INTERPOLATED HALFTONING, REHALFTONING, AND HALFTONE COMPRESSION

Prof. Brian L. Evans
bevans@ece.utexas.edu
http://www.ece.utexas.edu/~bevans

Collaboration with Dr. Thomas D. Kite and
Mr. Niranjan Damera-Venkata

Laboratory for Image and Video Engineering
The University of Texas at Austin
http://anchovy.ece.utexas.edu/

OUTLINE

- Introduction to halftoning
- Halftoning by error diffusion
  - Linear gain model
  - Modified error diffusion
- Interpolated halftoning
- Rehalftoning
- JBIG2 halftone compression
- Conclusions

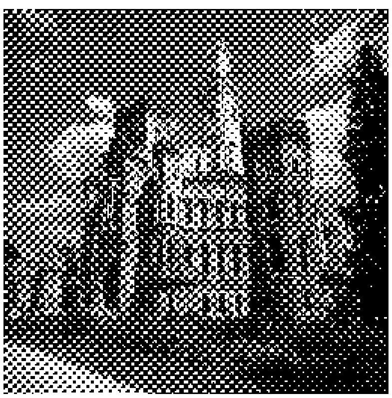
INTRODUCTION: HALFTONING

- Was analog, now digital processing
- Wordlength reduction for images
  - 8-bit to 1-bit for grayscale
  - 24-bit RGB to 8-bit for color displays
  - 24-bit RGB to CMYK for color printers
- Applications
  - Printers
  - Digital copiers
  - Liquid crystal displays
  - Video cards
- Halftoning methods
  - Screening
  - Error diffusion
  - Direct binary search
  - Hybrid schemes

EXAMPLE HALFTONES

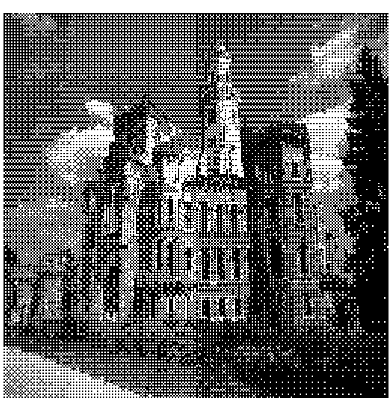
Original image

Direct binary search



Clustered dot screen

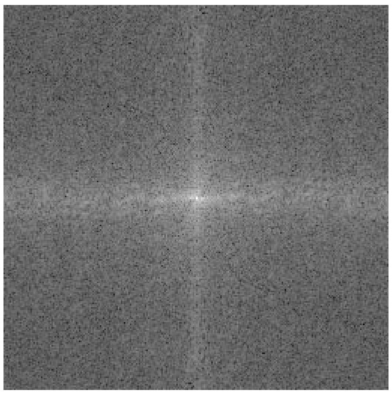
Floyd Steinberg



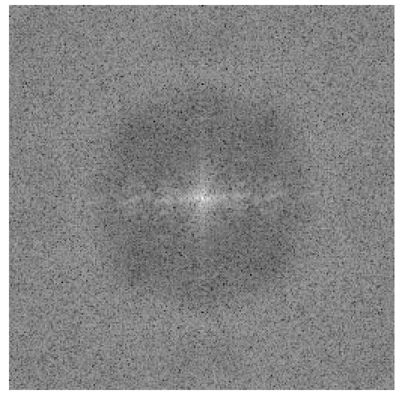
Dispersed dot screen

Modified Diffusion

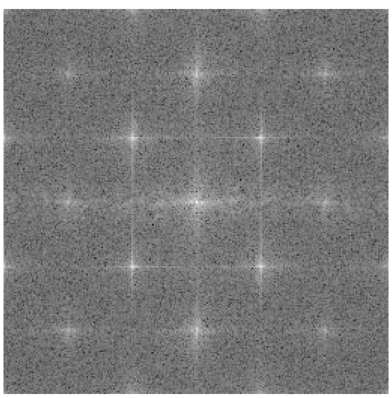
FOURIER TRANSFORMS



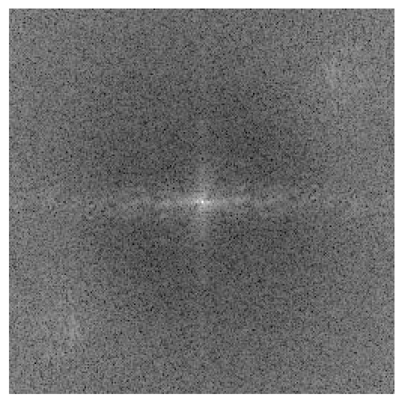
Original image



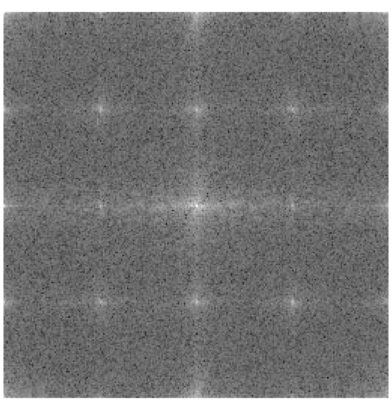
Direct binary search



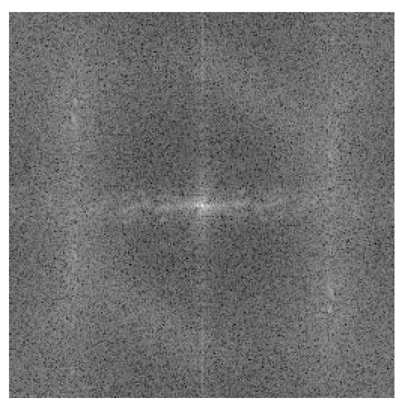
Clustered dot screen



Floyd Steinberg



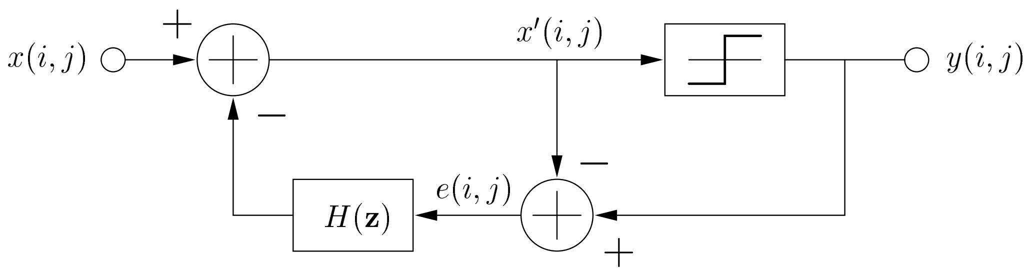
Dispersed dot screen



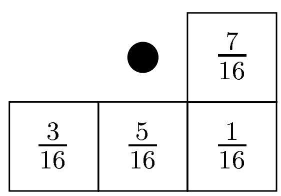
Modified Diffusion

ERROR DIFFUSION

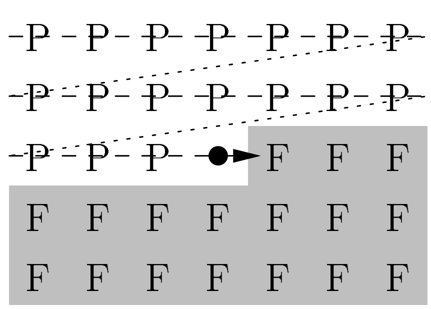
- 2-D delta-sigma modulator
- Noise shaping feedback coder



- Error filter



- Raster scan order



P = Past
F = Future

- Serpentine scan also used

ERROR DIFFUSION (cont.)

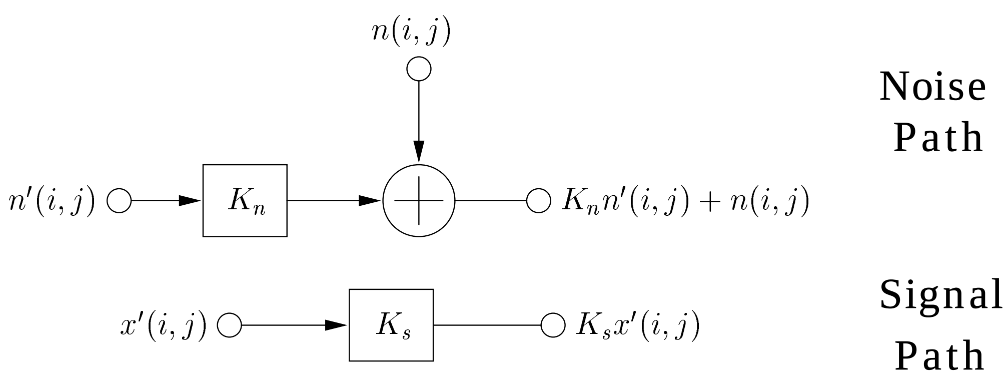
- Quantizer

$$y(i,j) = \begin{cases} 0, & x'(i,j) < 0.5 \\ 1, & x'(i,j) \geq 0.5 \end{cases}$$

- Governing equations

$$\begin{aligned} e(i,j) &= y(i,j) - x'(i,j) \\ x'(i,j) &= x(i,j) - h(i,j) * e(i,j) \end{aligned}$$

- Non-linearity difficult to analyze
- Linearize quantizer
  [Kite, Evans, Bovik & Sculley 1997]

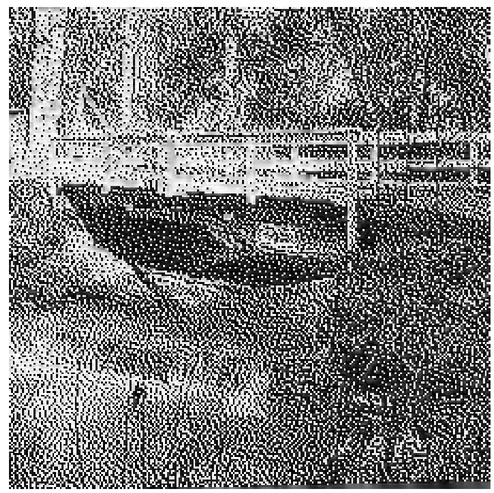


- Separate signal and noise paths
  [Ardalan & Paulos 1987]

LINEAR GAIN MODEL

- Quantization error correlated with input [Knox 1992]

Floyd-Steinberg



Jarvis, Judice & Ninke

- Least squares fit of quantizer input to output defines signal gain

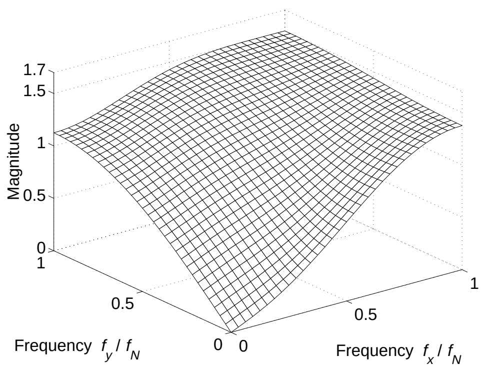
$$K_s = \frac{E[|x'(i,j)|]}{2E[x'(i,j)^2]}$$

- Signal gain: $K_s \approx \text{constant}$
- Noise gain: $K_n = 1$

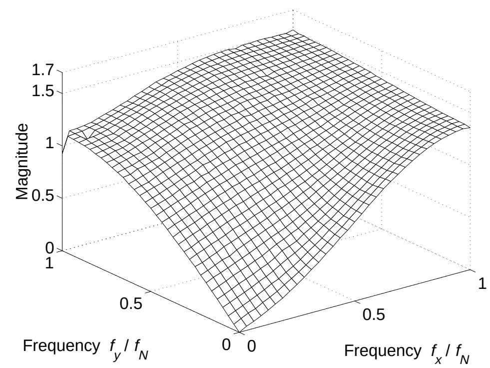
GAIN MODEL PREDICTIONS

- Noise transfer function (NTF)

$$\mathrm{NTF} = 1 - H(\mathbf{z})$$



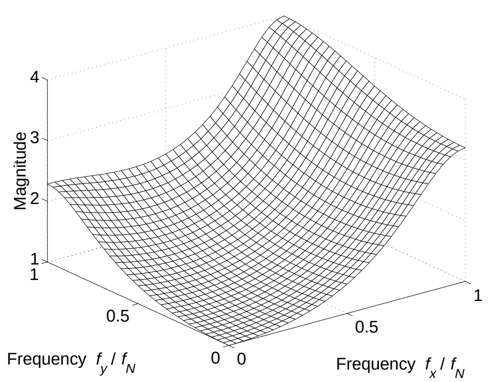
Predicted



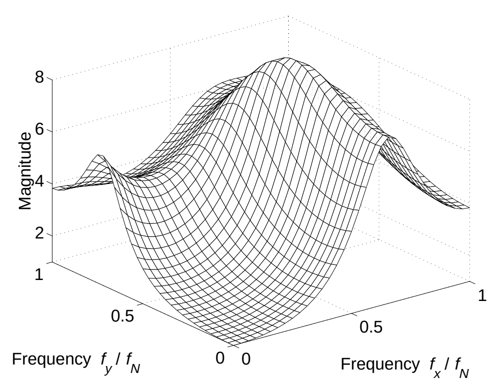
Measured

- Signal transfer function (STF)

$$\mathrm{STF} = \frac{K_s}{1 + (K_s - 1)H(\mathbf{z})}$$



Floyd-Steinberg



Jarvis *et al.*

Predicting Signal Gain K_s

- Predict K_s from error filter as:

$$K_s = 1.17R - 0.2$$

$$R = \frac{\int_{-\pi}^{\pi} \left| X(\omega_1, \omega_2) \right|^2 d\omega_1 d\omega_2}{\int_{-\pi}^{\pi} \left| X(\omega_1, \omega_2) \right|^2 \left| H(\omega_1, \omega_2) \right|^2 d\omega_1 d\omega_2}$$

$X(\omega_1,\omega_2)$ = Fourier Transform of input to error filter

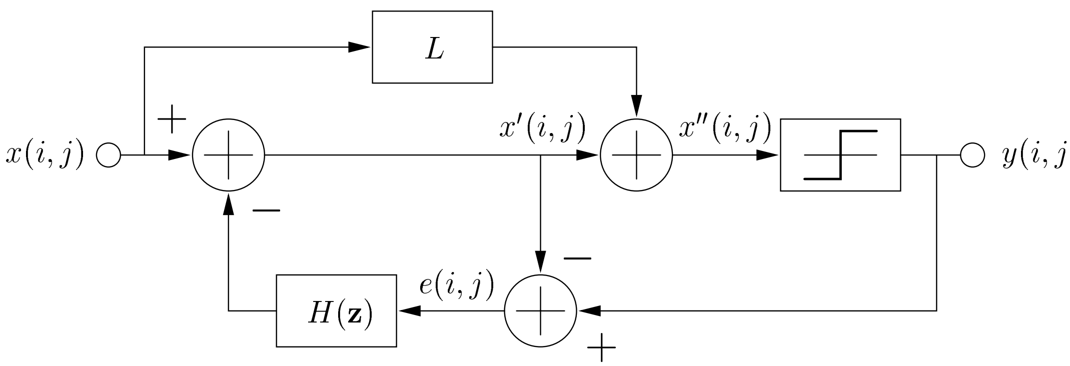
$H(\omega_1,\omega_2)$ = Fourier Transform of error filter

But $X(\omega_1,\omega_2)$ is the error spectrum, so

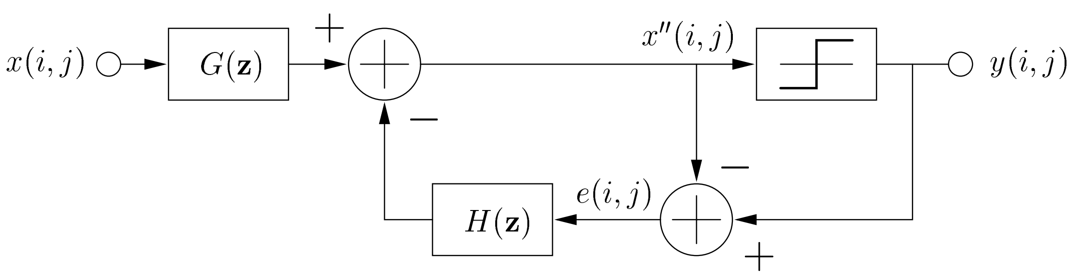
$$X(\omega_1, \omega_2) = 1 - H(\omega_1, \omega_2)$$

MODIFIED ERROR DIFFUSION

- Efficient method of adjusting sharpness [Eschbach & Knox 1991]



- Equivalent circuit: pre-filter



$$G(\mathbf{z}) = 1 + L\,(1 - H(\mathbf{z}))$$

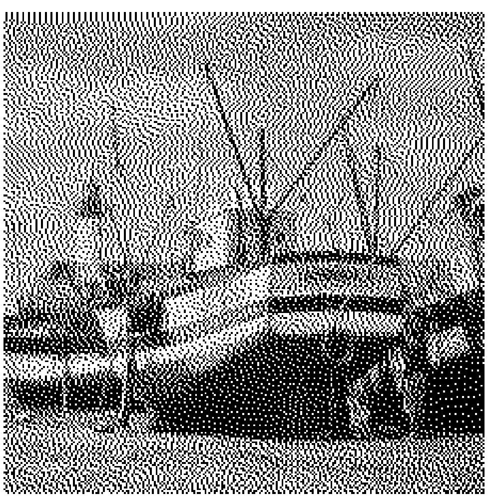
- L can be chosen to compensate for frequency distortion

UNSHARPENED HALFTONES

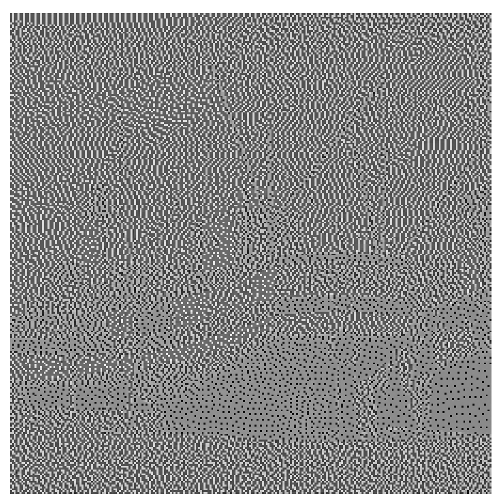
- If $L = \dfrac{1 - K_s}{K_s}$ then $\mathrm{STF} = 1$ (flat)
- Accounts for frequency distortion

Original image

Jarvis halftone



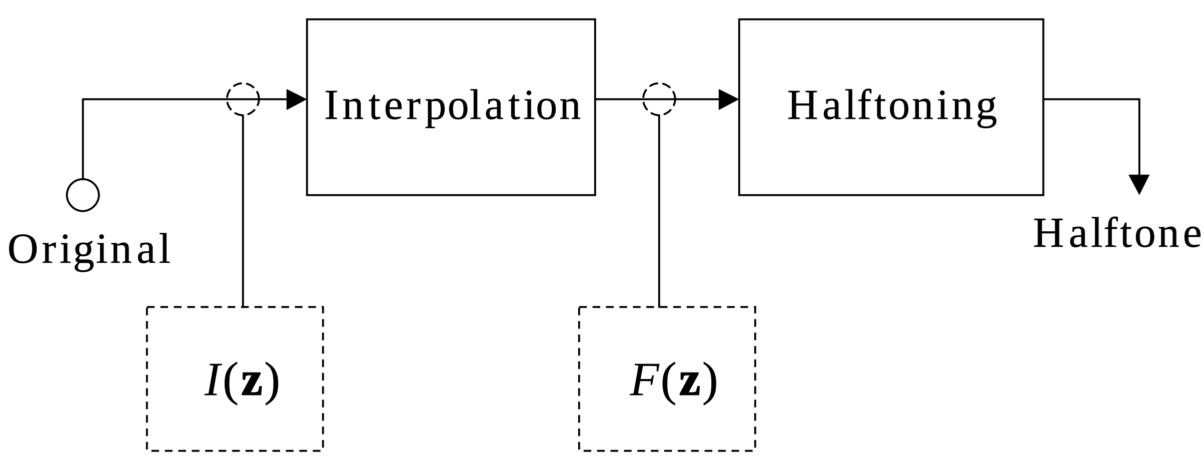
Unsharpened halftone



Residual

INTERPOLATION

- Image resizing
- Different methods (increasing cost)
  - Nearest neighbor (NN)
  - Bilinear (BL)
- Nearest neighbor, bilinear methods
  - Low computational cost
  - Artifacts masked by quantization noise in halftone
  - Correct blurring by using modified error diffusion



INTERPOLATION

- Design L for flat transfer function using linear gain model (L is constant for a given interpolator)
- Compute transfer function of interpolation by M

$$I_{NN}(\mathbf{z}) = \left(\frac{1 - z_x^{-M}}{1 - z_x}\right)\left(\frac{1 - z_x^{-M}}{1 - z_y}\right)$$

$$I_{BL}(\mathbf{z}) = \left(\frac{1 - z_x^{-M}}{1 - z_x}\right)^2\left(\frac{1 - z_x^{-M}}{1 - z_y}\right)^2$$

- Compute signal transfer function

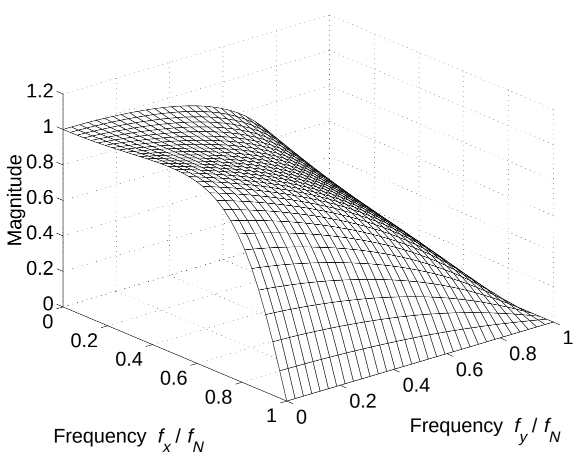
$$F(\mathbf{z}) = \frac{K_s(1 + L(1 - H(\mathbf{z})))}{1 + (K_s - 1)H(\mathbf{z})}$$

- Compute L to flatten the end-to-end transfer function of the system

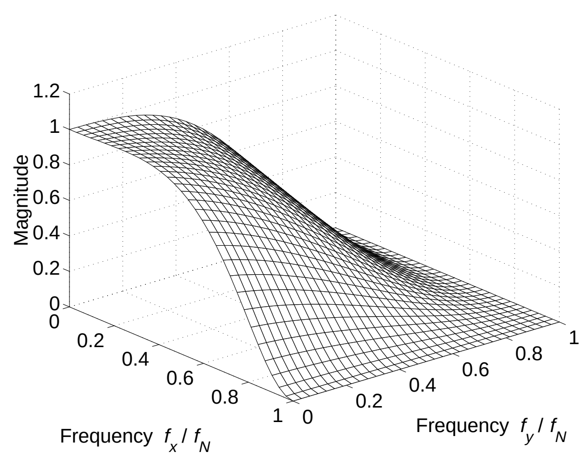
INTERPOLATION RESULTS

Nearest neighbor ×2

Bilinear ×2



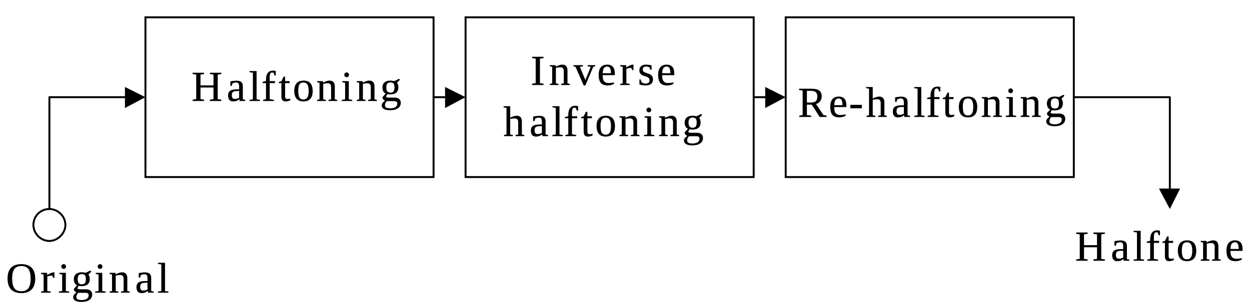
Transfer function
L = –0.0105



Transfer function
L = 0.340

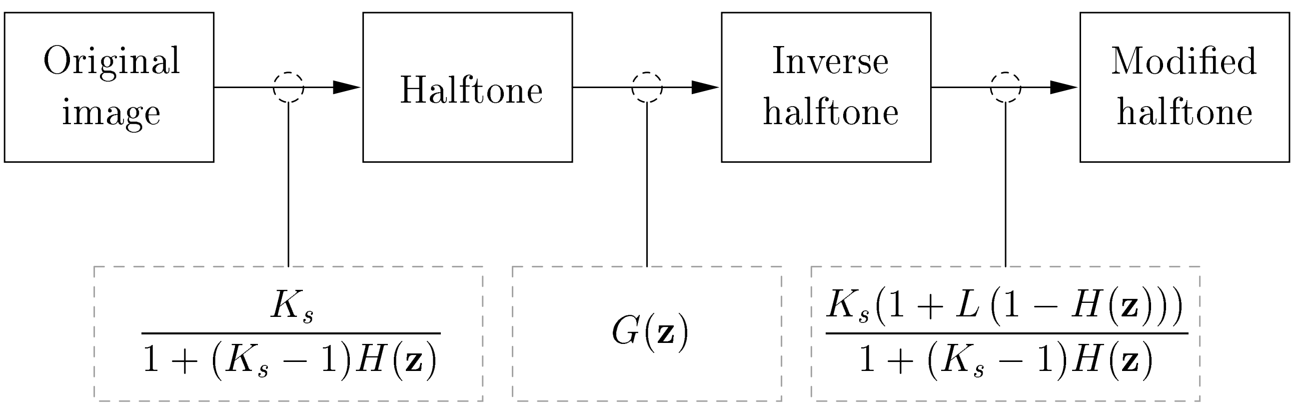
REHALFTONING

- Halftone conversion, manipulation
- Assume input and output are error diffused halftones
  - Blurring corrected by using modified error diffusion
  - Noise leakage masked by halftoning
  - 64 operations per pixel
- For a 512 x 512 image
  - 16 RISC MIPS
  - 0.4 s on a 167 MHz Ultra-2 workstation



REHALFTONING (cont.)

- Halftone conversion, manipulation
- Error diffused halftones
- Fixed lowpass inverse halftoning filter, compromise cut-off frequency
  - Noise leakage masked by halftoning
  - Correct blur by modified error diffusion
  - Computationally efficient



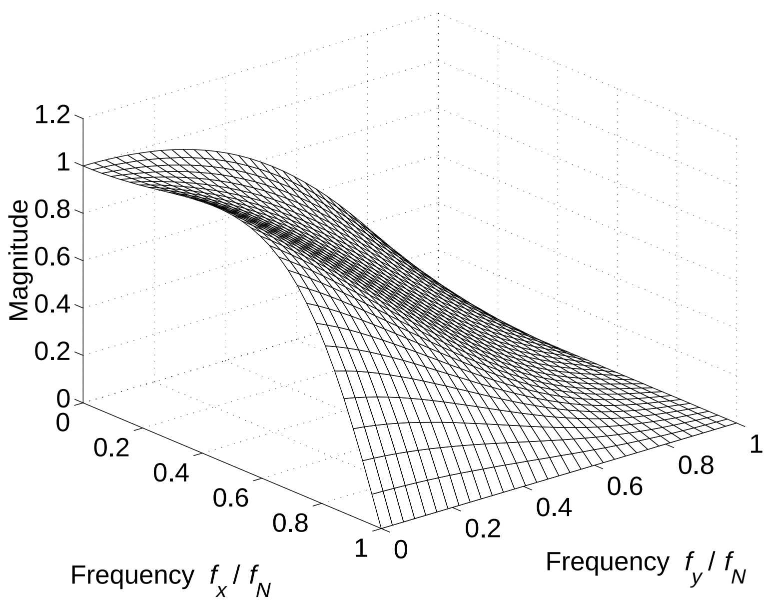
REHALFTONING (cont.)

- Use linear gain model to design L for flat response
- Use approximation for digital frequency: $e^{j\omega} \approx 1 + j\omega - \omega^2/2$
- Inverse halftoning filter is a simple separable FIR filter
- L is computed to flatten the end to end transfer function of the system
- We need to know halftoning filter coefficients for this scheme
- Improve halftoning results using knowledge of type of halftone being rehalftoned

REHALFTONING RESULTS

Original image

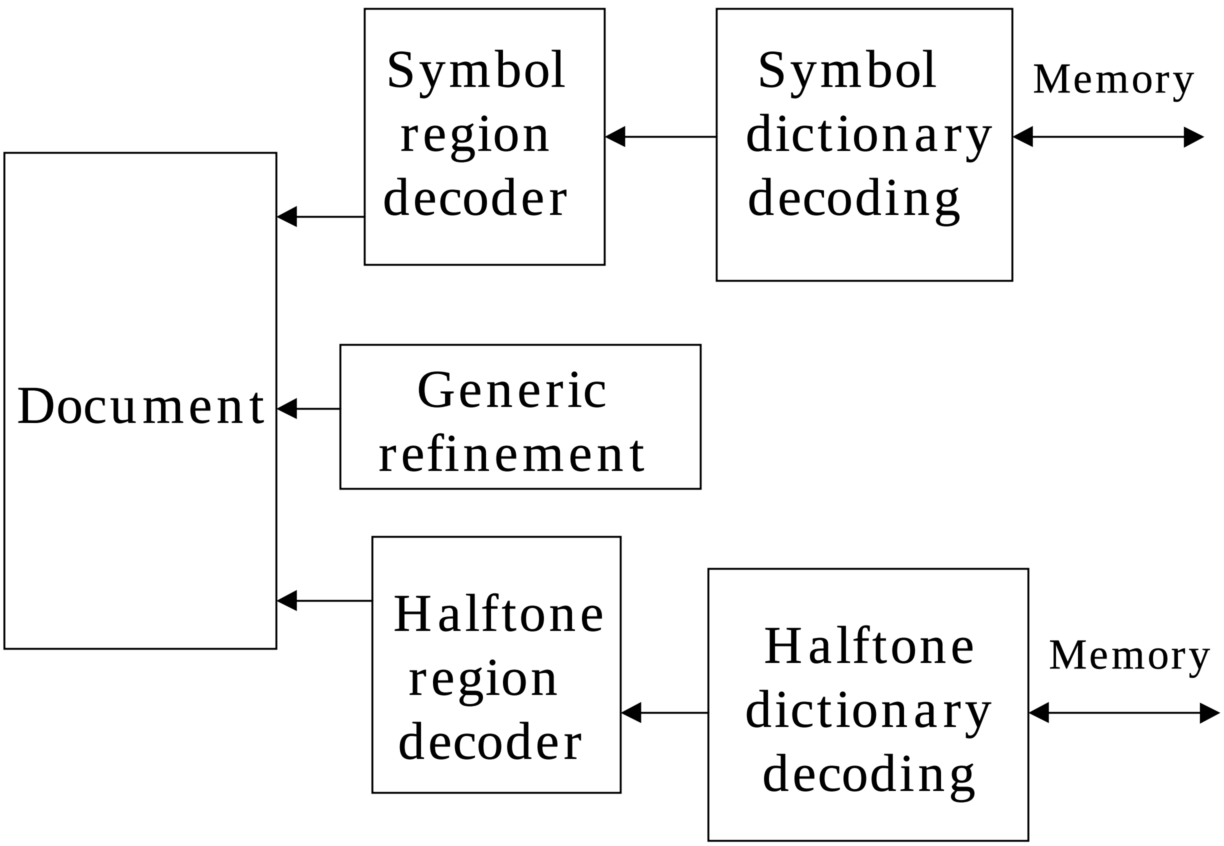
Rehalftone



Signal transfer function

THE JBIG2 STANDARD

- Lossy/lossless coding of bi-level text and halftone data



- Scan vs. random mode

THE JBIG2 STANDARD (cont.)

- Bi-level text coding
  - Hard pattern matching (lossy)
  - Soft pattern matching (lossless or near lossless) may be context based
- Halftone coding
  - Direct halftone compression
  - Context based halftone coding
  - Inverse halftoning and compression of grayscale image
- Implications
  - Printers, fax machines and scanners, will need to decode JBIG2 bitstreams
  - Fast decoding may require dedicated hardware and embedded software
  - Need for low complexity, low memory solutions

PROBLEMS TO BE SOLVED

- JBIG2 compression of halftones
  - Compress halftone directly, using a dictionary of patterns, or
  - Convert halftone to grayscale (inverse halftoning) and compress grayscale image
- Efficient coding of halftone data
  - Fax machines
  - Digital archiving, scanning, and copying
- Fast algorithms for JBIG2 codec
  - Interpolated halftoning in decoder
  - Rehalftoning in codec

PROBLEMS TO BE SOLVED

- JBIG2 embedded decoders
  - Low memory requirements
  - Low computational complexity
  - High parallelism
- Inverse halftoning: a robust solution for lossy coding of halftones
  - Rendering device can use a different halftoning scheme than encoder
  - Multiresolution halftone rendering (archive browsing)
  - High halftone compression ratios (6:1)
  - Quality enhancement if the encoder halftoning method is transmitted
- Low-cost embedded implementations

CONCLUSIONS

- Linear gain model of error diffusion
  - Validate accuracy of quantizer model
  - Link between filter gain and signal gain
- Rehalftoning and interpolation
  - Efficient algorithms
  - Impact on emerging JBIG2 standard
- Web site for software and papers
  - `http://www.ece.utexas.edu/~bevans/projects/inverseHalftoning.html`