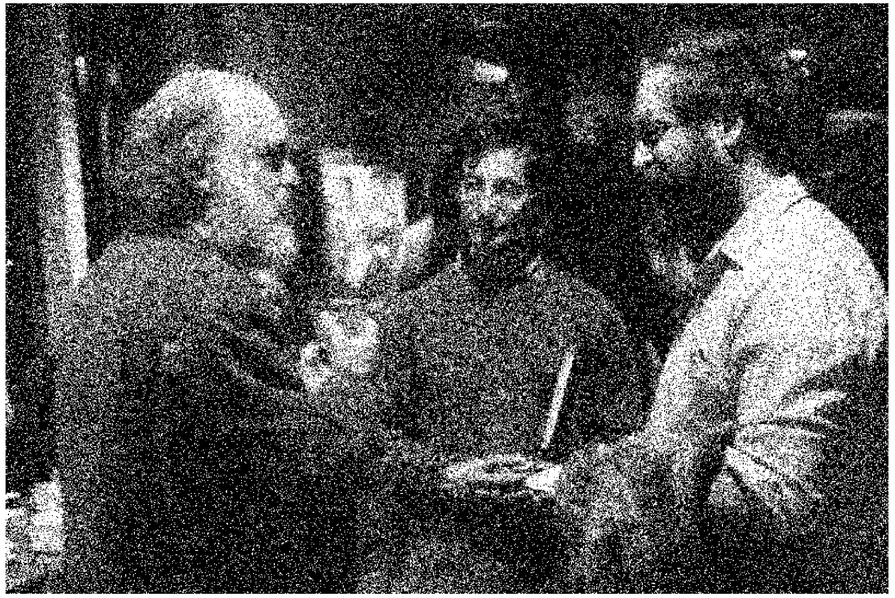


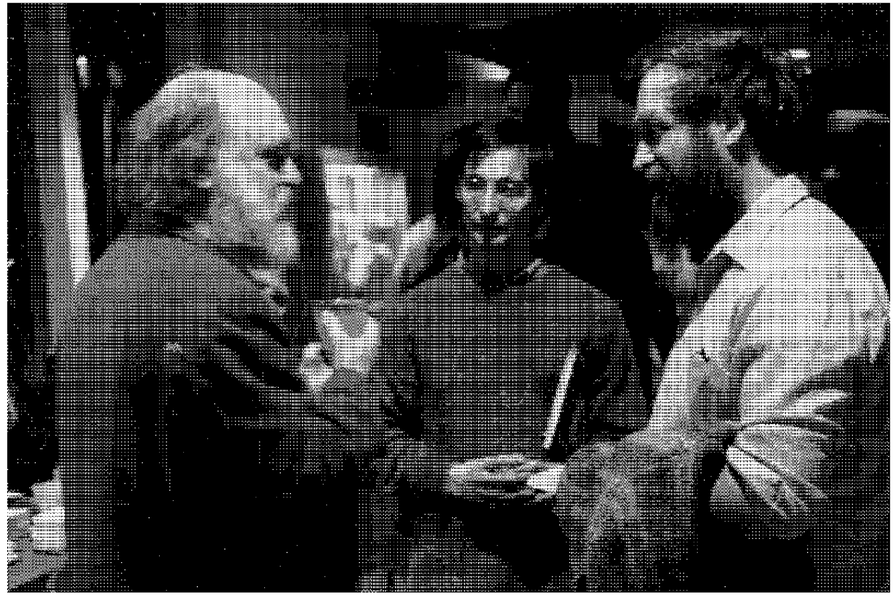
Ordered dither

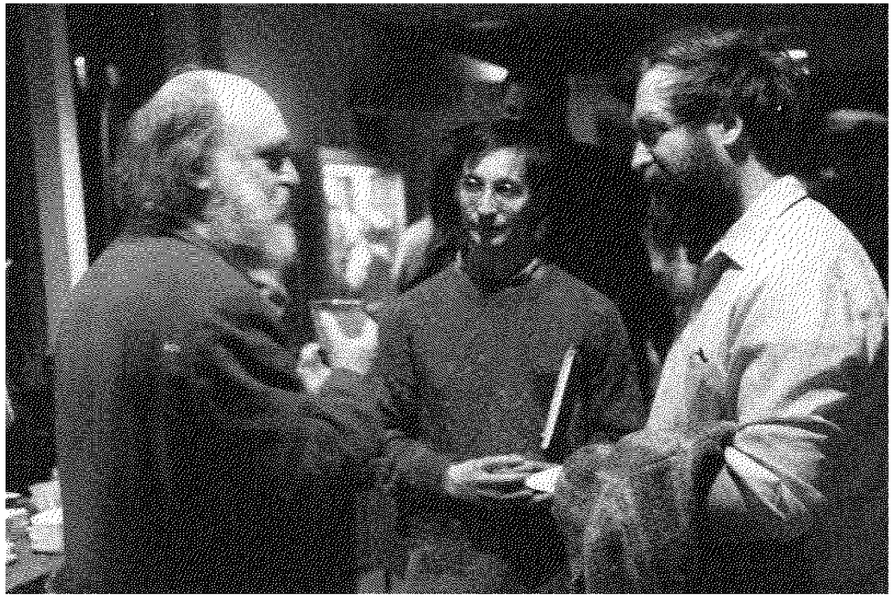


Floyd-Steinberg filter









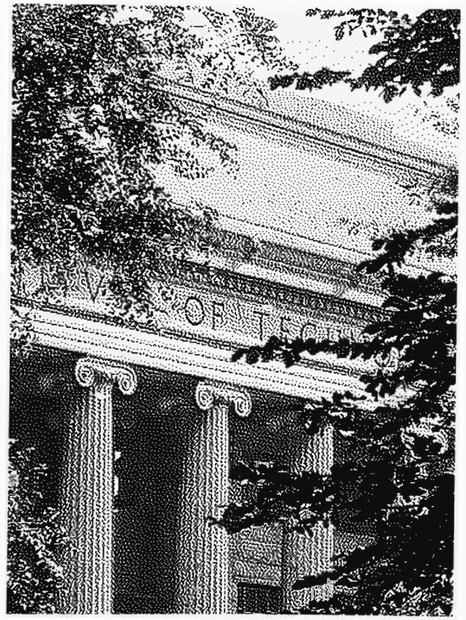
Other filters

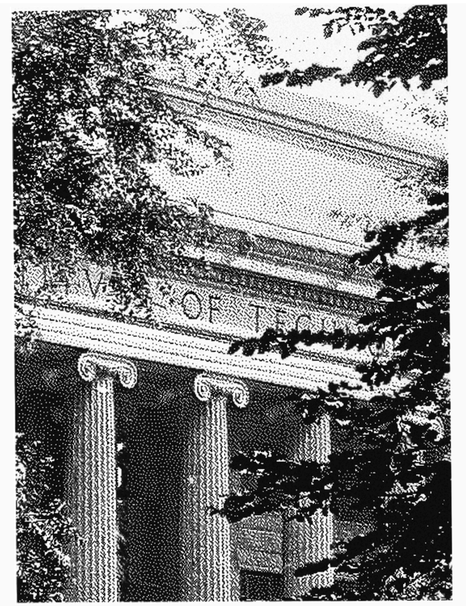
- Jarvis, Judice, and Ninke (1976)

$$\left(\frac{1}{48}\right)\begin{array}{ccccc} & & * & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{array}$$

- Stucki (1981)

$$\left(\frac{1}{42}\right)\begin{array}{ccccc} & & * & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{array}$$





Concluding remarks

- These were dated examples for converting to a two level system (black and white), but there are dithering techniques for more sophisticated problems, e.g. in audio processing
- If you ever find yourself in a situation where you have some data that you need to “quantize,” try shaking it up a bit before going with simple rounding / truncating!

References

- Lee Daniel Crocker, Paul Boulay, and Mike Morra, DHALF.TXT (1991)
- Robert Ulichney, *Digital Halftoning* (1987)
- B. E. Bayer, An optimum method for two-level rendition of continuous-tone pictures (1973)
- J. F. Jarvis, C. N. Judice, and W. H. Ninke, A survey of techniques for the display of continuous tone pictures on bi-level displays (1976)
- P. Stucki, MECCA — a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction (1981)