Error-Diffusion with Blue-Noise Properties for Midtones

Pierre-Marc Jodoin and Victor Ostromoukhov

Department of Computer Science and Operational Research
University of Montreal
Montreal, Canada

ABSTRACT

In this contribution, a new error-diffusion algorithm is presented, which is specially suited for intensity levels close to 0.5. The algorithm is based on the variable-coefficient approach presented at SIGGRAPH 2001.[9] The main difference with respect to the latter consists of the objective function that is used in the optimization process. We consider visual artifacts to be anomalies (holes or extra black pixels) in an almost regular structure such as a chessboard. Our goal is to achieve "blue-noise" spectral characteristics in the distribution of such anomalies. Special attention is paid to the shape of the anomalies, in order to avoid very common artifacts. The algorithm produces fairly good results for visualization on displays where the dot gain of individual pixels is not large.

Keywords: halftoning, blue-noise, artifacts, midtones, variable-coefficient approach.

1. INTRODUCTION

The main objective of this work can be divided into two different but complementary goals. The first goal is to understand the true nature of common artifacts located around intensity level 0.5. We will survey the most widespread artifacts and try to understand how they appear and what their relationship is to each other. Secondly, we will draw up a list of the artifact features we would like to see in the midtone area. Once that list is known, we will present a new error-diffusion approach that facilitates the manipulation of these features.

2. PREVIOUS WORK

Ever since the year when Robert W. Floyd and Louis Steinberg published the first error-diffusion algorithm,[2] many researchers have tried to find better error-diffusion coefficient sets[3, 11, 12] to enhance the visual quality of halftone images. Their goal was to take advantage of the simplicity of the error-diffusion algorithm while minimizing visual artifacts. Even if the results of some coefficient sets succeeded in producing images with few artifacts, none of these approaches eliminated all anomalies, nor did they shed light on the true nature of these artifacts. In this contribution, we will try to understand "under the hood" how error-diffusion works in the midtones and what the basic structure is of the halftone images produced under these intensities.

The error-diffision method that we will introduce takes advantage of more than one coefficient set. This idea was first introduced by Eschbach[17] who suggested to apply two different coefficient sets over the full intensity range. One set was for highlights and dark areas and one was for the other intensity levels. This excellent idea was pushed forward by Ostromoukhov[9] who suggested to use one coefficient set for each level. The reason for making this choice was to minimize the discontinuities induced by Eschbach method. Recently, another contribution was made in that direction by Li and Allebach.[4] Like in Eshbach's and Ostromoukhov's papers, Li and Allebach used a multiple coefficient approach. They explored a downhill search method along with a DBS[18] based cost function to automatically calculate the coefficient weights. They also proposed a varying threshold depending on the input pixel value.

For the remainder of this article, the behavior of error-diffusion in the midtone area using some of the most widespread coefficient sets will be studied in sections 2.1 and 2.2. Section 3 identifies a list of all the features we want to see around level 0.5, which will then help us elaborate a modified error-diffusion algorithm in section 4. The results will be presented in section 5, followed by a short conclusion.

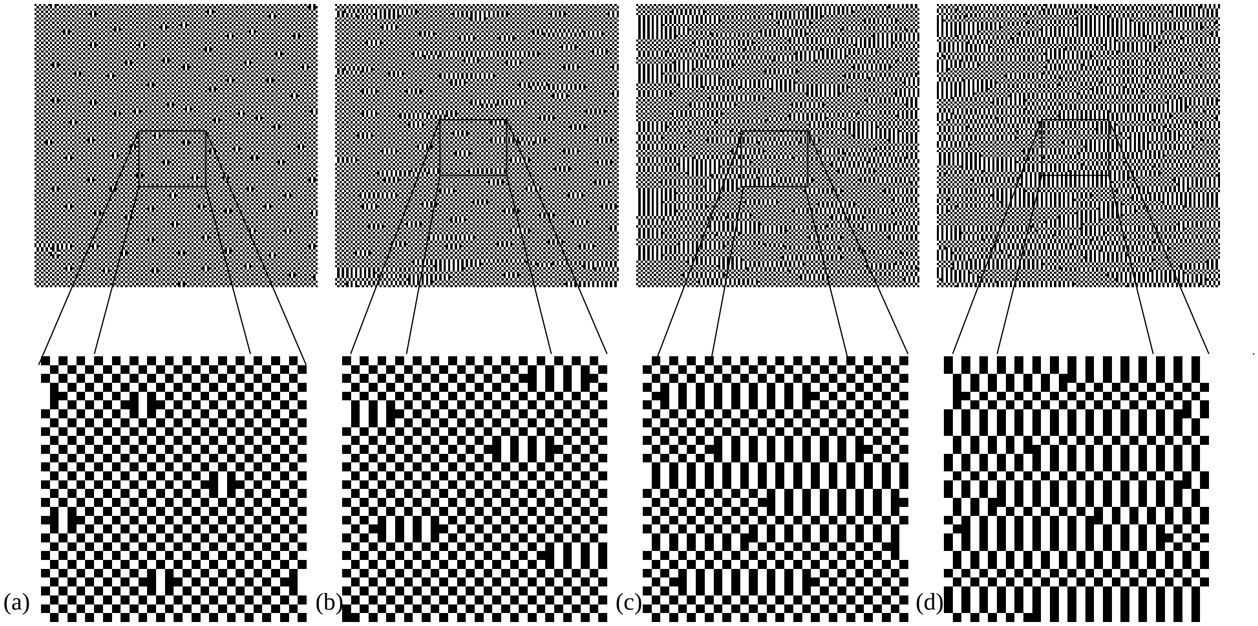


Figure 1: Common artifacts around intensity level 0.5.

2.1. Common Error-Diffusion Artifacts Around Level 0.5

To improve error-diffusion results in the midtones, we first need to identify what is deficient in this area. We isolated many typical artifacts visible in this region and closely observed their structure. Figure 1 presents four common artifacts around level 0.5 computed using conventional coefficient sets.[1, 9, 11] Even if they look different, they all share a common denominator. As shown in Figure 2, these four artifacts have the same basic structure consisting of an *entry point*, a *body* and a *closing point*. The only difference between them is their body length. We determined that the body length is related to the first coefficient d_{10}[*]. The bigger d_{10} is, the longer the artifacts are. Moreover, this behavior is pretty much the same for all the coefficient sets we have studied (see Figure 3).

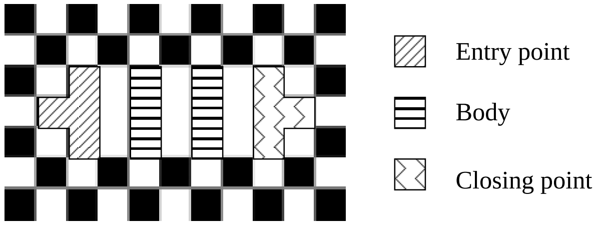


Figure 2. Most artifacts around 0.5 have the same basic structure: they start with an entry point followed by a body and end with a closing point. This artifact has length five because its body contains five pixels.

[*] The coefficient d_{10} is the first one on the right of the current pixel. It is $\frac{7}{48}$ for Jarvis *et al*, $\frac{8}{16}$ for Shiau-Fan and $\frac{7}{16}$ for Floyd-Steinberg.

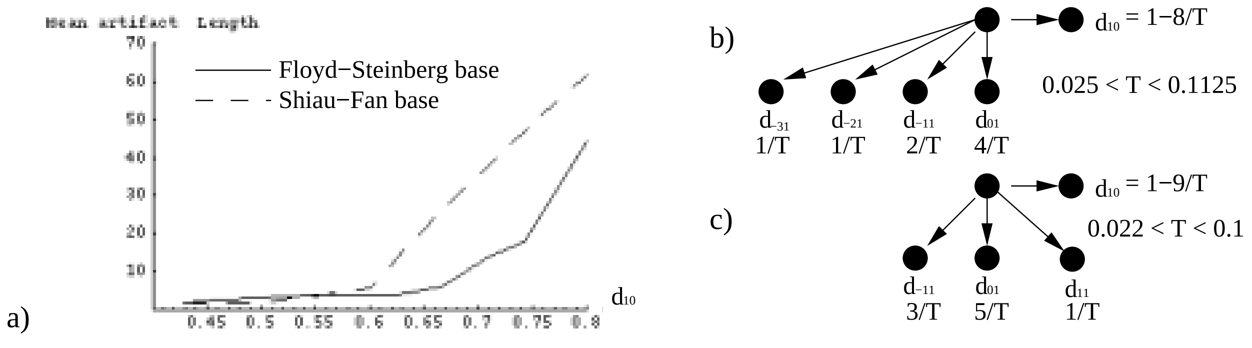


Figure 3. Plot (a) shows the relationship between the weight of the first coefficient d_{10} and the length of artifacts around 0.5. The dashed-line curve was computed using Shiau-Fan's coefficient set (a) while the solid curve was computed using Floyd-Steinberg's coefficient set (b). In each case, we kept constant all coefficients except the first one (d_{10}), which varied from 0.1 to 0.8. For every d_{10} value, we launched the error- diffusion algorithm on a flat 126 image and measured the mean artifact length.

2.2. Common Drawbacks of Artifacts

Evaluation of the most widespread artifacts around 0.5 underlined a number of undesired patterns. First, the artifacts with small length (Figure 1(a)) are visually unpleasant at both low and high resolution. They appear to any observer like a field of big holes. Long artifacts (Figure 1(c) and 1(d)) are shocking in low resolution since they give a chaotic look to the results. Moreover, these artifacts are not located over a uniform background. Observe that the uniform chessboard background is broken by the long overlapping artifacts. The fact that the background is not independent of the length of the artifacts is annoying for two reasons. The uniform background (Figure 1(a)) is more pleasing than a non-uniform one (Figure 1(b)) and the inability to work independently on the background and the artifacts may lead to a complex solution.

Another drawback related to conventional error-diffusion is the fact that artifacts do not have the same length when applied on different intensity levels as shown in Figure 4. Under such conditions, artifacts visual appearance and distribution in space are hard to handle. The last limitation comes from a simple observation: artifacts are not generally distributed in a "blue-noise" manner.

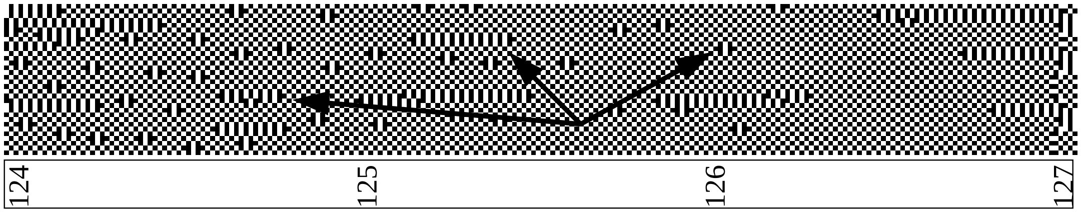


Figure 4. Image of Floyd-Steinberg artifacts around 0.5. Artifacts are not consistent through intensity levels since some have length 1, 7, 11 and more.

3. DESIRED ARTIFACT FEATURES

Starting from the previous section's observations, we will define the type of desired artifacts and background features we want to see in the midtone area. These features will then indicate a path for designing our algorithm.

First, we want to have discreet point artifacts over a constant chessboard background (see Figure 5). Such a structure is both visually pleasing and compact. For the reasons we evoked earlier, there shall be no relationship between the background and the artifacts. This will ensure that our algorithm keeps the basic error-diffusion design (Figure 6).

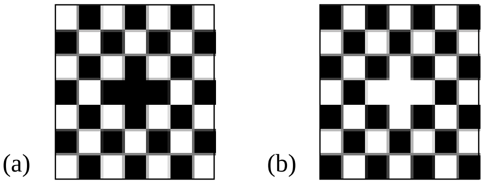


Figure 5. Image of the “ideal” artifact. It is a point over a uniform chessboard background, for intensity levels below 0.5 (a) and above 0.5 (b).

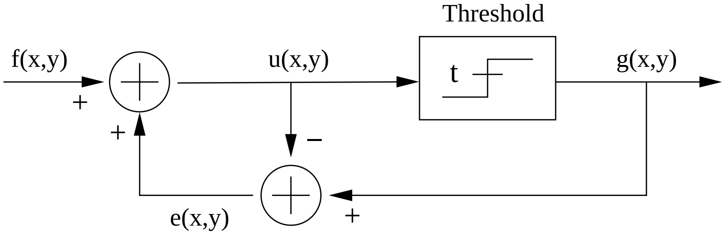


Figure 6: The error-diffusion circuit,[5,7] also known as a *noise-shaping feedback coder*.

It has been shown that a pleasing pixel distribution in space has to have blue-noise spectral characteristics.[10,13,15] Because the position and distribution in space of these simple artifacts can be easily manipulated, we intend to extend the blue-noise concept to midtone point artifacts. It is important to mention that we shall make no major modifications to the original error-diffusion algorithm. We want to maintain its simplicity, stability and speed.

4. THE METHOD

4.1. The General Algorithm

It is obvious that many error-diffusion coefficient sets generate similar artifacts around intensity level 0.5. We judge that simply changing the coefficient weights in one way or another does not modify the very basic artifact structure in that area. Therefore, even if we wish to keep the fundamental error-diffusion design, we have no choice but to slightly improve this algorithm in order to reach the “ideal” artifact features previouly mentioned.

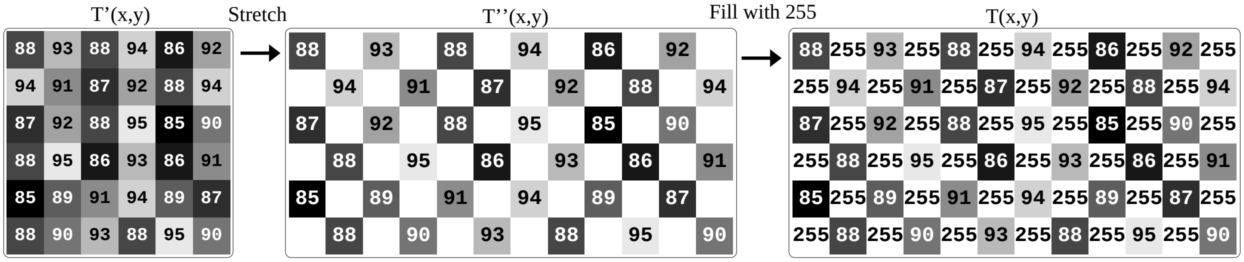
As stated earlier, there is a desire to have a chessboard background. For this reason, the first and major modification to the original error-diffusion algorithm is to force one pixel out of two to be black. To do so, we will reuse the error-diffusion basic design (see Figure 6) with a modified *threshold* function. Let $f(x, y)$ be the input continuous-tone image (where each pixel intensity is a level between 0 and 255), $u(x, y)$ the updated image, $g(x, y)$ the output halftone image and t the fixed threshold. If we consider $T(x, y)$ to be a black and white chessboard *threshold matrix,* our algorithm can be formalized as:

$$g(x,y) = \begin{cases} 0, & \text{if } T(x,y) \text{ is black OR } u(x,y) < t \\ \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

where

$$u(x+i, y+j) = f(x+i, y+j) - w(i,j)(g(x,y) - u(x,y)). \tag{2}$$

and $w(i, j)$ is the error-weighting matrix satisfying constraint $\sum_{i,j} w(i, j) = 1.0$. It is important to note that the algorithm presented here works for intensity levels below 0.5. Nevertheless, its behavior is symmetric and can be directly transposed to levels above 0.5.



88	93	88	94	86	92
94	91	87	92	88	94
87	92	88	95	85	90
88	95	86	93	86	91
85	89	91	94	89	87
88	90	93	88	95	90

88		93		88		94		86		92	
	94		91		87		92		88		94
87		92		88		95		85		90	
	88		95		86		93		86		91
85		89		91		94		89		87	
	88		90		93		88		95		90

88	255	93	255	88	255	94	255	86	255	92	255
255	94	255	91	255	87	255	92	255	88	255	94
87	255	92	255	88	255	95	255	85	255	90	255
255	88	255	95	255	86	255	93	255	86	255	91
85	255	89	255	91	255	94	255	89	255	87	255
255	88	255	90	255	93	255	88	255	95	255	90

Figure 7. Zoomed view of the threshold array $T(x,y)$ used by our algorithm to relax the background constraint between intensity level 96 and 85. Using the void-and-cluster method, we computed the dither array $T'(x,y)$. We then stretched it to get the second array $T''(x,y)$. $T(x,y)$ was obtained by filling $T''(x,y)$'s empty spots with the value 255.

This simple algorithm gives good results for intensity levels between 96 and 127. Unfortunately, results below 96 are unpleasant and far from being remarkable. This underlines the fact that the background constraint should only be applied to intensity levels between 96 and 127. For lower intensities, we will remain with the regular error-diffusion algorithm. We now need to gradualy switch between the "forced background" levels (96–127) to the other ones (0–96). We have empirically chosen the 85–96 area to relax the background constraint. For a smooth transition, we replaced the black pixels of T(x,y) with pixels having a level between 85 and 96. With this modification, our algorithm now becomes:

$$g(x,y) = \begin{cases} 0, & \text{if } f(x,y) > T(x,y) \text{ OR } u(x,y) < t \\ & \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

In order to obtain the smoothest transition possible, we need to uniformly distribute the pixel levels in the threshold array $T(x,y)$. To do so, we used a slightly modified version of Ulichney's void-and-cluster method[14] that works this way:

1. Compute $T'(x,y)$ with levels going from 85 to 96 using the void-and-cluster method.
2. Compute array $T''(x,y)$ by stretching $T'(x,y)$.
3. Generate $T(x,y)$ by filling $T''(x,y)$'s empty spots with value 255. (see Figure 7).

A result showing the effect of using that threshold array is presented in Figure 8.

4.2. Blue-Noise Distribution of Artifacts

We now have an algorithm that forces point artifacts over a chessboard background between level 127 and 96 and smoothly relaxes that constraint between 96 and 85. A feature that has not been taken into account until now is the artifacts blue-noise distribution in space. To do so, Ostromoukhov's[9] error-diffusion algorithm was used. This algorithm is essentially the same as the original error-diffusion method but instead of using one error-weighting matrix for all intensities, it uses 255 matrices: one for each level. The reason for doing so is that each matrix gives good results for some intensity level while giving poor ones at other levels. All that is needed now is to find a matrix coefficient set for each level between 127 and 96 that would distribute the artifacts in a blue- noise manner.

In the present case, we will reuse Ostromoukhov's coefficient sets for levels between 0 and 96 but find new ones for levels 96 to 127. As it was mentioned earlier, because our algorithm is symmetric, all coefficients used between 0 and 127 will be transposed to levels 255 down to 128.

To find a matrix coefficient set for a given intensity level I, a simplex downhill method[16] along with a blue- noise based cost function was used. As shown in Figure 9, this cost function calculates the area between a given analytical function $F(f)$ and the radial Fourier transform $RFT(f)$. It was found that a Gaussian profile for $F(f)$ was the best compromise between simplicity and result quality. We empirically observed that a standard deviation of $\frac{f_g}{10}$ was good for all intensities between 96 and 127.

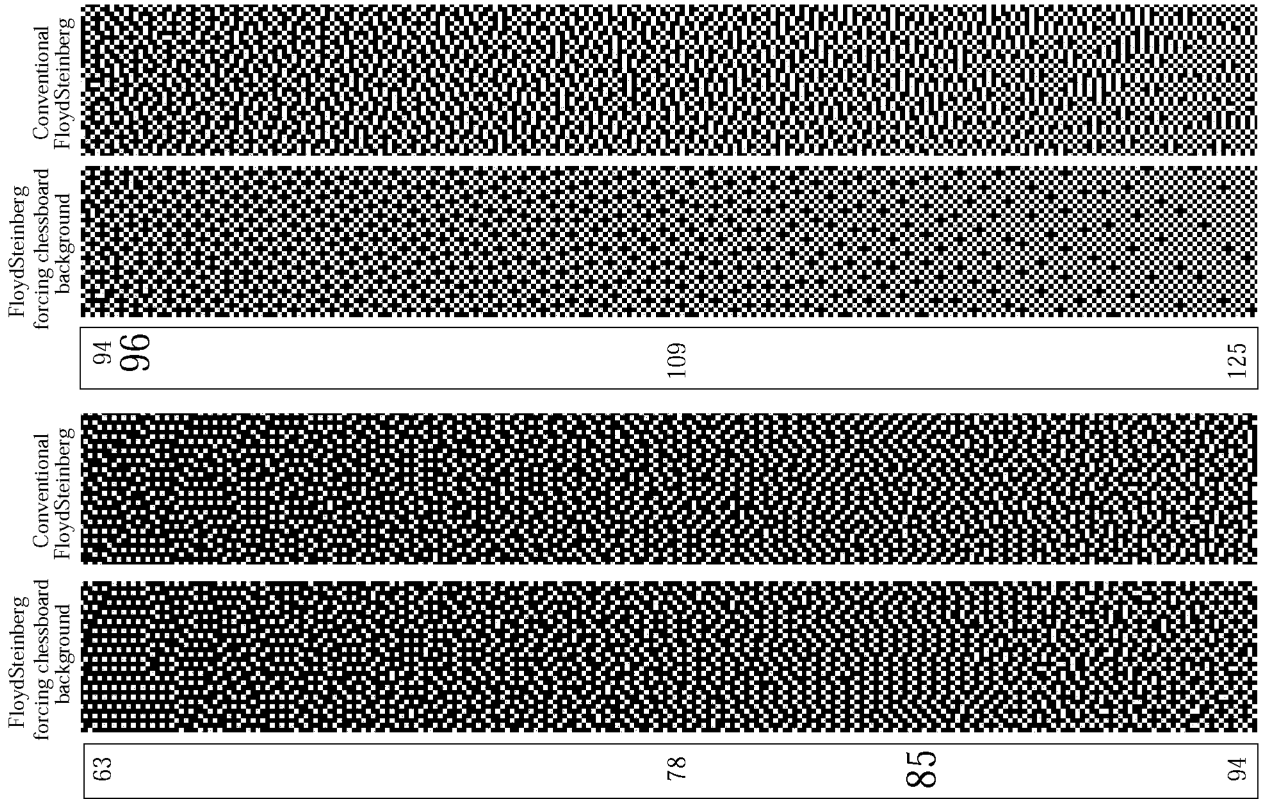

Conventional FloydSteinberg
FloydSteinberg forcing chessboard background
94
96
109
125
Conventional FloydSteinberg
FloydSteinberg forcing chessboard background
63
78
85
94

Figure 8. Here we can see the dual effect of forcing the chessboard background and using the threshold array $T(x, y)$. For levels over 96, a chessboard background can be seen while below 85 no constraint was applied. Between 96 and 85, the threshold array gradually relaxes the background constraint.

That way, the simplex method converges to the coefficient set that generates the best blue-noise artifact distribution. Similar methods have also been used in the past to compute optimal values for the weights.[4, 9] The cost function $c(I, CS)$ works this way:

1. With the coefficient set *CS*, generate a flat halftone image **A** of intensity *I*.
2. Filter image **A** to get rid of the chessboard background and keep the point artifacts. The result is put in image **B**.
3. Find $RTF(f)$, the *radial* Fourier transform of image **B**.
4. Compute $C(f) = ||RTF(f) - F(f)||$.
5. Integrate $C(f)$ between 0 and f_g and return the resulting value (see Figure 9).

Note that the coefficient sets CS are always initialized with random values.

5. RESULTS

We created three gradient images of size 512 x 64 pixels, 1024 x 64 pixels and 2048 x 64 pixels, and applied our algorithm to it. The first one is shown in Figure 10 and had intensity levels going from 127 down to 100 in order to explicitly show the contribution of our algorithm. We can indeed see the difference in the stability and the visual quality of our algorithm over Floyd-Steinberg and Shiau-Fan. Using our fully controled approach, we obtained coherent and visually pleasing results for all levels between 127 and 96.

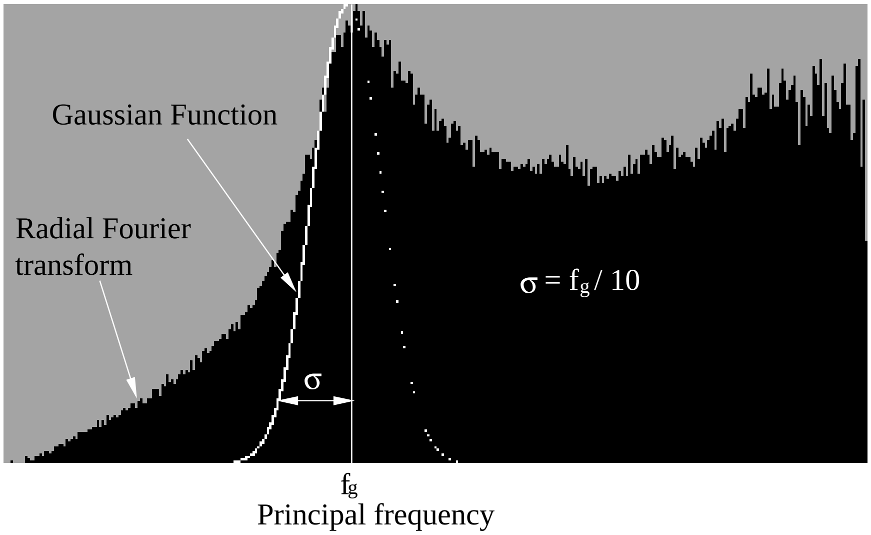


Figure 9. This image shows how the cost function $c(I, CS)$ calculates the error value. It first computes the halftone image of a flat intensity image "I". It then gets its radial Fourier transform while calculating a Gaussian function. That function is centered on principal frequency ${f_g}$[7, 15] and has standard deviation $\sigma = \frac{f_g}{10}$. The error value is the area between the Gaussian function $F(f)$ and the radial Fourier transform $RFT(f)$: $error = \int_{f=0}^{f_g} ||RFT(f) - F(f)|| df$

The second image illustrates the transition between the region where we explicitly force a chessboard background (96–127) and the region where we directly use the basic error-diffusion algorithm without any constraint (0–85). In fact, in Figure 8 (a) we can see the result of our algorithm using the Floyd-Steinberg coefficient set $(7/16, 1/16, 3/16, 5/16)$ over that gradient image. Between level 125 and 63, three *regions* are visible: the one where we force the chessboard background (96–125), the one where no constraint is applied (63–85) and the one where the constraint is relaxed (85–96).

The last image is a 0–127 gray ramp and is shown in Figure 11.

The time used by the simplex algorithm to find the best coefficient sets is not the same for all intensity levels but we can say that it took roughly an average of 30 seconds for a 1.3 Ghz pentium III to do the job.

6. CONCLUSION

In this article, the objective was to better understand error-diffusion midtone artifacts and find a solution to control these anomalies. It was observed that for any coefficient sets, error-diffusion produces artifacts having the same basic structure in the midtones. A slight modification to the error-diffusion algorithm was proposed in order to force simpler artifacts over a constant chessboard background. This artifact constraint is applied in the midtones (between 96 and 127) and is gradually relaxed between 96 and 85 using a threshold matrix. Our algorithm also uses Ostromoukhov's variable-coefficient approach to make sure the distribution of midtone point artifacts follows a blue-noise spectral profile. The coefficient sets for intensity levels between 96 and 127 were computed using a simplex downhill method over a blue-noise based cost function $c(CS, I)$.

The presented method gives good results for low resolution devices but the traditional error-diffusion algorithm can give more discreet artifacts on higher resolutions. Although the chessboard background is more discreet and visually pleasing on digital displays, it is not appropriate for printers because of the dot gain effect.[6]

7. FUTURE WORK

We have demonstrated control of error-diffusion between intensity levels 96 and 127. In the future, we wish to understand and fully control error-diffusion at other intensities. In addition, we are working on an improved threshold matrix T(x,y) in order to soften further the transition between 85 and 96.

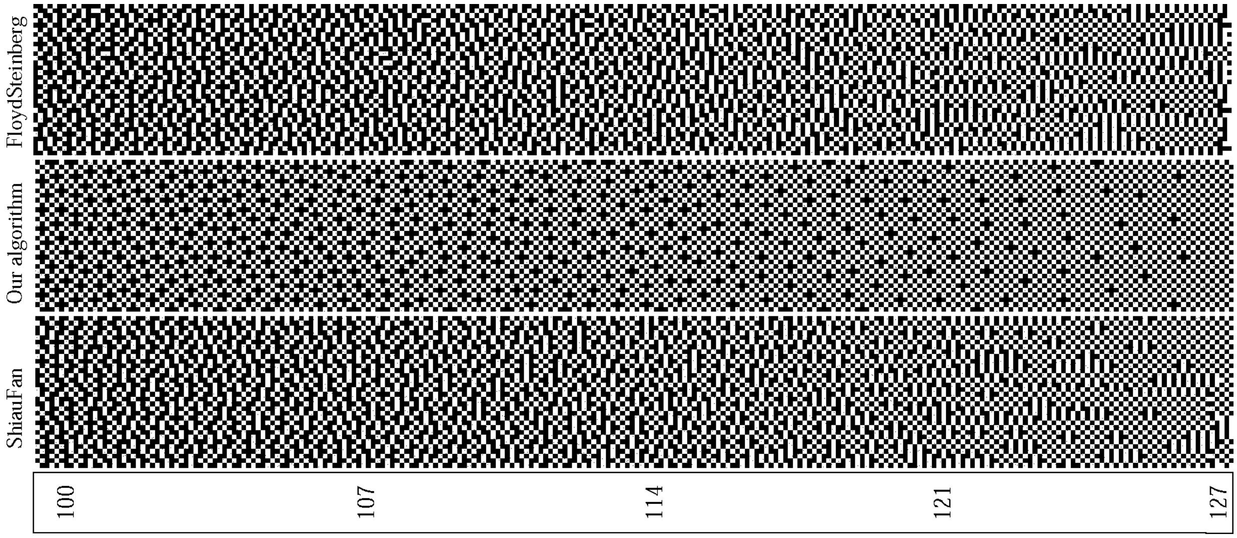


Figure 10: Comparison of Floyd-Steinberg, Shiau-Fan and our method over a midtone gray ramp.

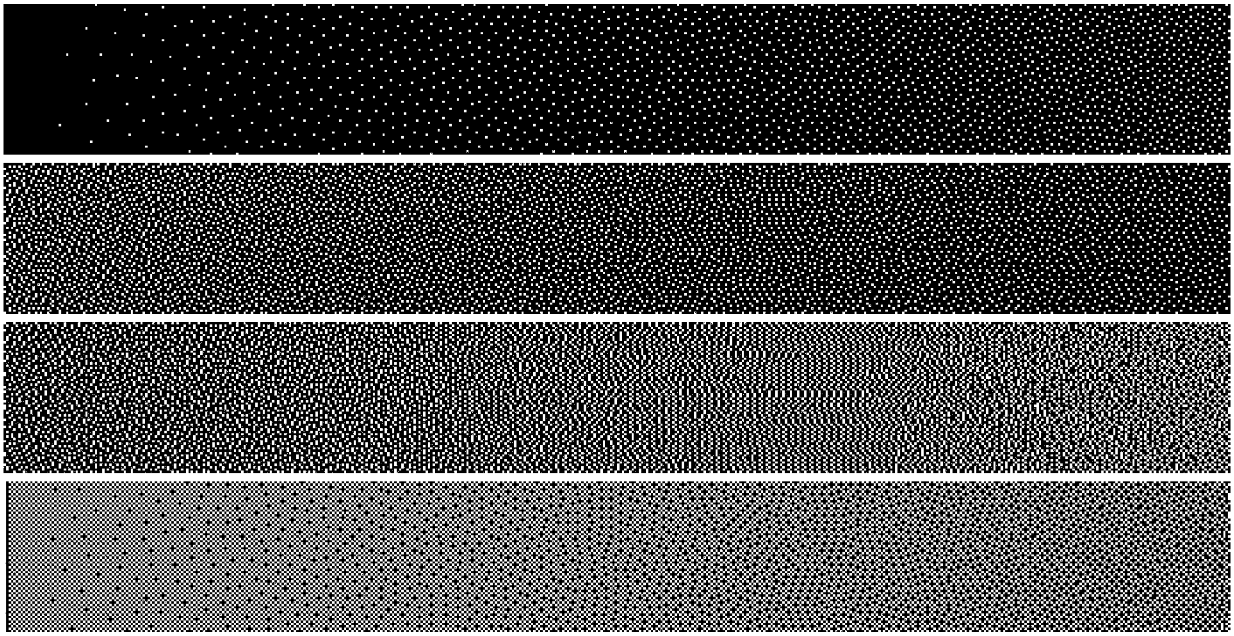


Figure 11: Our algorithm over a 0-127 gray-scale ramp.

8. ACKNOWLEDGMENT

The authors would like to thank Jack Sweeney and Justin Bur for their help.

REFERENCES

1. R. Floyd and L. Steinberg, *An adaptative Algorithm for Spatial Grey-Scale*, Proc. of the SID, Vol 17, No.2, pp. 75-77, 1976.
2. R. Floyd and L. Steinberg, *An adaptive Algorithm for Spatial Gray Scale*, Society for Information Display Symposium Digest of Technical Papers, pp. 36-37, 1975.
3. J.Jarvis, C.Judice and W.Ninke D.Snow *A Servey of Techniques for the Display of Continuous Tone Pictures on Bi-level Displays*, Computer and Image Processing, Vol.5 pp.12-40, 1976
4. P.Li and J.Allebach *Tone Dependent Error Diffusion*, IEEE Trans. on Img. Proc., July 28, 2001
5. T. D. Kite, B. L. Evans, and A. C. Bovik, *Modeling and Quality Assessment of Halftoning by Error Diffusion*, IEEE Transactions on Image Processing, vol. 9, no. 5, pp. 909-922, May 2000.
6. K. T. Knox, *Introduction to digital halftones*, IS&T's 47th Annual Conference, ICPS'94: The Physics and Chemistry of Imaging Systems, May 1994, vol. 2, pp. 456–459.
7. K. T. Knox, *Error Image in Error Diffusion*, Proc. SPIE Conf. Electronic Imaging, (San Jose, CA), Feb.9-14, 1992.
8. K.T.Knox, *Evolution of Error Diffusion*, Journal of Elec. Img.,vol 8(4),1999,pp422-429.
9. V.Ostromoukhov *A Simple and Efficient Error-Diffusion Algorithm*, Proc. SIGGRAPH '01, pp. 567, 2001
10. T. Mitsa and K. J. Parker, *Digital halftoning using a blue noise mask*, in ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, vol. 2, (Toronto, Canada), pp. 2809–2812, IEEE, May 1991.
11. J.Shiau, Z.Fan *A Set of Easily implementable Coefficients in Error Diffusion With Reduced Worm Artifacts*, Proc. SPIE Vol.2658, pp. 222-225, 1996.
12. P. Stucki *MECCA-a Multiple-Error Correction Computation Algorithm for Bilevel Image Hardcopy Reproduction*, Research Report RZ1060, IBM Res. Lab., Zurich, Switzerland, 1981.
13. R.A.Ulichney *Digital Halftoning*, The MIT Press, Cambridge, 1987.
14. R.A.Ulichney *The void-and-cluster method for dither array generation*, IS&T/SPIE Symposium on Electronic Imaging Science& Tech.,San Jose, Ca, val 1913, 1993. pp.9-18.
15. R.A.Ulichney *Dithering With Blue Noise*, Proc. IEEE, vol. 76, no. 1, pp. 56-79, 1988
16. W. T. Vertterling et al. *Numerical Recipes: Example Book (C)*. Cambridge University Press, 1988.
17. R. Eschbach *Reduction of artifacts in error diffusion by mean of input-dependent weights*, Journal of Elec. Img, 2(4):352 358, 1993.
18. M. Analoui and J. P. Allebach, *Model-based halftoning using direct binary search*, Proc. of SPIE/IS&T Symp. on Electronic Imaging Science and Tech., February 1992, San Jose, CA, pp. 109 – 121.