

# Backward Diffusion Methods for Digital Halftoning

Industrial Representative: C.W. Wu (IBM)

Faculty: P. Dubovski (Stevens), J. Harlim (Maryland, College Park), C. Please (Southampton, UK), D. Schwendeman (RPI), B.M. Vernescu (WPI)

Students: C.L. Chen (SUNY, Buffalo), Y. Cheng (NJIT), M. Gratton (Duke), H. Hu (Indiana), T. Kiehl (RPI), D. Onofrei (WPI), N. Zeev (Delaware), Y. Zhang (NJIT), Y. Zhou (Maryland, Baltimore County)

Summary Presentation: J. Harlim, 16 June 2006.

Report Preparation: M. Gratton

## Abstract

We examine using discrete backward diffusion to produce digital halftones. The noise introduced by the discrete approximation to backwards diffusion forces the intensity away from uniform values, so that rounding each pixel to black or white can produce a pleasing halftone. We formulate our method by considering the Human Visual System norm and approximating the inverse of the blurring operator. We also investigate several possible mobility functions for use in a nonlinear backward diffusion equation for higher quality results.

## 1 Introduction

Halftoning involves transforming a normal continuous-tone image into an array of black-and-white dots. These dots, collectively called a *halftone*, should resemble the original image at a comfortable viewing distance in an effect very similar to stippling in print engraving. Printers are only binary devices; they either place ink at a location on a page or they do not. In this sense, all printed images are half-toned if they originally contain shades different from the color paper and ink. While color images can also be halftoned, this report considers only the black-and-white case, where grays are represented by patterns of black dots on white paper. There are two difficulties in finding a halftone. First, at poor resolutions, it is difficult to convey varied shades of gray because the resolution constraints what patterns are possible. Sufficient resolution is needed so that the viewer interprets the dots as a continuous tone. Secondly, at high resolutions, generating a halftone requires operating on very large problems. Modern printers may print more than six hundred dots per linear inch, meaning a full-page picture may involve an array of 25 million dots or more. Efficiency in the algorithm used to halftone is therefore essential.

Traditional printing employed screening techniques from the 1890's to the 1980's, where a photograph of the graphic is taken through a special screen to produce different sized dots to vary the perceived shade [9]. Digital halftoning with uniform dot sizes using computers has since been the preferred method. Of the new techniques, one of the most popular is

Floyd Steinberg error diffusion (FS), where the image is halftoned in one processing pass over all of the pixels [3]. The difference between each dot and the corresponding location on the picture is distributed according to particular weights to the left and lower neighbors of the dot, and so “diffusing” the error. This method produces generally pleasing results, but has a tendency to produce regions with maze-like patterns. These patterns obstruct the eye’s tendency to blur fine detail and harm the halftoned effect. Research suggests that the proper distribution of dots should be blue noise, where the power spectrum goes like the inverse of the frequency, to best enhance the perception of the halftone as a continuous-tone image [9].

First, we establish some basic notation. An image  $u$  is an  $N \times M$  grid of points with intensities  $u_{ij} \in [0, 1]$ , where 0 is fully white and 1 is fully black. A halftone  $h$  only has values  $h_{ij} \in \{0, 1\}$ . Each  $h_{ij}$  represents a decision by a printer to print or not to print a dot at a location on a page corresponding to  $(i, j)$ . Since the halftone is generated to please the human eye, the results are evaluated using a Human Visual System (HVS) norm. This norm operates by blurring slightly both the original image and the halftone then taking the  $l^2$  norm of the difference:

$$\|h\|_u^2 = \sum_{i,j=1,1}^{i,j=N,M} (B(h)_{ij} - B(u)_{ij})^2, \quad (1)$$

where  $B : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{N \times M}$  is a blurring operator modeling the HVS, usually through convolution with a two dimensional Gaussian kernel. Better models of human vision are of possible, but we choose this one because of its simplicity. Note that once  $B$  is applied to  $h$ ,  $B(h)_{ij}$  is no longer necessarily an integer. Since a good halftone has a small norm, searching for an  $h$  that globally minimizes  $\|h\|_u$  is one possible algorithm; however, the size of the problem makes this computationally infeasible. Instead, algorithms such as Direct Binary Search (DBS) look for local minima from some starting halftone. DBS is considered a high-quality algorithm, but the time to converge is too long for many applications.

This report explores a halftoning method that attempts to approximate  $B^{-1}$ , a backward diffusion process that sharpens the image and generates noise due to its instability. This noise is exploited to provide texture to regions of little variation in the original image. It works as follows. Starting with an image  $u$ , compute  $B(u)$  the blurred image. Apply the approximation to the inverse

$$C \approx B^{-1} \quad (2)$$

to get  $C(B(u))$ . Use the high-frequency noise in this image to generate a half-tone  $h \approx C(B(u))$ . Now, when evaluating the fidelity of halftone  $h$ , the HVS norm computes  $B(h)_{ij} - B(u)_{ij}$ , but

$$B(h)_{ij} \approx B(C(B(u)))_{ij} \approx B(B^{-1}(B(u)))_{ij} = B(u)_{ij}. \quad (3)$$

As long as the approximations above are close, the halftone will have a small HVS norm.

To approximate  $C$ , this report explores using a backward diffusion evolution equation whose forward-diffusion cousin accomplishes the blurring operation  $B(u)$  by running for some fixed time on initial data  $u$ . Running the equation backward in time sharpens the image, provided this operation can be made stable.

Partial differential equations methods in image processing have had several successes. Diffusion techniques in image processing are well know (for instance, [1]). The celebrated Perona-Malik equation [7] [5] is a fourth-order parabolic equation that respects edges while providing some diffusion elsewhere. The equations we consider are only second order and rely on the details of the nonlinear mobility function to achieve varied results. Forward and backward diffusion together has also been used in image processing by Gilboa *et. al.* [4], but they apply it to continuous toned images without considering the problem of halftoning. Sapiro [8] has an overview of the use of PDEs in image processing.

This report will begin by establishing the partial differential equations that will be used in the image processing and the challenges posed by backward diffusion. We will proceed to discretize those equations and describe how discretization and limiters can be used to overcome these challenges. Next, we apply the discretized equations to images and produce halftones be either simple rounding or error diffusion, presenting the results of these algorithms. We conclude with a discussion of this PDE-based method as related to established halftoning methods.

## 2 Partial differential equation framework

Consider a function  $v(x, y, t)$  continuous in all variables. The restriction of  $v$  at a particular time to an  $N \times M$  grid gives an image. We want to use the continuity of  $v$  to consider image processing from the standpoint of partial differential equations. The variable  $t$  is an artificial evolution parameter, useful for controlling the strength of an operation.

We will focus on the second order parabolic general diffusion equation,

$$\frac{\partial v}{\partial t} = \nabla \cdot (M(v)\nabla v), \quad (4)$$

where  $M(v)$  is the *mobility function* that sets the local rate of diffusion based on the value of  $v$ . For forward diffusion,  $M(v) \geq 0$  and basic linear diffusion corresponds to  $M(v) = D$ , a constant. Boundary conditions should be chosen to preserve total image intensity

$$I(v) = \int_{\Omega} v \, dx dy, \quad (5)$$

where  $\Omega$  is the full domain. While periodic boundary conditions would easily satisfy this requirement, there is no reason to believe in general that images will be periodic, and they could create artifacts at the edges. Instead, we apply no-flux conditions

$$\nabla v \cdot \mathbf{n} = 0 \quad (6)$$

at  $\partial\Omega$ .

For a given mobility  $M$ , (4) can be solved by convolving the initial data with a Green's Function [2]. If the Green's Function at some particular time (say  $t = 1$ ) is the same as the kernel used to compute the blurring operator  $B$ , then the action of  $B$  can be performed by solving (4) from  $t = 0$  to  $t = 1$ . The details of matching  $B$  and the Green's Function will be discussed later.

This work is primarily concerned with *backward diffusion*, that is to run (4) backwards in time to approximate  $B^{-1}$ . To avoid confusion, we consider

$$\frac{\partial v}{\partial t} = -\nabla \cdot (M(v)\nabla v), \quad (7)$$

with  $t > 0$ . Again, we impose no-flux conditions (6) at the domain boundaries to preserve intensity. However, there are serious problems with solving this equation.

First, the equation is ill-posed. To see this, we compute the Fourier transform of the one-dimensional version of (7) with  $M(v) = 1$  and solve the resulting differential equation to get

$$\hat{v}(k, t) = \hat{v}_0(k) \exp(k^2 t), \quad (8)$$

where  $\hat{v}_0(k)$  is the transform of the initial data. All modes are growing exponentially, with the high-frequency modes growing fastest. The solution will become undefined instantaneously unless we impose a maximum frequency  $k_{\max}$  on the initial condition to provide an upper bound on the growth of the solution. Other mollifications can be performed on the equation, such as those discussed by Laurent [6], but these increase computational complexity.

Second, there are issues with intensity preservation. Taking the time derivative of the intensity and using (7),

$$\begin{aligned} \frac{dI}{dt} &= \int_{\Omega} -\nabla \cdot (M(v)\nabla v) \, dx dy \\ &= \int_{\partial\Omega} -M(v)\nabla v \cdot \mathbf{n} \, ds \\ &= 0, \end{aligned} \quad (9)$$

making use of (6) to eliminate the boundary terms, shows that intensity is preserved. Notice that the sign on the left side of (7) is irrelevant; the same is true for (4). However, forward diffusion has a maximum principle and a corresponding minimum principle, so that if initial data is constrained to have  $v(x, y, 0) \in [0, 1]$ , then it will remain in that interval for all  $t > 0$ . For backward diffusion, no such principle exists. In fact, while (9) technically is still true, it is accomplished by allowing the solution to become negative so as to support large positive spikes. Thus, there is no expectation that the value of  $v(x, y, t)$  will be in the unit interval for  $t > 0$ . These intensity values outside  $[0, 1]$  are not physical and are undesirable.

These two issues, while they seem daunting, are perhaps advantageous. The “noising” aspect of backward diffusion provides texture for solid toned patches that would otherwise be rounded to solid black or white blobs or distracting patterns. In the next section we show that the discretization can ameliorate the worst of the noise problems and suggests methods for controlling the solution within the physically relevant range.

### 3 Discretized formulation

There are many choices one could make in discretizing (7). We will guide our choices by speed and simplicity, as these methods must operate quickly on very large problems. To

this end, we choose an explicit first order in time, second order in space difference scheme. It is convenient to fix  $\Delta x = \Delta y = 1$ , as the scale of continuum variables  $x$  and  $y$  is arbitrary. Since  $t$  is artificial, we will not consider  $\Delta t$  directly. Instead, we introduce an amplification parameter  $\alpha$  analogous to

$$\alpha = \frac{\Delta t}{\Delta x^2 \Delta y^2} \quad (10)$$

from discretization. The size of  $\alpha$  and the total number of time-steps  $K$  can be chosen simultaneously to match the kernel of  $B$ .

The discretization is

$$\begin{aligned} u_{i,j}^{k+1} &= u_{i,j}^k - \alpha (J_r - J_l + J_u - J_d), \\ J_r &= M(u_{i+1/2,j}^k)(u_{i+1,j}^k - u_{i,j}^k) \\ J_l &= M(u_{i-1/2,j}^k)(u_{i,j}^k - u_{i-1,j}^k) \\ J_u &= M(u_{i,j+1/2}^k)(u_{i,j+1}^k - u_{i,j}^k) \\ J_d &= M(u_{i,j-1/2}^k)(u_{i,j}^k - u_{i,j-1}^k), \end{aligned} \quad (11)$$

where each  $J$  is a discrete flux in one of four grid directions and

$$u_{i+1/2,j}^k = [u_{i,j}^k + u_{i+1,j}^k]/2, \quad (12)$$

the average of the adjacent integer-numbered grid-points. The boundary condition (6) corresponds to taking the appropriate discrete flux to be zero. For example, along the left boundary at  $i = 1$ ,  $J_l = 0$ . The amplification  $\alpha$  should be positive for backward diffusion and negative for forwards diffusion.

As an example, let us suppose  $B$  is accomplished through the discrete convolution of  $u$  with the kernel

$$G(x, y) = \frac{1}{4\pi a} \exp\left(-\frac{x^2 + y^2}{4a}\right), \quad (13)$$

where  $a$  is the strength of the blur. To simulate this convolution using (11), we choose  $M(u) = -1$  so that the Gaussian is the Green's function of the related PDE and set

$$\alpha = a/K. \quad (14)$$

This method does not compute the action of  $B$  exactly, but introduces errors of magnitude  $O(\alpha)$ . As  $\alpha$  and  $K$  are related by (14), we can choose to take multiple small steps for accuracy or fewer steps for speed. While this is not the most useful method for computing a convolution, it does suggest how to choose  $\alpha$  and  $K$  for approximating  $B^{-1}$ .

Before discussing (11) for  $\alpha > 0$ , we should note that the simple fact of computing a discrete solution imposes an effective maximum frequency on the initial data for (11) and the related (7). The highest frequency that can be distinguished on a grid is three grid cells long (black to white to black, say), and so

$$k_{\max} \leq \frac{2\pi}{3} \quad (15)$$

for any initial data  $u_{i,j}^0$ . The issue of the solution taking values outside the unit interval can be addressed in a number of ways. One method was to consider the case where  $M(u)$  was not constant. The initial thought was to choose  $M(u)$  so that where the solution was in the unit interval the equation acted as a backward diffusion, but where the solution was outside the unit interval the equation acted as a forward diffusion. An example of such a function would be

$$M_2(u) = 4u(1 - u) , \quad (16)$$

where the constant normalizes  $M_2(u)$  to have an average strength of one over  $u \in [0, 1]$ . We considered a number of such functions and will discuss them later. One approach adopted within the discretization was to choose some  $M(u)$  while the solution was in the unit interval but to use a crude “limiter” at any time that the solution strayed outside  $[0, 1]$ . This limiter was implemented the following way. If after taking a timestep a value of  $u_{ij}$  was greater than unity, the excess was redistributed amongst the neighboring grid points. Hence,

$$\begin{aligned} \gamma &= \max(0, u_{ij} - 1) \\ u_{ij} &:= 1 \\ u_{i-1,j} &:= u_{i-1,j} + \gamma/4 \\ u_{i+1,j} &:= u_{i+1,j} + \gamma/4 \\ u_{i,j-1} &:= u_{i,j-1} + \gamma/4 \\ u_{i,j+1} &:= u_{i,j+1} + \gamma/4 \end{aligned} \quad (17)$$

A similar procedure was adopted if  $u_{ij}$  became negative. The limiter was applied between iterations in a single pass over all grid points. This procedure forces the value back into the unit interval, but, because it is applied in a sweep across the region it cannot guarantee values remain within the unit interval. The method does however provide strong diffusion to any points that stray outside the physically meaningful range.

As an example of the backward diffusion process, we consider two mobility functions, (16) and

$$M_1(u) = 1, \quad (18)$$

the former providing backwards diffusion for  $u \in [0, 1]$  and forwards diffusion otherwise, and the later representing linear backwards diffusion applied evenly to all intensities. We use an initial condition  $u^0$  that has a region of constant gradient with adjoining portion of a cosine function, as shown in figure 1. This function has a discontinuity in the first derivative at point  $A$  and a discontinuity in the second derivative at point  $B$ . The linear mobility excites noise at the maximum frequency fastest at  $A$ , while the logistic mobility excites the function fastest at  $B$ . Detail of the cosine portion of the initial condition shows that it is growing slowly, as predicted by the linear analysis. The constant gradient portion of the initial condition is unchanged except at its edges. This happens because regions of constant gradient will change slowly, as the fluxes in (11) will exactly cancel. While edge-effects from the no-flux boundaries will cause slow change, these regions will not experience the same noising as the rest of the image.

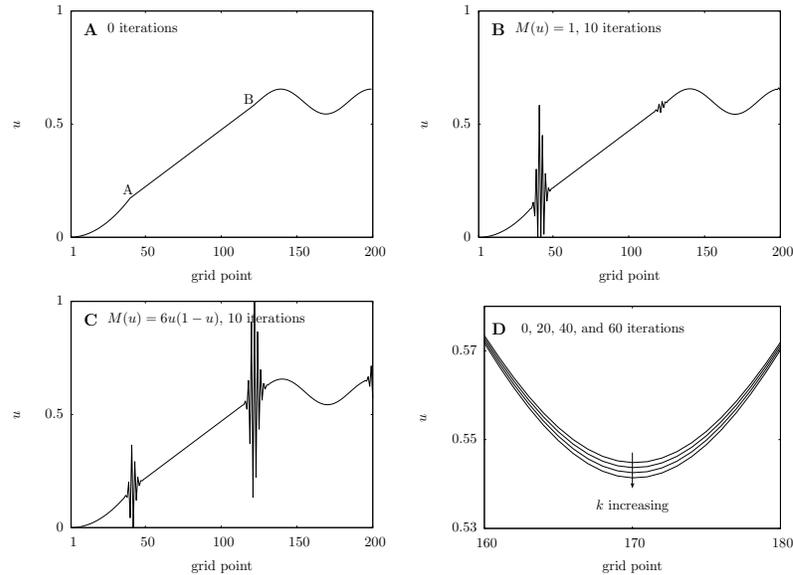


Figure 1: Backward diffusion in one dimension. The initial condition (A) has a discontinuity in the first derivative at  $A$  and a discontinuity in the second derivative at  $B$ . The linear mobility function excites the highest frequency (period three) oscillations quickly at  $A$ , while the logistic mobility function excites oscillations fastest at  $B$ . Detail of the cosine piece of  $u$  seen in (D) shows that this low frequency is also growing, but very slowly. In each case, the area of constant gradient in the middle of the domain is unchanged.

## 4 Backward diffusion halftoning

In this section, we develop the outline of an algorithm for halftoning and discuss strategies for picking mobility functions and amplification factors.

To halftone an image  $u^0$ , we do the following:

1. Optionally add a small amount of white noise to break up constant gradients.
2. Compute  $B(u^0) = w^0$  through convolution with the HVS kernel  $G$ .
3. Choose  $\alpha$  and  $K$  by (14) so as to match  $G$ .
4. Take  $K$  steps of (11) with a reasonable choice of  $M$  to produce  $w^K$ , applying the limiter (17) after each step.
5. Halftone the result to produce  $h$  by either
  - (a) rounding values to  $\{0, 1\}$ , or
  - (b) applying Floyd-Steinberg error diffusion.

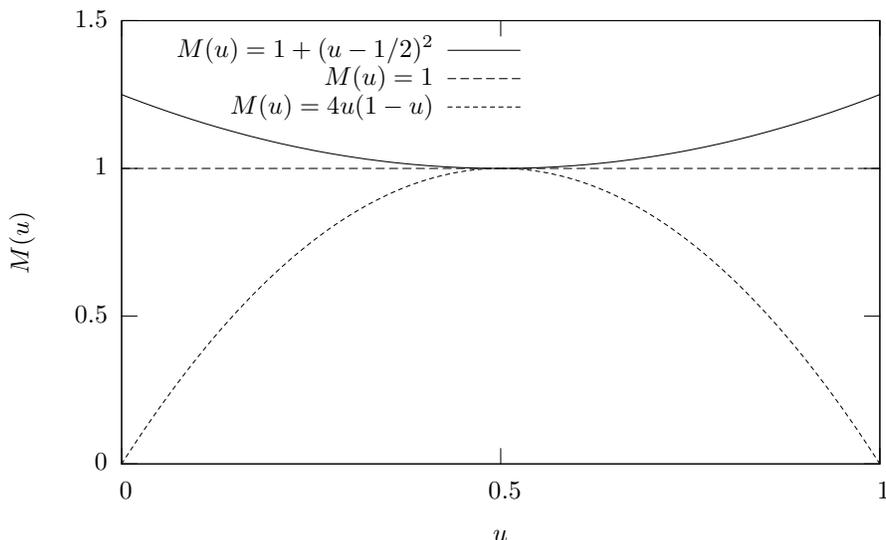


Figure 2: The three mobilities considered for backward diffusion halftoning.

Adding noise can accelerate the process, requiring fewer iterations to produce good data for the halftoning step, but it also introduces speckles that make it difficult to have a more accurate approximation of  $u$  as  $B(h)$  differs by at least the added noise from the original image.

The appropriate mobility function for halftoning is not obvious. We considered three possibilities, (18), (16), and

$$M(u) = 1 + (u - 1/2)^2. \quad (19)$$

This third possibility applies at least linear backward diffusion to all points, but provides additional backward diffusion for extreme points. This causes faster oscillations in bright and dark patches to create more variation in the halftone. It also discards forward diffusion entirely, relying on the limiter (17) to provide corrections for extreme values. Figure 2 shows each of the mobility functions used.

To test these mobilities, we considered a one-dimensional problem that was a linear gradient sampled at one hundred locations with a small amount of mean zero noise applied. We then applied (11) with different mobilities for twenty iterations. At each step, the data was post-processed by rounding to  $\{0, 1\}$ , generating a halftone. The results (shown in figure 3) show how variation appears from the initial halftone. Initially, the halftone is white from  $0 \leq i \leq 50$  and black elsewhere. As iterations of (11) are applied, the gradient is broken up, yielding halftones with more variation. A good halftone should have a cluster of white dots on the left, then gradually increase the frequency of black dots progressing to the right. The linear mobility function produces maximum frequency noise at the point in the initial condition corresponding to  $u_i = 0.5$  and slight variation elsewhere. The logistic mobility function produces much more rapid change across the entire domain, leaving the

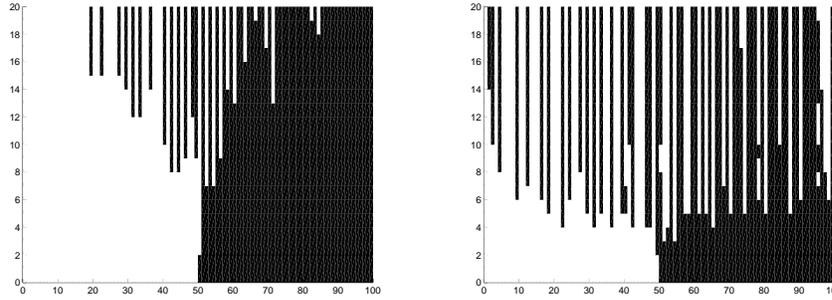


Figure 3: One dimensional halftoning of a linear gradient one hundred pixels long with some small noise added. The discrete scheme was applied for twenty iterations. Each of these twenty steps was post-processed by rounding to produce a halftone which is plotted along the  $x$  axis. The left image was made using the linear mobility function and shows a slow growth in variation, while the right image was made using the logistic mobility function and shows rapid variation that quickly overwhelms the linear gradient.

final halftone a collection of random dots. Even terminating before twenty iterations, the result is unsatisfactory.

In two dimensions, we computed halftones using mobility functions (18), (16), and (19), then rounding the results. These images appear in figure 4, along with halftones from DBS and direct FS. Below each is the halftone blurred by a Gaussian blur, approximating the image seen by the HVS norm. The image sharpening performed by the backward diffusion makes the blurred halftones from our method look much closer to the original, but the sharpening also enhances the edges, making the parrot appear separated from the background. The linear mobility function image (D, in the figure) produces good results, though with some visible patterns in the background. The logistics mobility function image (E) seems darker, owing to the lack of white dots inside the beak. Lastly, the extreme mobility function image (F) provides somewhat better shading than (D) and better contrast than (E). The HVS norms of these images are summarized in table 1. While the PDE methods perform poorly in the HVS norm, being an order of magnitude larger than the Direct Binary Search halftone, the figure shows that they can still be pleasing approximations of the original image.

Another possibility is to use Floyd-Steinberg Error Diffusion to produce the actual halftone, but use (11) as a preprocessor. By first blurring, then deblurring the image, certain features are enhanced, and errors in slow-changing background detail can be reduced. Figure 5 shows a test image that is blurred, sharpened by backwards diffusion using mobility (18), then halftoned via FS, as well as the image fed directly through FS. Artifacts visible in the lower left corner of the unprocessed halftone are not visible in the preprocessed halftone. The processing results in a crisper halftone, but again suffers from seeming to flatten the image through edge enhancement. The HVS norm, due to its blurring, is insensitive to the maze-artifacts, but reacts to the edge enhancement from the sharpening. The direct FS

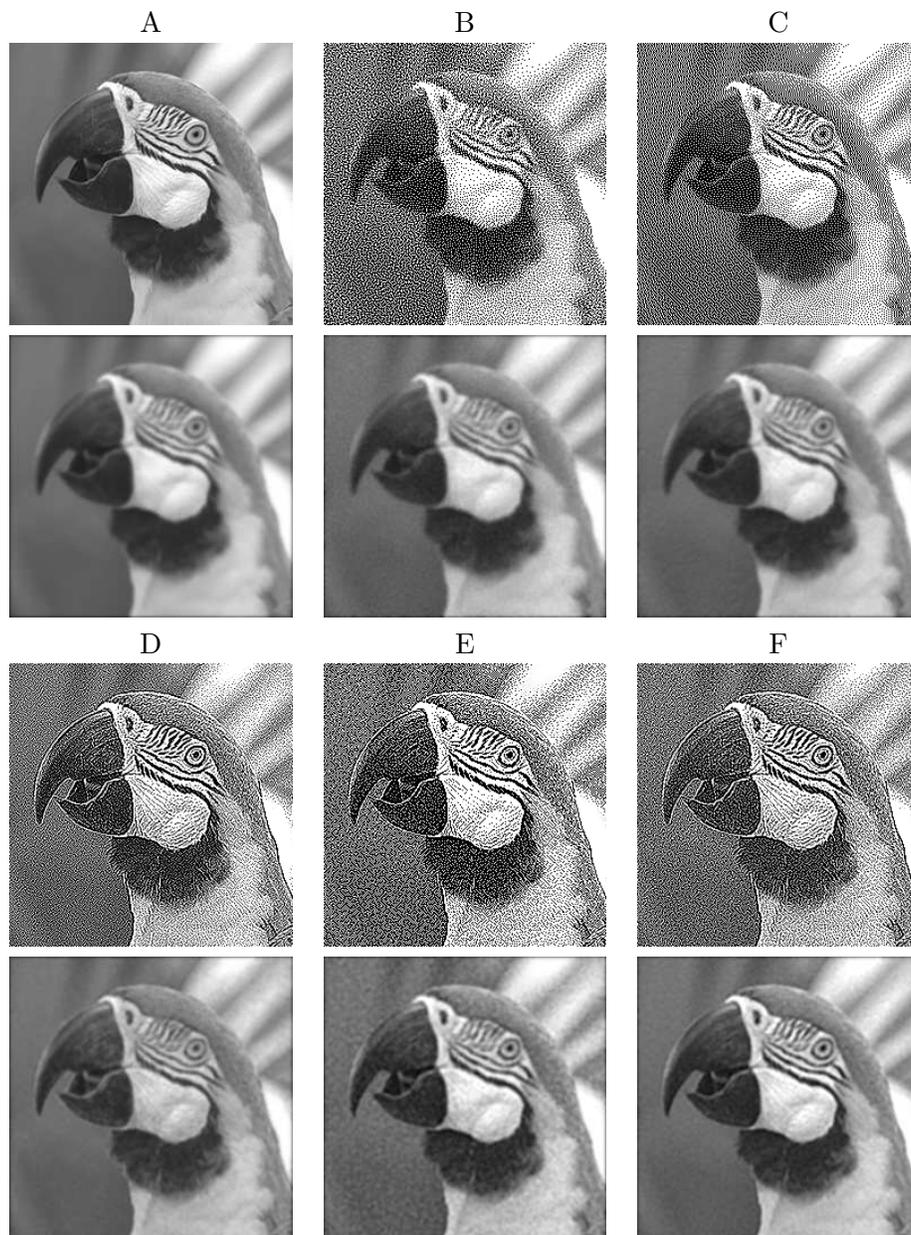


Figure 4: A test image (A) halftoned by several different algorithms. Under each image is  $B(u)$ , the HVS blur of the halftone for comparison. The methods are Direct Binary Search (B), Floyd-Steinberg error diffusion (C), backward diffusion with a linear mobility (D), logistic mobility (E), and extreme mobility (F) functions.

Halftoning Method	$\ h\ _u^2$	Relative to DBS
Direct Binary Search	6.950	1.000
Floyd-Steinberg	7.798	1.059
PDE, $M(u) = 1$	87.177	3.542
PDE, $M(u) = 4u(1 - u)$	63.729	3.029
PDE, $M(u) = 1 + (u - 1/2)^2$	72.362	3.227

Table 1: HVS norm performance of the PDE methods versus DBS and FS. DBS, being a search for the halftone that minimizes the norm, has the smallest value. The results in column three are shown relative to this value. The relative norms for the PDE methods are three times as large as for the standard methods. However, performance in the HVS norm is not the only criterion for a good halftoning scheme.

image has a normalized HVS of 0.0104 while the preprocessed image has a normalized HVS of 0.0298, nearly three times larger.

## 5 Discussion

We have discussed how to apply ideas from partial differential equations to image halftoning. Understanding the HVS blurring operator  $B$  as several applications of a discretized diffusion equation allowed us to approximate its inverse through backward diffusion. The principle problems with backward diffusion, namely the generation of noise and the non-positivity of the solution are beneficial effects in this application. Noise generation provides texture to regions of constant gray, allowing for better halftoning. The lack of a maximum or minimum principle can be cured by imposing forward diffusion on bad values, as with (16), or by a limiter like (17). The later choice frees us to choose a mobility function that acts more rapidly, such as (19). However, if intensity preservation is not important, unphysical values may simply be rounded during the halftoning step.

The proper choice of  $M$  is difficult. Our results show that the most basic choice – linear backward diffusion – performs very well, though a more thorough understanding of the kernel for (7) with different mobilities may shed light on a systematic approach to choosing this function.

This iterative scheme is not fast, however. Simple one-pass error diffusion is  $K$  times faster than our algorithm, and for large problem sizes, this limits the usefulness of our method. This scheme does grapple with minimizing the HVS norm through a novel means. Better understanding of this norm is essential to produce algorithms that produce high-quality results like Direct Binary Search without the tedious computation.

## References

- [1] Luis Alvarez and Luis Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal of Numerical Analysis*, 31(2):590–605, 1994.



Figure 5: Top left: the test image. Top right: the test image, blurred, then sharpened with  $M(u) = 1$ . Bottom left: halftone generated by applying FS directly to the original image. Bottom right: halftone generated by applying Floyd-Steinberg to the blurred-sharpened image. This image lacks the maze-like artifacts visible in the bottom left corner of the direct FS halftone.

- [2] Lawrence C Evans. *Partial Differential Equations*. Graduate Studies in Mathematics. American Mathematical Society, 1998.
- [3] RW Floyd and L Steinberg. Adaptive algorithm for spatial grey scale. *SID Int. Sym. Digest of Tech. Papers*, pages 36–37, 1975.
- [4] Guy Gilboa, Nir Sochen, and Yehoshua Y Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Transactions on Image Processing*, 11(7):689–703, 2002.
- [5] John B. Greer and Andrea L. Bertozzi. H-1 solutions of a class of fourth order non-linear equations for image processing. *Discrete and Continuous Dynamical Systems*, 10(1&2):349–366, 2004.
- [6] Thomas Laurent. Local and global existence for an aggregation equation. *Comm. Part. Diff. Eq.*, 2006. (To appear).
- [7] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

- [8] Guillermo Sapiro. *Nonlinear Image Processing*, chapter 8: Nonlinear Partial Differential Equations in Image Processing, pages 225–248. Academic Press, 2001.
- [9] Robert Ulichney. *Digital Halftoning*. MIT Press, 1987.