A Simple and Efficient Error-Diffusion Algorithm

Victor Ostromoukhov

Université de Montréal [*]

Abstract

In this contribution, we introduce a new error-diffusion scheme that produces higher quality results. The algorithm is faster than the universally used Floyd-Steinberg algorithm, while maintaining its original simplicity. The efficiency of our algorithm is based on a deliberately restricted choice of the distribution coefficients. Its pleasing nearly artifact-free behavior is due to the off-line minimization process applied to the basic algorithm's parameters (distribution coefficients). This minimization brings the Fourier spectra of the selected key intensity levels as close as possible to the corresponding "blue noise" spectra. The continuity of the algorithm's behavior across the full range of intensity levels is achieved thanks to smooth interpolation between the distribution coefficients corresponding to key levels. This algorithm is applicable in a wide range of computer graphics applications, where a color quantization algorithm with good visual properties is needed.

Keywords: Halftoning, Error-Diffusion, Image Quality, Color Quantization

1 Introduction

Since its introduction in 1975 by Floyd and Steinberg [7], the error-diffusion (E-D) algorithm has attracted much attention in the graphics community (see for example [10, 13, 25]. The main advantage of the initial version of the algorithm is its simplicity combined with the fairly good overall visual quality of the produced binary images. In addition, it is a public domain algorithm. For all these reasons, it became very popular in various graphics applications where color quantization is combined with spatial distribution, such as display visualization using a limited color palette.

Let us recall the initial Floyd-Steinberg E-D algorithm (see Fig 1). Pixels are processed according to a processing path, which is a simple sequence of scan-lines in the original version of the algorithm. The input signal intensity level is compared to a fixed threshold, and the output binary signal is generated according to this comparison. The difference between the input and the output signals (quantization error) is distributed to the 8-connected neighbors which have not yet been processed, according to distribution coefficients (7/16, 3/16, 5/16 and 1/16 in the original version).

Still, the Floyd-Steinberg E-D algorithm contains a number of inherent drawbacks. First, this algorithm produces clearly identifiable visually harmful artifacts in highlights and in dark areas, sometimes referred to as worm artifacts (see in [4, 21, 23], also detailed analysis in [13]). Second, at certain intensity levels, close to 1/2, 1/3, 2/3, 1/4, 3/4 etc. patches of regular structure may appear. Not only are such patches visually disturbing, but also the uneven transition between "structured" and "unstructured" areas may be clearly visible and undesirable (see Fig 4). This phenomenon can be greatly amplified by the dot gain when the visualization media is printed paper. In some extreme cases analyzed in [19], it may be questionable whether Floyd-Steinberg E-D algorithm could ever be used for printing purposes. Computer displays have much smaller inter-pixel dependency. Nevertheless, harmful visual perturbations due to uneven patches of regular structures should be avoided whenever possible.

In order to cope with the major drawbacks mentioned above, various solutions have been proposed. Let us enumerate some simple improvements to basic E-D algorithm reported in the literature:

- modification of the processing path: usage of serpentine path instead of scanlines [23], or space-filling curves [28, 26, 27]. Usually, this improves the behavior for some intensity levels, and worsens it for others. Nevertheless, the serpentine path became a standard. The algorithm described in [27] shows the best results in this category; still, it shows some visually perturbing artifacts (see Fig 3).

- modification of the distribution range and distribution coefficients: distribution to second-level unprocessed neighbors (neighbors of neighbors) [9, 22] or use of different distribution coefficients [21]. Eschbach suggested applying two distinct sets of distribution coefficients, one set for highlights and dark areas, and another set for the rest [4]. It was an excellent idea, but unfortunately a visually disturbing discontinuity appears on the boundary between two areas where two distinct sets of coefficients are applied. Clearly, we build our algorithm upon Eschbach's idea, solving the problem of discontinuity. We propose a simple and straightforward method for calculating the distribution coefficients that provide satisfactory visual quality.

- variation of the threshold: usage of an input-dependent threshold instead of a constant one. Eschbach and Knox [6] propose inversely-proportional dependence. This permits the control of inherent edge enhancement, but has little impact on the structure-artifact effect.

None of the proposed methods in this family of simple improvements can be considered as fully satisfactory. Usually, they show their drawbacks in a true torturous test that contains patches of all intensity levels, big enough to show the artifact structure proper to each intensity level (see Fig 4 bottom).

The best visual quality, in our personal evaluation, has been achieved using the algorithm proposed by Shiau and Fan [21]. Alas, like many other proprietary algorithms, it is covered by a patent [20]. For various reasons, a considerable part of research in halftoning has been done as part of industrial research, and its usage is restricted by patents and intellectual property laws.

In the family of sophisticated improvements, the output is sometimes better than that of the simple improvements enumerated

To appear in Proceedings of SIGGRAPH 2001

[*] http://www.iro.umontreal.ca/˜ostrom

above. Nevertheless, these algorithms usually suffer from two big drawbacks: they are too computationally costly to be implemented in simple and fast visualization software (e.g. see Eschbach's pulse-density modulation [5], Allebach's direct binary search-based halftoning [2, 14], or Marcu's roadmap [15]), and they are often proprietary, which considerably limits their potential application for the computer graphics community. The best implementation of the DBS-based halftoning is approximately 10 times slower than a typical error-diffusion, according to Jan Allebach's estimation [1].

Yet another solution consists in building threshold matrices such as Ulichney's "void-and-cluster" technique [24], or Mitsa and Parker's "blue noise mask" [16]. In both cases, a threshold matrix is built to be used with standard dithering, in such a way that its output mimics the behavior of a typical error-diffusion having "blue noise" spectrum. These solutions are inherently fast. Nevertheless, once again, they suffer from two drawbacks. First, the matrices are not publically available. Secondly, the output of either "void-and-cluster" or "blue noise mask" techniques appear less sharp, i.e. a bit blurry when compared with that produced with a typical E-D technique. The phenomenon of additional sharpness of E-D is due to its inherent edge enhancement, which is well studied in the literature [11, 12]. It has been shown that the effect of edge enhancement of E-D is very close to the effect of a simple Laplacian or a similar sharpening filter [6, 12]. Consequently, an equivalent edge enhancement in point-processes can be achieved by appropriate pre-filtering. Still, such a pre-filtering may be prohibitive for real-time visualization, due to its extra cost.

It is generally accepted that E-D algorithms achieve better quality at low resolution, compared to other fast halftoning methods: dispersed or clustered dithering, blue-noise-based point-processes without pre-filtering [10, 13, 23]. It is important to mention that practically all public domain quantizers do use the Floyd-Steinberg E-D algorithm.

2 Proposed error-diffusion algorithm

The algorithm we propose in this contribution is as simple as the original Floyd-Steinberg E-D algorithm; however it incorporates a few crucial differences: we use a serpentine processing path, we distribute errors to three instead of four neighbors (only to N_{10}, N_{-11} and N_{01} as shown in Fig 1), and we use 256 different distribution coefficient sets for each of the 256 input intensity levels (in this section we suppose, without loss of generality that the input one-channel signal is sampled on 8 bits, i.e. it has 256 distinct levels of intensity). The choice of three distribution coefficients instead of four was deliberate in order to reduce the number of dimensions for the optimization (see below), as well as to increase the algorithm's execution speed. Two coefficients would be even better; unfortunately, we did not find satisfactory solutions in the two-dimensional parametric space.

It is obvious that in this form the proposed algorithm conserves the simplicity and the efficiency of the original Floyd-Steinberg error-diffusion algorithm. In fact, it is even faster, thanks to a smaller number of arithmetical operations and memory accesses. The fact that it uses input-dependent distribution coefficients does not imply any conceptual or programming sophistication, because the distribution coefficients can be hardcoded in the program.

In the rest of this section we shall present the method that we used in order to determine 256 input-dependent distribution coefficient sets. We based our research on the following empirical assumptions:

A1 The E-D algorithm's output is considered to be good when its Fourier spectrum is close to the "blue noise" [23, 12]. An aperiodic structure is said to have the blue noise property when its Fourier spectrum is radially symmetric, and does not possess energy at low frequencies, as shown in Fig 2e. The characteristic frequency f_g of the blue noise is determined by the mean distance between black pixels of a patch for a given intensity level.

A2 If in our algorithm we have a set of distribution coefficients $D^1 = \{d^1_{10}, d^1_{-11}, d^1_{01}\}$ for input intensity level g^1, and another set $D^2 = \{d^2_{10}, d^2_{-11}, d^2_{01}\}$ for input intensity level g^2, which generate two distinct artifact structures, then for all intermediate intensity levels g^i between g^1 and g^2, the artifact structure varies very smoothly when the distribution coefficients $D^i = \{d^i_{10}, d^i_{-11}, d^i_{01}\}$ are calculated as a smooth interpolation between D^1 and D^2. The simplest case is a linear interpolation between D^1 and D^2. For any intensity level g^i, the sum of the distribution coefficients must be 1, in order to preserve the tone reproduction fidelity: $d^i_{10} + d^i_{-11} + d^i_{01} = 1$.

A3 There are intensity levels that are potentially problematic. The original Floyd-Steinberg E-D algorithm produces more or less strong artifacts at intensity levels 1/255, 64/255, 85/255, 127/255. We call them key levels.

A4 Artifact structures for input intensity levels g_j and g_{255-j} are identical, with black and white values inverted. This permits us to restrict our search to half-range [0..127], then extend it symmetrically around 127.5.

Based on these assumptions, we propose a method for finding all 256 sets of distribution coefficients $D^i = \{d^i_{10}, d^i_{-11}, d^i_{01}\}$:

Step 1. For each intensity level among potentially problematic ones (key levels in assumption A3) we look for a set of distribution coefficients $D^{key} = \{d^{key}_{10}, d^{key}_{-11}, d^{key}_{01}\}$ that produce the output having Fourier spectrum as close as possible to blue noise (assumption A1). This is a classical minimization problem [18]. We start with an arbitrary set of distribution coefficients, say $\{1/3,1/3,1/3\}$, then vary them by trying to minimize the total error term between the obtained Fourier spectrum and the known blue-noise spectrum. The cost function, used in the minimization, contains two components: (1) the correlation between the achieved Fourier spectrum, and the radially-symmetrical "ideal" blue-noise spectrum calculated for this intensity (see [23, 25]), and (2) a weighted sum of all Fourier picks in the range of low frequencies, which are responsible for low-frequency artifacts, such as alignments or worms. The proportion between these two components has been chosen experimentally. For calculations, we use relatively big (1024x1024) uniform patches of constant input intensity.

Step 2. Between two key intensity levels g^1 and g^2 having distribution coefficient sets D^1 and D^2, we apply a linear interpolation between corresponding distribution coefficients (assumption A2). The algorithm behaves smoothly between g^1 and g^2, but not necessarily well. For some intermediate input levels, it may produce visually unpleasant structure. If this is the case, we add this "bad" intermediate level to the initial set of key intensity levels, and we repeat the whole process starting from Step 1. The key intensity levels as used in this paper are highlighted in Fig 4.

Step 3. We extend the half-range [0..127] symmetrically around 127.5 (assumption A4): $D_0 = D_{255}; D_1 = D_{254}; ...; D_{127} = D_{128}$.

All calculations presented above are done off-line. Once all 256 distribution coefficients are found, they can be statically used in the proposed variable-coefficient E-D algorithm. Appendix I enumerates the set of proposed distribution coefficients. Fig 4 compares the output of the proposed E-D algorithm to the serpentine-path Floyd-Steinberg and Shiau-Fan E-D algorithms.

Extension of the proposed algorithm to color (separate color channels) and to multiple-level output are analogous to that of the Floyd-Steinberg E-D algorithm [10].

The proposed algorithm may be combined with the threshold modulation introduced in [6], in order to control inherent edge enhancement, when needed. With a constant threshold, our algorithm produces visually appealing output images, with the degree of sharpness appropriate for most of visualization tasks.

3 Conclusions and Future Work

We have presented a simple and efficient variable-coefficient error-diffusion algorithm. It solves important visual problems inherent to the original Floyd-Steinberg algorithm. At the same time, it is faster than most other error-diffusion algorithms, and it produces sharp, visually pleasing output.

The efficiency of our algorithm is based on a deliberately restricted choice of the distribution coefficients. Its attractive nearly artifact-free behavior is due to the off-line minimization process applied to the set of the basic algorithm's parameters (distribution coefficients). This minimization brings the Fourier spectra of the selected key intensity levels as close as possible to the corresponding "blue noise" spectra. The continuity of the algorithm's behavior across the full range of intensity levels is achieved thanks to the smooth interpolation between the key level distribution coefficients.

Although our algorithm produces fairly satisfactory output, we hope to improve these results in the future. A study of a more sophisticated cost function for the minimization process is under way. Ideally, all structure anomalies, even very local and non-repetitive ones, should be taken into account in the minimization process. We expect that oriented multi-resolution feature-detecting filters will provide better support for the cost function than actual Fourier amplitude spectra. Another promising research direction would be a more systematic study of influence of distribution of key levels on the achieved quality.

Our algorithm has been developed in an academic framework. It is our firm intention to put it into the public domain, and provide a core implementation, publicly available from our web site[1]. Further improvements will be available at the same location.

This error-diffusion algorithm belongs to a family of fundamental algorithms for computer graphics since it can be applied to a large variety of applications where the visual quality of the output with a restricted palette plays an important role. It may be efficiently used in general-purpose visualization, in network-based imaging, in printing, in quantizers attached to compression algorithms, etc.

Computers become faster and faster; but this does not mean that considerations of speed become unimportant in digital halhtoning. With today's proliferation of small and portable devices, the visualization software may run on a cheap device having restricted resources. We believe that a simple and efficient halftone algorithm may be particularly useful in such devices.

The algorithm presented in this article is very simple, and we believe this to be one of its main virtues. For anyone who might think that it is too simple, let us cite Karl Menger, a famous Austrian mathematician, who once said that "one should never reason that an idea is too simple to be correct" [8].

We hope that the simplicity of the presented algorithm, along with its visual quality and public availability, will lead to its adoption by the computer graphics community.

4 Acknowledgements

The author would like to express his acknowledgements to many people who were involved at different degree in discussions related to the present paper. I would like to thank especially Frédo Durand, Bob Ulichney, Jan Allebach, Gabriel Marcu, Reiner Eschbach, Justin Legakis, Julie Dorsey, Shlomo Gortler, Roger Hersch, Isaac Amidror, Byong Mok Oh, Max Chen, Neil Stewart, Jean Vaucher, Sebastien Roy, Ovarith Troeung, Aurelien Calais and Pierre-Marc Jodoin.

Thanks to Bob Ulichney, Jan Allebach, Gabriel Marcu, Luiz Velho and Kevin Parker for providing sample images of their original halftoning techniques.

References

[1] J. P. Allebach. Private communication, 2001.

[2] M. Analoui and J. P. Allebach. Model based halftoning using direct binary search. *SPIE*, 1666:96–108, 1992.

[3] B. E. Bayer. An optimum method for two-level rendition of continuous-tone pictures. *IEEE Intl. Conf. on Communications*, 1:2611–2615, 1973.

[4] R. Eschbach. Reduction of artifacts in error diffusion by mean of input-dependent weights. *JEI*, 2(4):352–358, 1993.

[5] R. Eschbach and R. Hauck. A 2-D pulse density modulation by iteration for halftoning. *Optics Communications*, 62(5):300–304, 1987.

[6] R. Eschbach and K. T. Knox. Error-diffusion algorithm with edge enhancement. *JOSA (A)*, 8(12):1844–1850, 1991.

[7] R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. *Proc. Soc. Inf. Display*, 17:75–77, 1976.

[8] L. Golland and K. Sigmund. Exact thought in a demented time: Karl Menger and his viennese mathematical colloquium. *Math. Intelligencer*, 22(1):34–45, 2000.

[9] J. F. Jarvis, C. N. Judice, and W. H. Ninke. A survey of techniques for the display of continuous tone pictures on bilevel displays. *Computer Graphics and Image Processing*, 5:13–40, 1976.

[10] H.R. Kang. *Digital Color Halftoning*. SPIE Press, 1999.

[11] K. T. Knox. Edge enhancement in error diffusion. In *Advance Printing of Paper Summaries, SPSE's 42nd Annual Conference*, pages 310–313, Boston, MA, May 1989.

[12] K. T. Knox. Error diffusion: A theoretical view. *SPIE*, 1913:326–331, 1993.

[13] K. T. Knox. Evolution of error diffusion. *JEI*, 8(4):422–429, 1999.

[14] D. J. Lieberman and J. P. Allebach. A dual interpretation for direct binary search and its implications for tone reproduction and texture quality. *IEEE Trans. on Image Processing*, 9:1950–1963, 2000.

[15] G. Marcu. An error diffusion algorithm with output position constraints for homogeneous highlights and shadow dot distribution. *JEI*, 9(1):46–51, 2000.

[16] T. Mitsa and K. J. Parker. Digital halftoning using a blue-noise mask. *Journal of the Optical Society of America A*, 9(11):1920–1929, 1992.

[17] V. Ostromoukhov, R. D. Hersch, and I. Amidror. Rotated dispersion dither: a new technique for digital halftoning. *Proceedings of SIGGRAPH 94*, pages 123–130, 1994.

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes*. Cambridge University Press, 1989.

[19] C. I. Rosenberg. Measurement-based evaluation of a printer dot model for halftone algorithm tone correction. *JEI*, 2(3):205–212, 1993.

[20] J. Shiau and Z. Fan. Method for quantization gray level pixel data with extended distribution set, 1994. US patent 5,353,127.

[21] J. Shiau and Z. Fan. A set of easily implementable coefficients in error diffusion with reduced worm artifacts. *SPIE*, 2658:222–225, 1996.

[22] P. Stucki. Mecca-a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction, 1981. Research Report RZ1060, IBM Res. Lab.

[23] R. Ulichney. *Digital Halftoning*. MIT Press, 1987.

[24] R. Ulichney. The void-and-cluster method for dither array generation. *SPIE*, 1913:332–343, 1993.

[25] R. Ulichney. A review of halftoning techniques. *SPIE*, 3963:378–391, 2000.

[26] L. Velho and J. Gomes. Digital halftoning with space filling curves. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):81–90, 1991.

[27] L. Velho and J. Gomes. Stochastic screening dithering with adaptive clustering. *Proceedings of SIGGRAPH 95*, pages 273–276, 1995.

[28] I. H. Witten and R. M. Neal. Using peano curves for bilevel display of continuous-tone images. *IEEE Computer Graphics & Appl.*, 2:47–52, 1982.

[1] `http://www.iro.umontreal.ca/˜ostrom/varcoeffED`

APPENDIX I

This table shows the distribution coefficients in form of $i : A_{10}, A_{-11}, A_{01}$ for the input levels i in the interval [0..127]. $d_{10}(i) = A_{10}(i)/M(i); d_{-11}(i) = A_{-11}(i)/M(i); d_{01}(i) = A_{01}(i)/M(i);\ M(i) = (A_{10}(i) + A_{-11}(i) + A_{01}(i))$, and $D(i) = \{d_{10}(i), d_{-11}(i), d_{01}(i)\} == D(255 - i)$. Highlighted are the key levels.

0:	13,	0,	5
1:	13,	0,	5
2:	21,	0,	10
3:	7,	0,	4
4:	8,	0,	5
5:	47,	3,	28
6:	23,	3,	13
7:	15,	3,	8
8:	22,	6,	11
9:	43,	15,	20
10:	7,	3,	3
11:	501,	224,	211
12:	249,	116,	103
13:	165,	80,	67
14:	123,	62,	49
15:	489,	256,	191
16:	81,	44,	31
17:	483,	272,	181
18:	60,	35,	22
19:	53,	32,	19
20:	237,	148,	83
21:	471,	304,	161
22:	3,	2,	1
23:	481,	314,	185
24:	354,	226,	155
25:	1389,	866,	685
26:	227,	138,	125
27:	267,	158,	163
28:	327,	188,	220
29:	61,	34,	45
30:	627,	338,	505
31:	1227,	638,	1075
32:	20,	10,	19
33:	1937,	1000,	1767
34:	977,	520,	855
35:	657,	360,	551
36:	71,	40,	57
37:	2005,	1160,	1539
38:	337,	200,	247
39:	2039,	1240,	1425
40:	257,	160,	171
41:	691,	440,	437
42:	1045,	680,	627
43:	301,	200,	171
44:	177,	120,	95
45:	2141,	1480,	1083
46:	1079,	760,	513
47:	725,	520,	323
48:	137,	100,	57
49:	2209,	1640,	855
50:	53,	40,	19
51:	2243,	1720,	741
52:	565,	440,	171
53:	759,	600,	209
54:	1147,	920,	285
55:	2311,	1880,	513
56:	97,	80,	19
57:	335,	280,	57
58:	1181,	1000,	171
59:	793,	680,	95
60:	599,	520,	57
61:	2413,	2120,	171
62:	405,	360,	19
63:	2447,	2200,	57
64:	11,	10,	0
65:	158,	151,	3
66:	178,	179,	7
67:	1030,	1091,	63
68:	248,	277,	21
69:	318,	375,	35
70:	458,	571,	63
71:	878,	1159,	147
72:	5,	7,	1
73:	172,	181,	37
74:	97,	76,	22
75:	72,	41,	17
76:	119,	47,	29
77:	4,	1,	1
78:	4,	1,	1
79:	4,	1,	1
80:	4,	1,	1
81:	4,	1,	1
82:	4,	1,	1
83:	4,	1,	1
84:	4,	1,	1
85:	4,	1,	1
86:	65,	18,	17
87:	95,	29,	26
88:	185,	62,	53
89:	30,	11,	9
90:	35,	14,	11
91:	85,	37,	28
92:	55,	26,	19
93:	80,	41,	29
94:	155,	86,	59
95:	5,	3,	2
96:	5,	3,	2
97:	5,	3,	2
98:	5,	3,	2
99:	5,	3,	2
100:	5,	3,	2
101:	5,	3,	2
102:	5,	3,	2
103:	5,	3,	2
104:	5,	3,	2
105:	5,	3,	2
106:	5,	3,	2
107:	5,	3,	2
108:	305,	176,	119
109:	155,	86,	59
110:	105,	56,	39
111:	80,	41,	29
112:	65,	32,	23
113:	55,	26,	19
114:	335,	152,	113
115:	85,	37,	28
116:	115,	48,	37
117:	35,	14,	11
118:	355,	136,	109
119:	30,	11,	9
120:	365,	128,	107
121:	185,	62,	53
122:	25,	8,	7
123:	95,	29,	26
124:	385,	112,	103
125:	65,	18,	17
126:	395,	104,	101
127:	4,	1,	1

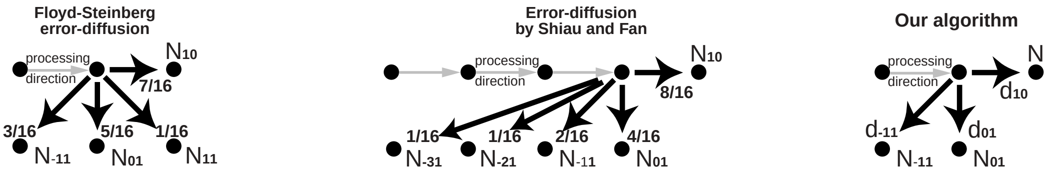

Figure 1: Distribution coefficients as in Floyd-Steinberg error-diffusion algorithm (left), in Shiau and Fan's algorithm (middle), and in the proposed variable-coefficients error-diffusion algorithm (right).

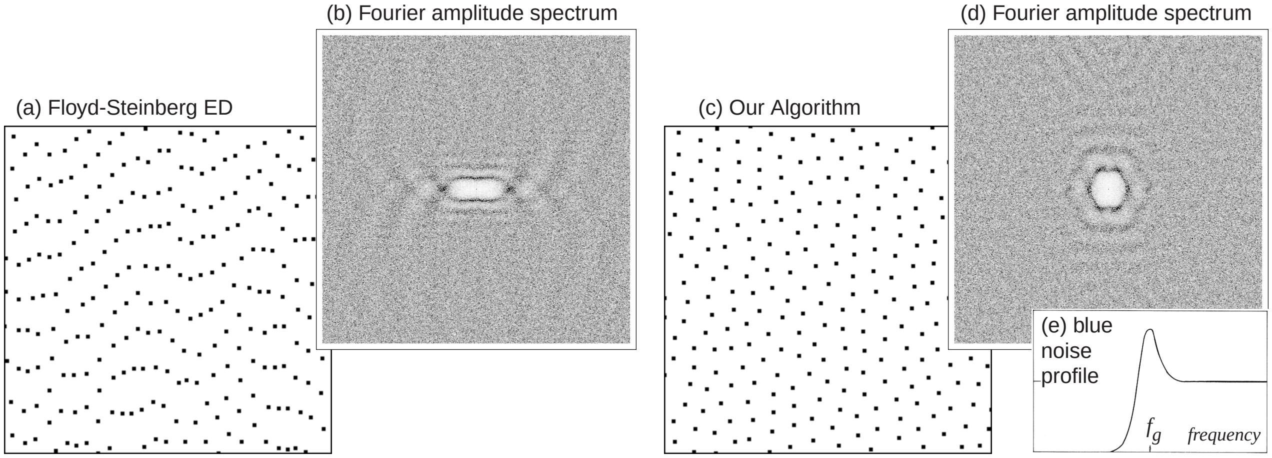

Figure 2: A light gray uniform patch halftoned with the serpentine-path Floyd-Steinberg E-D algorithm (left) and with the proposed algorithm (right). The blue noisie profile is taken from [23].

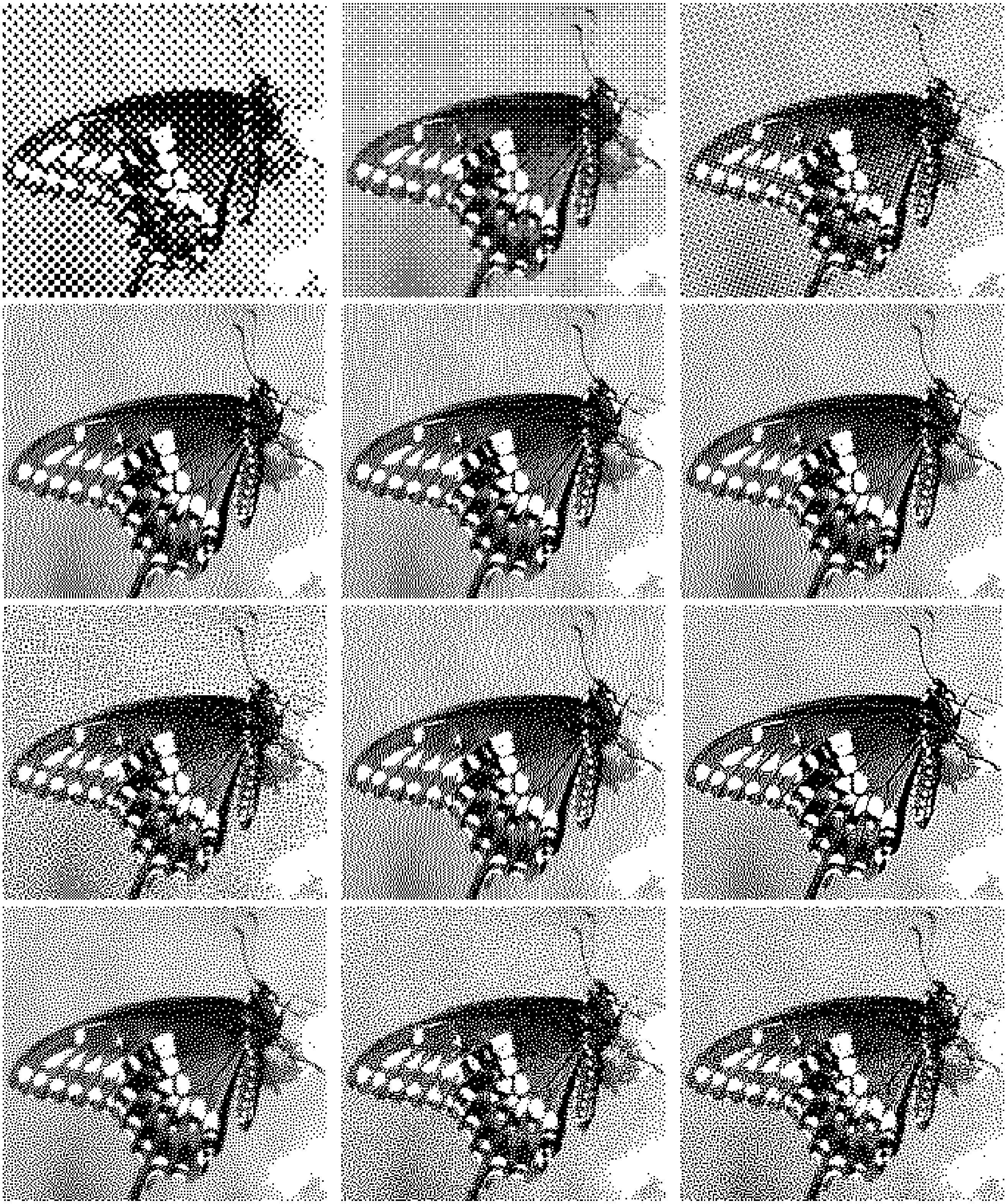

Figure 3: A grayscale sample produced with different halftoning techniques, printed at 100 dpi. **Top row,** left: clustered-dot dither, cluster size = 32 pixels; middle: Bayer's dispersed dither [3]; right: Rotated dispersed dither by Ostromoukhov et al. [17]. **Second row,** left: Shiau-Fan E-D [21]; middle: Floyd-Steinberg E-D [7]; right: our new E-D method. **Third row,** left: halftoning with space-filling curves by Velho and Gomes [26, 27], cluster size=1 (Courtesy of Luiz Velho); middle: Floyd and Steinberg filter with 50% random weights processed on a serpentine raster [23] (Courtesy of Robert Ulichney); right: Marcu's roadmap halftoning [15] (Courtesy of Gabriel Marcu). **Bottom row,** left: DBS-based halftoning [2] (Courtesy of Jan Allebach); middle: void-and-cluster [24] (Courtesy of Robert Ulichney); right: blue noise mask [16] (Courtesy of Kevin Parker).

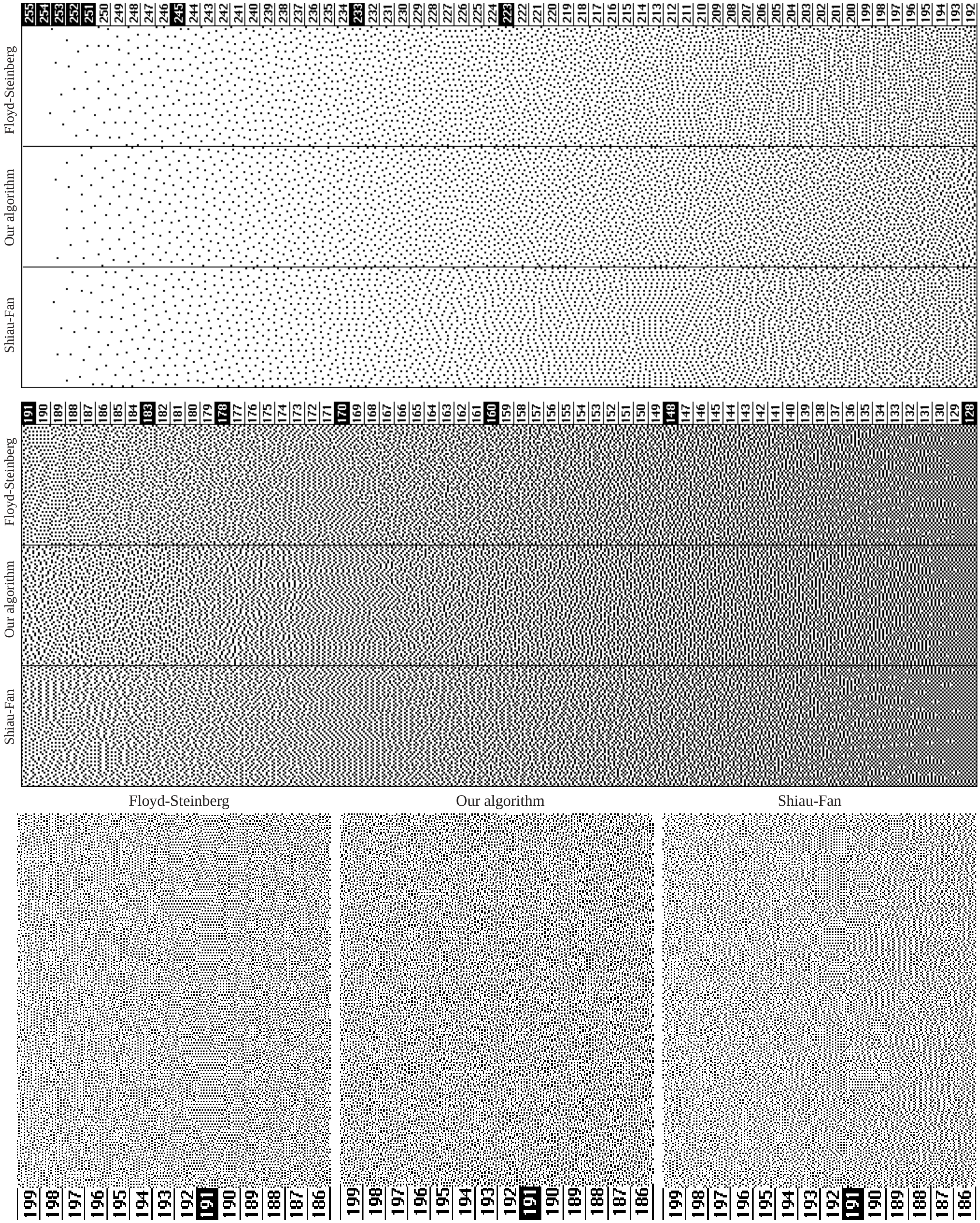

Figure 4: Grayscale ramps produced with our method, compared with very good E-D algorithms: Floyd-Steinberg E-D [7] (public domain) and Shiau-Fan E-D [20, 21], (proprietary). Images are shown here at typical display resolutions: 75 dpi (above) and 100 dpi (below). Highlighted are the key levels.