

Colored Noise Inversion in Digital Halftoning

Robert Geist
Robert Reynolds

Department of Computer Science
Clemson University
Clemson, South Carolina 29634-1906

Abstract

A new digital halftone resolution technique, based on a neural network with connection strengths derived from stochastic power spectra, is proposed. Ulichney's "blue noise" spectrum is used as a basis; alternative spectra are also examined. The new technique is compared with standard resolution techniques, ordered dither, Floyd-Steinberg, and dot diffusion. The new technique appears to offer resolutions with improved image detail.

Keywords: Digital Halftone, Stochastic Power Spectra, Blue Noise, Neural Networks.

Introduction

The digital halftone resolution problem may be stated as follows: given an $n \times n$ array V of real numbers, $V_{i,j} \in [0, 1]$, produce an $n \times n$ array ω of binary integers, $\omega_{i,j} \in \{0, 1\}$, such that ω , when displayed on a binary output device such as a computer monitor or laser printer, is a "good" representation of the real information, the intensities contained in V . The obvious resolution algorithm, round the values in V , fails to satisfy most interpretations of "good". For instance, if $V_{i,j} = .4999999$ for all i, j , then $\omega_{i,j} = 0$ for all i, j , and a desired gray image is displayed as white. Consideration of neighborhood intensities seems imperative.

Many halftone resolution algorithms have been proposed (see [8]), and some of the more commonly used ones are described in the next section. One reason for this multitude of algorithms is that the usual measure of success for halftoning, the quality of the output image, has been gauged in a largely subjective manner. A notable exception to this subjective evaluation has appeared recently in [12, 13] where Ulichney offers a convincing case for the use of *stochastic power spectra* in measuring the success of certain classes of halftone resolution algorithms. With the support of numerous examples, he argues that high quality images are produced by those algorithms whose associated (radially averaged) power spectrum can be characterized as "blue noise," that is, a flat noise function shifted toward the higher frequency, "blue" end of the spectrum.

The advantage of such an approach is that it offers a purely objective measure of algorithm success as well as lending some quantification to visual aesthetics.

The purpose of this paper is to show that idealized power spectra, such as Ulichney's "blue noise", can be inverted to yield natural, deterministic algorithms for halftone resolution. The inversion itself yields a collection of pixel correlation coefficients which serve as connections in a *Hopfield neural network*. A fast neural network convergence algorithm provides the actual resolution.

The paper is organized as follows. In the next section we consider three commonly used halftone algorithms, ordered dither, Floyd-Steinberg, and dot diffusion. Then, we discuss Hopfield neural networks and their use in halftone resolution. Next, we compare the results of resolution by "blue noise" inversion with the more commonly used algorithms. In the following section we consider "red noise" and "green noise" inversion, and finally we offer conclusions.

We note that both neural networks and frequency domain considerations have been included in earlier halftone resolution algorithms. In [1] a frequency-weighted mean squared error function is minimized by mapping the function onto a neural network. The weights are determined from psychophysical experiments reported in [10]. In [4] we offer an algorithm for resolving $2^{10} \times 2^{10}$ pixel images using 2^{16} neural networks in parallel. Each network contains 16 neurons with connections determined by a pixel magnetism model. Convergence of each network is perfect; that is, energy minima are guaranteed. The approach described herein offers significantly stronger motivation for selection of the network interconnection parameters.

Standard Halftone Algorithms

The most commonly used resolution algorithm is probably the *ordered dither* [3], in which we tile the image matrix V with a smaller fixed array D of threshold values, and then turn on the pixel (set $\omega_{i,j} = 1$) if and only if $V_{i,j}$ exceeds the corresponding

threshold value. A standard 4×4 tile is

$$D = \begin{bmatrix} 1/32 & 17/32 & 5/32 & 21/32 \\ 25/32 & 9/32 & 29/32 & 13/32 \\ 7/32 & 23/32 & 3/32 & 19/32 \\ 31/32 & 15/32 & 27/32 & 11/32 \end{bmatrix}$$

Note that a uniform intensity of 0.5 would cause 8 of every 16 pixels (every other one) to be turned on. Also note that the entries in the tile are carefully chosen to break horizontal and vertical lines, which are easily recognized by the eye.

In figure 1 we show a 256×256 pixel image of a digitized photo resolved by this ordered dither. This image was printed on a conventional 300 pixel per inch laser printer at an expanded resolution of 75 pixels per inch.

Although the ordered dither algorithm can be highly parallel in implementation, one of the standard complaints lodged against this technique is that it imparts an artificial texture to the image. This "computery" look is quite evident in our figure.

A substantial improvement in smoothing can be achieved, at the expense of parallelism, by the Floyd-Steinberg algorithm [2]:

```
for(i=1 to n){
  for(j=1 to n){
    if  $V_{i,j} < 1/2$  then  $\omega_{i,j}=0$ ;
    else  $\omega_{i,j} = 1$ ;
    err =  $V_{i,j} - \omega_{i,j}$ ;
     $V_{i,j+1} = V_{i,j+1} + (\text{err} * 7/16)$ ;
     $V_{i+1,j-1} = V_{i+1,j-1} + (\text{err} * 3/16)$ ;
     $V_{i+1,j} = V_{i+1,j} + (\text{err} * 5/16)$ ;
     $V_{i+1,j+1} = V_{i+1,j+1} + (\text{err} * 1/16)$ ;
  }
}
```

The algorithm thus diffuses the rounding error at each step to adjacent image cells in a serial fashion. Applying this algorithm to the same data, we obtain the results shown in figure 2, where greater detail in the ring is available and the artificial texturing is substantially reduced. However, much ring detail is still obscured, no distinct features of the background are visible, and the signature initials in the lower right (which were at least hinted at in the dithered picture) have been obliterated.

In an effort to achieve both the smoothness of the Floyd-Steinberg algorithm and the highly desirable parallelism of the ordered dither, Knuth has recently developed *dot diffusion* [9]. In this algorithm the image V is tiled with a very carefully constructed matrix of "class" numbers. An 8×8 class matrix given in [9] is

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

The algorithm is then

```
for(k=0 to 63){
  for(all (i,j) of class k){
    if  $V_{i,j} < 1/2$  then  $\omega_{i,j}=0$ ;
    else  $\omega_{i,j}=1$ ;
    err =  $V_{i,j} - \omega_{i,j}$ ;
    distribute err to neighbors of higher class;
  }
}
```

The distribution of *err* sends twice as much to each orthogonal neighbor (of higher class) as it does to each diagonal neighbor (of higher class). The algorithm thus diffuses error in a manner similar to the Floyd-Steinberg algorithm, but achieves parallelism by dissipating the error at isolated pixels of high class ("barons") rather than at the edges of the image.

If we apply this algorithm to our image, we obtain the resolution shown in figure 3. The results are similar to those of the Floyd-Steinberg algorithm, although the appearance is somewhat grainier. Though both this algorithm and Floyd-Steinberg omit much detail, they are both far superior to the ordered dither.

Neural Networks

A neural network is a collection of simple analog processing elements designed to mimic biological neurons. The computational paradigm provided by such networks is a radical departure from that of the classical von Neumann architecture. The "input" to such a network is a matrix of interconnections among the processing elements, together with an initial voltage that is applied to each element. The networks are designed so that the stable output voltages of the analog elements are binary. The collection of all binary output levels is then interpreted as the "result" of the computation.

Of particular interest to us here is the class of neural networks proposed by Hopfield [6]. A four element example, from [11], is shown in figure 4. Neurons are represented by amplifiers, each providing both standard and inverted outputs (voltage $\theta_i \in [-1,1]$). Synapses are represented by the physical connections between input lines to the amplifiers and, in feedback, output lines from the amplifiers. Resistors are used to make these connections. If the input to amplifier i is connected to the output of amplifier j by a resistor with value R_{ij} , then the *conductance* of the connection is T_{ij} , whose

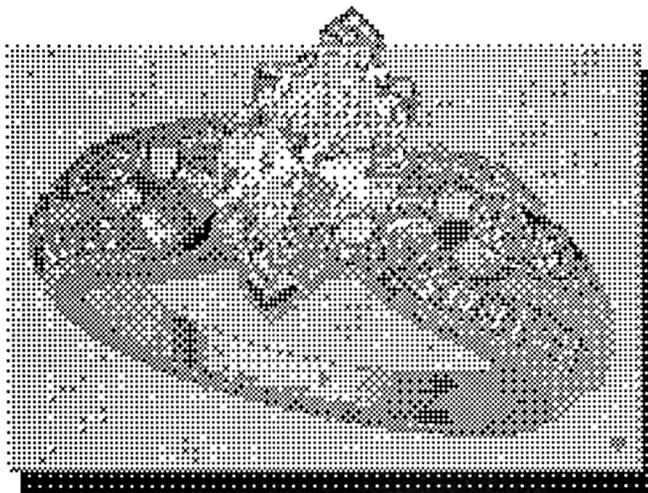


Figure 1: Digitized photo resolved by ordered dither.

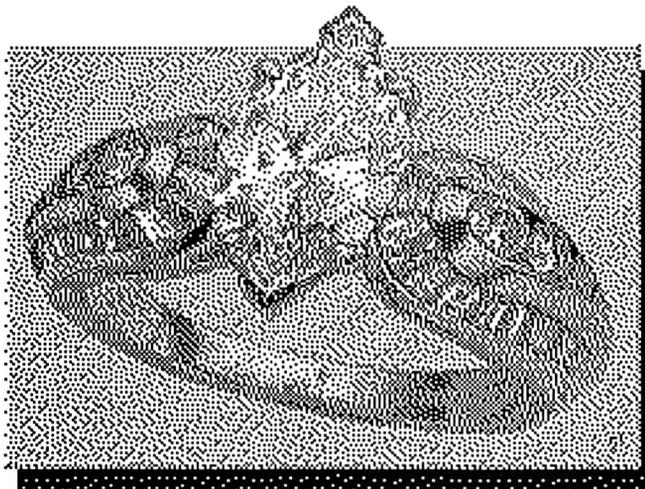


Figure 2: Digitized photo resolved by Floyd-Steinberg.

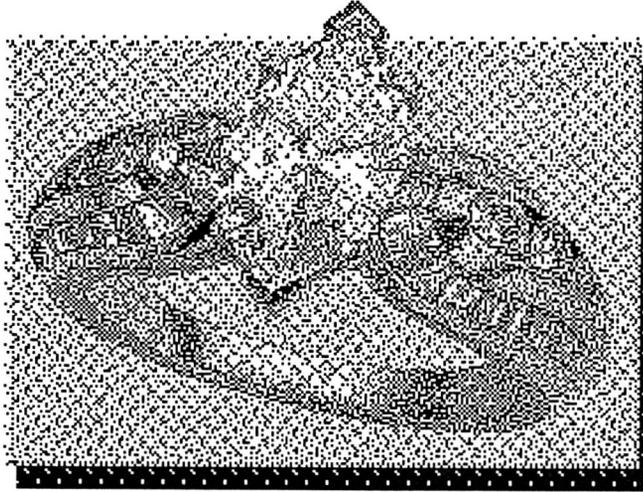


Figure 3: Digitized photo resolved by dot diffusion.

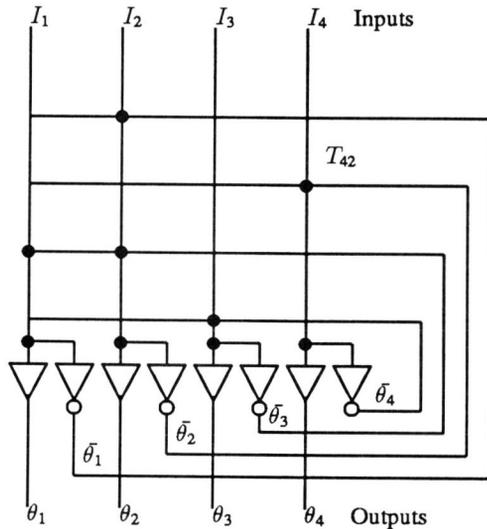


Figure 4: Four element Hopfield network.

magnitude is $1/R_{ij}$ and whose sign is determined by whether the connection to amplifier j is from the standard or inverted output. Hopfield showed that when the matrix T is symmetric with zero diagonal and the amplifiers are operated in "high-gain" mode, the stable states of the network are binary ($\{-1,1\}$) and are the local minima of the computational energy,

$$E(\theta) = (-1/2) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} T_{ij} \theta_i \theta_j - \sum_{i=0}^{N-1} \theta_i I_i. \quad (1)$$

Here I_i is the external input to amplifier i .

If we now choose to represent each pixel in our $n \times n$ display

by a neuron, then a binary stable state of a network of $N = n^2$ neurons, $\{\theta_k | k = 0, 1, \dots, N-1\}$, can be naturally regarded as a halftone resolution, namely $\omega_{i,j} = (\theta_{i+n+j} + 1)/2$. Of course a major difficulty with this approach is translation of the desired image intensities, the V matrix of the previous section, into network connection parameters, the $T_{i,j}$'s and the I_i 's. Reasonable choices abound, and it is the purpose of the next section to derive a "natural" selection.

We should note, however, that any selection of connection parameters is likely to benefit from consideration of global image intensity. Specifically, a reasonable constraint on resolution ω is that $\sum_{i,j} \omega_{i,j}$ should not deviate greatly from $\sum_{i,j} V_{i,j}$.

To incorporate the additional constraint, we use a variation on a technique suggested in [11]: let $m = \lfloor \sum_{i,j} V_{i,j} + 0.5 \rfloor$, the rounded total intensity. We now add to $E(\theta)$ a summand of the form

$$C \left(\sum_{i=0}^{N-1} \frac{\theta_i + 1}{2} - m \right)^2$$

where $C > 0$. To reinforce integral solutions while maintaining the Hopfield constraint of a zero-diagonal T matrix, we add another of the form

$$C \left(\sum_{i=0}^{N-1} \frac{1 - \theta_i^2}{4} \right).$$

The net effect is to add $C(m - N/2)$ to each I_i and $-C/2$ to each non-diagonal T_{ij} in (1).

The only remaining issue of network application is the method by which we obtain the stable state from the specified network. Hopfield networks of $N = 256^2 = 65536$ neurons have not yet been built, and we are forced to resort to net simulation.

Net simulation is traditionally approached (e.g. [11]) as a

numerical integration of the system of N differential equations describing the operation of the amplifiers [6]:

$$C_i du_i/dt = \sum_j T_{i,j} g(u_j) - u_i/R_i + I_i. \quad (2)$$

Here the u_i are internal input voltages to the amplifiers, and are related to the desired output voltages, the θ_i , by a sigmoidal gain function, $g(x)$. A reasonable choice for $g(x)$ is a scaled hyperbolic tangent, $g(x) = \tanh(\lambda x)$. Here λ is called the gain. The C_i are the input capacitances of the amplifiers, and $R_i = 1/(1/\rho + \sum_j |T_{i,j}|)$, where ρ is amplifier input resistance.

We have found numerical integration of large (2^{16} neuron) systems of the form (2) to be extremely time consuming, and therefore have developed an alternative approach. Any equilibrium of (2) is given by

$$0 = \sum_j T_{i,j} g(u_j) - u_i/R_i + I_i$$

that is,

$$u_i = R_i \left(\sum_j T_{i,j} g(u_j) + I_i \right)$$

or simply

$$u = G(u),$$

where $G(u) = \text{diag}(R)(Tg(u) + I)$, $\text{diag}(R)$ has R_i 's on the diagonal and 0's elsewhere, and $g(u) = (g(u_1), g(u_2), \dots)$. Thus we seek a fixed point of a certain N -dimensional function. If $\| \cdot \|$ denotes the max norm on Euclidean N -space and $\| \cdot \|$ its induced matrix norm, then since $R_i < 1/\sum_j |T_{i,j}|$ we have

$$\begin{aligned} |G(u) - G(u')| &= |\text{diag}(R)T(g(u) - g(u'))| \\ &\leq \|\text{diag}(R)T\| \cdot |g(u) - g(u')| \\ &< |g(u) - g(u')| \\ &\leq \lambda |u - u'| \end{aligned}$$

Thus, for gain $\lambda < 1$, convergence of the simple iteration scheme, $u^{k+1} = G(u^k)$, is straightforward (see, e.g. [7]). Unfortunately, the Hopfield result speaks only of high-gain operation, and we must consider $\lambda > 1$, where the simple iteration is likely to diverge. Fortunately, there is an intriguing alternative.

In [5] Hiram established a remarkable result for functions on the real line: if $f : [a, b] \rightarrow [a, b]$ satisfies $|f(x) - f(y)| \leq M|x - y|$, then the iteration scheme

$$x_{n+1} = \frac{1}{M+1} f(x_n) + \frac{M}{M+1} x_n \quad (3)$$

converges to a fixed point of f . On the conjecture that this result might extend to higher dimensions, Hiram noted that a completely new approach would be needed, since his proof relied heavily upon the total ordering of the real line. To our knowledge, this conjecture remains unresolved.

Nevertheless, we have found substantial empirical evidence to support it, at least within the neural net environment. Using (3) with $M = \lambda$, we find that convergence to a fixed point of G (that is, average component error $|u_i - G(u_i)| < 10^{-10}$) usually requires fewer than 150 iterations, even for these large (2^{16} neuron) systems. We have not found a net for which this scheme fails to converge.

Blue Noise Inversion

Hereafter we number pixels in our $n \times n$ array left-to-right and top-to-bottom using the non-negative integers and refer to intensities $\{V_i | i = 0, 1, 2, \dots, N-1\}$ and resolutions $\{\omega_i | i = 0, 1, 2, \dots, N-1\}$.

If we regard the value of pixel i , ω_i as a stationary stochastic process with mean μ , then its autocovariance sequence (in space) is given by

$$R_k = E[(\omega_i - \mu)(\omega_{i+k} - \mu)] \quad k = 0, 1, \dots$$

where $R_0 = \sigma^2$, the variance. The associated autocorrelation sequence is given by

$$\rho_k = \frac{R_k}{R_0} \quad k = 0, 1, \dots$$

and the power spectrum is the Fourier transform,

$$P(\lambda) = \sum_{k=0}^{+\infty} R_k \cos(k\pi\lambda) \quad -1 \leq \lambda \leq 1$$

Note that zero spatial correlation of pixel value would give $R_k = 0, k > 0$ and hence $P(\lambda) = \sigma^2$, all λ . A constant frequency function such as this is often regarded as white noise.

In [12] Ulichney argues that for stochastic halftone resolution of a fixed intensity, $V_i = \mu$, all i , it is the low frequency noise that gives rise to unpleasant visual effects. If we remove low frequency noise from the white noise function, $P(\lambda) = \sigma^2$, we obtain the spectrum of figure 5 with a symmetric picture at negative frequencies. Here f_p is the so-called principal frequency which is taken, following [12], from the desired constant intensity μ :

$$f_p = \begin{cases} \frac{\sqrt{\mu}}{\sqrt{1-\mu}} & 0 \leq \mu \leq 1/2 \\ \frac{1}{\sqrt{1-\mu}} & 1/2 < \mu \leq 1 \end{cases}$$

We now observe that this idealized spectrum is easily inverted. In general the inverse transform is given by

$$R_k = \int_{-1}^1 P(\lambda) \cos(k\pi\lambda) d\lambda.$$

In this instance we obtain

$$\begin{aligned} R_k &= 2 \int_{f_p}^1 \frac{\sigma^2}{2(1-f_p)} \cos(k\pi\lambda) d\lambda \\ &= \frac{\sigma^2}{1-f_p} \left[\frac{-\sin(k\pi f_p)}{k\pi} \right] \end{aligned}$$

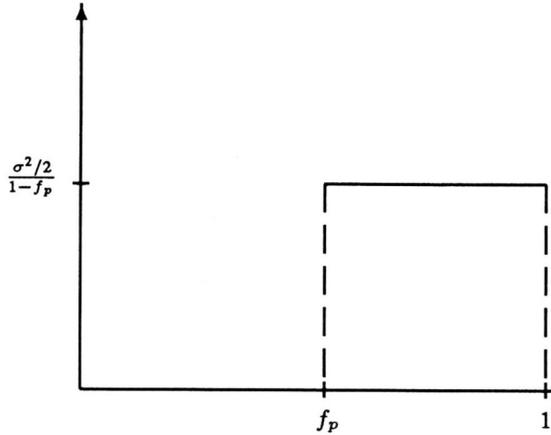


Figure 5: Blue noise.

and so

$$\rho_k = \frac{-\sin(k\pi f_p)}{(1 - f_p)k\pi}$$

We can take this correlation coefficient ρ_k as the desired relative strength of connection between two pixels at distance k in a halftone resolution of a region of fixed intensity value μ . We contend that when intensities in a region deviate markedly from a fixed value, i.e. high standard deviation of intensities, the image is likely to contain significant detail in the region, and these connection strengths should be reduced accordingly to yield to individual cell intensity values.

Thus our connection structure is specified as follows: for each pixel i let μ_i and σ_i denote the mean and standard deviation of the intensities, the V 's, in a neighborhood of radius R about pixel i . If pixel j is at distance $k \leq R$ from i , where planar distance is given by the sum metric, $\|(x_1, y_1) - (x_2, y_2)\| = |x_1 - x_2| + |y_1 - y_2|$, we set

$$T_{i,j} = \frac{-\sin(k\pi\sqrt{\mu_i})}{(1 - \sqrt{\mu_i})k\pi(1 + \sigma_i)}$$

and

$$I_i = V_i - K \times A_i$$

where A_i is the mean of the intensities of the pixels adjacent to i , and K is a constant. The effect of the latter assignment is to give each cell an externally imposed tendency.

We note that the resulting T matrix is not necessarily symmetric, but symmetry is a sufficient condition for convergence, not a necessary one, and in practice the asymmetry causes no problem.

In figure 6 we see the results of applying to the same data our neural net with gain 1.6 and neighborhood radius 5. Convergence required 116 iterations. Greater detail is visible in the ring, for instance around the clasps. The marbling in the background is evident for the first time, as is the signature

initial in the lower right. Initials on the inside of the ring band, not even suggested by the other resolution methods, are beginning to emerge.

Red Noise and Green Noise

If blue noise is defined as in the previous section, then one might define "red noise" as the inverse of blue noise, with a spectrum as shown in figure 8. This idealized spectrum is also easily inverted:

$$\begin{aligned} R_k &= 2 \int_0^{f_p} \frac{\sigma^2/2}{f_p} \cos(k\pi\lambda) d\lambda \\ &= \frac{\sigma^2}{f_p} \left[\frac{\sin(k\pi f_p)}{k\pi} \right] \end{aligned}$$

And so

$$\rho_k = \frac{\sin(k\pi f_p)}{k\pi f_p}$$

We also chose to investigate a spectrum with a central non-zero component. We define "green noise" to have the spectrum of figure 9. This spectrum is also easily inverted:

$$\begin{aligned} R_k &= 2 \int_{\frac{f_p}{2}}^{\frac{1+f_p}{2}} \sigma^2 \cos(k\pi\lambda) d\lambda \\ &= 2\sigma^2 \left[\frac{\sin\left(\frac{k\pi(1+f_p)}{2}\right) - \sin\left(\frac{k\pi f_p}{2}\right)}{k\pi} \right] \end{aligned}$$

And so

$$\rho_k = 2 \left[\frac{\sin\left(\frac{k\pi(1+f_p)}{2}\right) - \sin\left(\frac{k\pi f_p}{2}\right)}{k\pi} \right]$$

Now, specifying our interconnection structure in a way analogous to that for blue noise, we create a red noise neural net and a green noise neural net. Applying these nets to our data results in pictures which, while clearly inferior to figure 6, are not as poor as one might expect if the defined blue noise spectrum were the ideal for improved resolution. The resolution obtained via the red noise algorithm is shown in figure 7.

Conclusions

We have considered the development of new algorithms for digital halftone resolution, based on Ulichney's use of stochastic power spectra. We have shown how idealized spectra may be easily inverted to produce connection coefficients in a Hopfield neural network which forms the basis for these algorithms. When compared with standard resolution techniques, ordered dither, Floyd-Steinberg, and dot diffusion, we find our "blue noise" algorithm to offer improved resolution detail.

We have considered not only Ulichney's "blue noise" spectrum, but also some alternatives: a "red noise" spectrum of

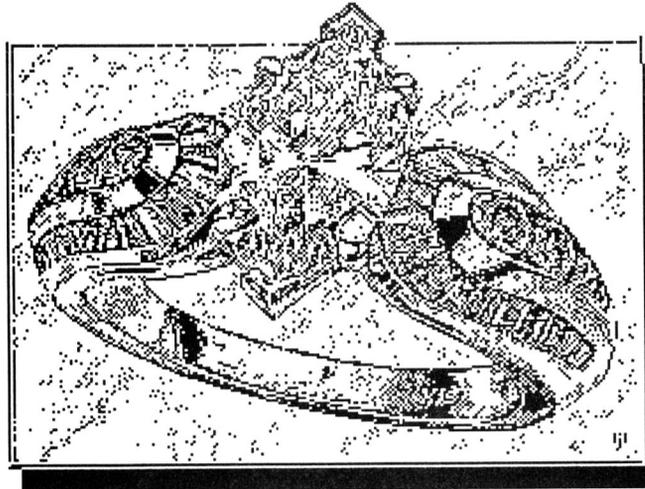


Figure 6: Digitized photo resolved by blue noise inversion.

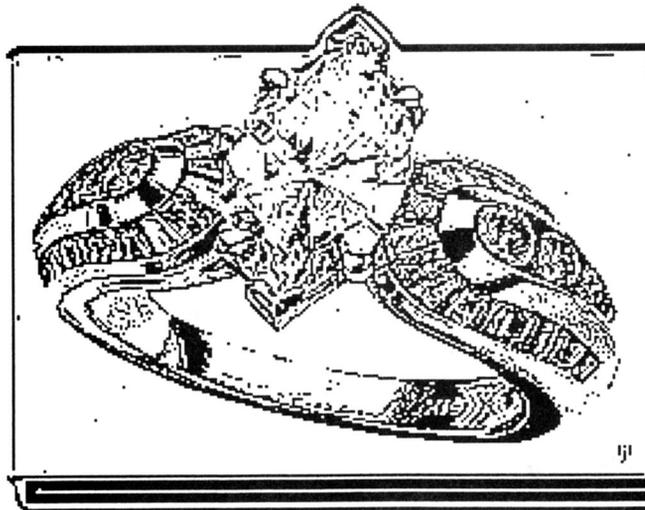


Figure 7: Digitized photo resolved by red noise inversion.

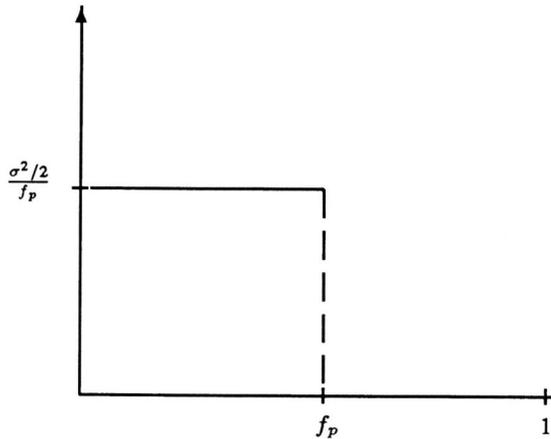


Figure 8: Red noise.

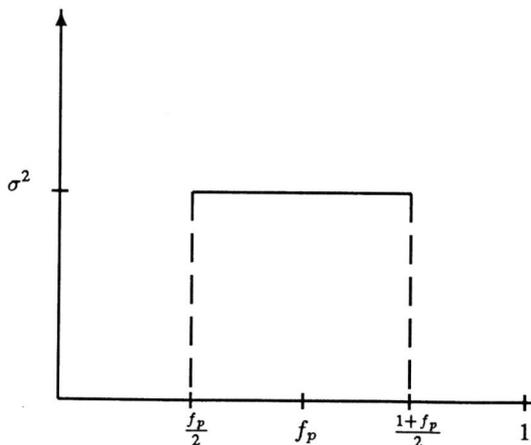


Figure 9: Green noise.

low frequencies and a "green" noise spectrum of central frequencies. The high frequency blue spectrum definitely gives the best visual results, but the resolutions provided by the other spectra encourage further experimentation with spectrum definition.

References

- [1] D. Anastassiou. Neural net based digital halftoning of images. *Proc. IEEE Int. Symp. on Circuits and Systems*, June 1988.
- [2] R. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *SID 75 Digest*, pages 36-37, 1975.
- [3] J. Foley and A. Van Dam. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, Reading, Massachusetts, 1984.
- [4] R. Geist, G. Roa, and R. Reynolds. A gibbs measure approach to digital halftones for ray-traced images. *Tech. Rep., Dept. of Comp. Sci., Clemson U.*, 1989.
- [5] B.P. HILLAM. A generalization of krasnoselski's theorem on the real line. *Math. Mag.*, 48:167-168, 1975.
- [6] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci.*, 81:3088-3092, 1984.
- [7] E. Isaacson and H.B. Keller. *Analysis of Numerical Methods*. John Wiley & Sons, New York, 1966.
- [8] J. Jarvis, C. Judice, and W. Ninke. A survey of techniques for the display of continuous tone pictures on bilevel displays. *Comp. Graphics Image Process.*, 5:13-40, 1976.
- [9] D. Knuth. Digital halftones by dot diffusion. *ACM Trans. on Graphics*, 6:245-273, 1987.
- [10] J. Mannos and D. Sakrison. The effects of a visual fidelity criterion on the encoding of images. *IEEE Trans. on Inf. Theory*, IT-20, July 1974.
- [11] E. Page and G. Tagliarini. Algorithm development for neural networks. In *Proc. SPIE Symp. on Innovative Science and Technology*, Los Angeles, CA, January 1988.
- [12] R. Ulichney. *Digital Halftoning*. MIT Press, Cambridge, Massachusetts, 1988.
- [13] R. Ulichney. Dithering with blue noise. *Proc. of IEEE*, 76:56-79, 1988.