# An Error Diffusion Algorithm with Homogenous Response in Highlight and Shadow Areas

***Reiner Eschbach***
***Xerox Digital Imaging Technology Center***
***800 Phillips Road, 0128-27E***
***Webster, New York 14580***

## Abstract

This paper describes a modified error diffusion algorithm that results in a homogenous pulse distribution in the highlight and shadow areas of the binarized images. The distinction to standard error diffusion algorithms is a dynamic threshold imprint function that depends on the local input values and binary output pixel. The imprint function is generated by diffusing a one-dimensional function in the othogonal direction, thereby allowing a fast implementation of a two-dimensional threshold imprint.

## 1 Introduction

The error diffusion algorithm, as originally proposed by Floyd and Steinberg,[1] has found widespread application in the printing of digital images, especially on devices that have a stable reproduction of isolated dots, such as ink-jet printers. Over the years, a large number of modifications have been made to the error diffusion algorithm, trying to improve the pulse distribution,[2-6] printability,[7-9] detail response,[10] etc.[11] Additionally, several modifications were performed with the intention to adapt the error diffusion algorithm to different input types, such as business graphics and scanned text.

One of the major areas of modification to the error diffusion algorithm was the elimiation of error diffusion specific artifacts. Here, two major artifact types can be distinguished

1. artifacts that occur around the input levels 1/2, 1/3, 1/4, etc. and have the form of stable patterns that "randomly" change into another stable pattern[3,12]—sometimes referred to as "fingerprints", and
2. artifacts that occur in the highlight and shadow region, i.e.: at very high and very low input levels, sometimes referred to as "worms".

The first type of artifacts is often reduced or eliminated by adding random, or pseudo-random effects into the error diffusion algorithm, such as adding a threshold matrix into the binary output decision of the algorithm,[2] randomizing the weight-distribution,[13,14] or injecting noise into the input or the threshold.[15] A theroretical analysis of the stability of different output patterns can be found in an article by Fan.[12]

Similar approaches have been used to reduce the second artifact. However, due to the large spatial range of the worm artifacts, injection of noise and pseudo-random effects have generally a lower influence on the output structure in shadow and highlight regions. Changing the weights—as in the randomization of the error diffusion weights—has a larger effect on the worms, since the effect of the weighting coefficients is not spatially limited, as for instance the addition of a threshold matrix.

It should be clear, that in practice every change in the error diffusion algorithm will influence both artifacts, with the exception of some hybrid algorithms that change their behaviour as a function of the input values. Two straight-forward methods are the change of the error feedback in highlight and shadow regions and the use of different error weights for different input intensity regions.[16]

The algorithm described in this paper deviates from the previous approaches by introducing a threshold imprint function into the error diffusion process that controls the local relationsship between pulses. This imprint function is designed in a way that the effects of the imprint cancel each other out when the correct number of black and white output pixel is set. Additionally, the imprint function can be designed to have zero amplitude in the midtones, thereby only affecting the highlight and shadow regions and leaving the midtone region of the error diffused image unaltered.

## 2 Error Diffusion

The error diffusion algorithm has been introduced to the binarization of image data in 1975 by Hale[17] and Floyd & Steinberg.[1] The algorithm is strongly related to the $\Sigma$ $\delta$-modulation that is used in 1-dimensional signal processing. In the error diffusion (ED) algorithm, the input pixel value $\boldsymbol{i}$ is updated using past errors, resulting in a "modified input" value $\boldsymbol{i_{mod}}$. This value is then compared to a threshold $\boldsymbol{t}$ to determine the binary output value $\boldsymbol{b}$. If $\boldsymbol{i_{mod}}$ exceeds $\boldsymbol{t}$, the output is set to "***1***", otherwise to "***0***". Here, "***1***" and "***0***" refer to the two binary output states that can be represented on the screen or on paper. Differences between absorbance, reflectance, etc have to be taken into account for the processing, and literature exists describing the processing of images where the data space (or color space) for the images, the error calculation, and the reproduction might be different.

The output value $\boldsymbol{b}$ is then compared to the modified input $\boldsymbol{i_{mod}}$, and an error term $\boldsymbol{e}$ is calculated via $\boldsymbol{e = b - i_{mod}}$. This error is used to generate the correction terms for future input pixel values. In the simplest form, the error is directly subtracted from the next input pixel to generate the new modified input value

$$i_{\text{mod}}(m,n) = i(m,n) - e(m - \ell, n). \quad (1)$$

This equation can easily be extended to 2-dimensions by adding weighted error terms from different pixels to the current pixel:

$$i_{\text{mod}}(m,n) = i(m,n) - \sum_{k,\ell \in S} a_{k,\ell} \times e(m-k, n-\ell), \quad (2)$$

where $k, \ell \in S$ refers to all the neighboring pixels within a predefined neighborhood $S$. It is common to have the error weights sum to “1”

$$\sum_{k,\ell \in S} a_{k,\ell} = 1, \quad (3)$$

because this guarantees a reproduction of the average graylevel of the input image.

The requirement of Eq. (3) does not have to be fullfilled dependent on the application. Several papers have shown the effects of using weights that do not sum to one.[18-22] It should additionally be noted, that fullfilling Eq. (3) is not a sufficient requirement for the numerical stability of the process and the correct visual reproduction of the input image.

The binarization step can be written as

$$b(m,n) = step\left\{ i(m,n) - \sum_{k,\ell \in S} a_{k,\ell} \times e(m-k, n-\ell) - t(m,n) \right\}, \quad (4)$$

where b(m,n) is the binary output at location (m,n), step{x} is the step function with step{x} = 0 for x<0 and step{x}=1 otherwise and t(m,n) is the threshold function. The standard error diffusion uses a constant (commonly equal to 0.5) as threshold, but varying thresholds have been used to modify the output texture, inhibit unwanted patterns, edge enhance the result,[23] and modify the granularity of the binary patterns. The error at location (m,n) is simply calculated as

$$e(m,n) = b(m,n) - \left[ i(m,n) - \sum_{k,\ell \in S} a_{k,\ell} \times e(m-k, n-\ell) \right]. \quad (5)$$

Several papers[20,22,15] have been published that give a theoretical description of the spectral properties of the error diffusion algorithm as described in Eq. (4).

## 2.1 Shadow and Highlight Behavior of Error Diffusion

This paper describes a method to improve the error diffusion response in highlight and shadow areas, where we assume that homogenous pulse distribution is preferred over a non-homogeneous one. Figure 1 shows an example of a typical pulse distribution generated by error diffusion. In this case, the standard weights as published by Floyd and Steinberg were used. The left part of Figure 1 shows the output for the input level of 250 (out of 255) and the right half shows the response for an input of 253.

It is clear from Figure 1 that the output distribution in the highlight area is highly non-homogeneous. The equivalent holds true for the shadow area, where the white pixels are non-homogeneously distributed. Please note, that we are operating on “digital counts” in the example. In printing applications the additional calibration step will lead to a mapping of a requested digital input value to a corrected value that will reproduce the correct gray level when printed. It is this printer calibra-

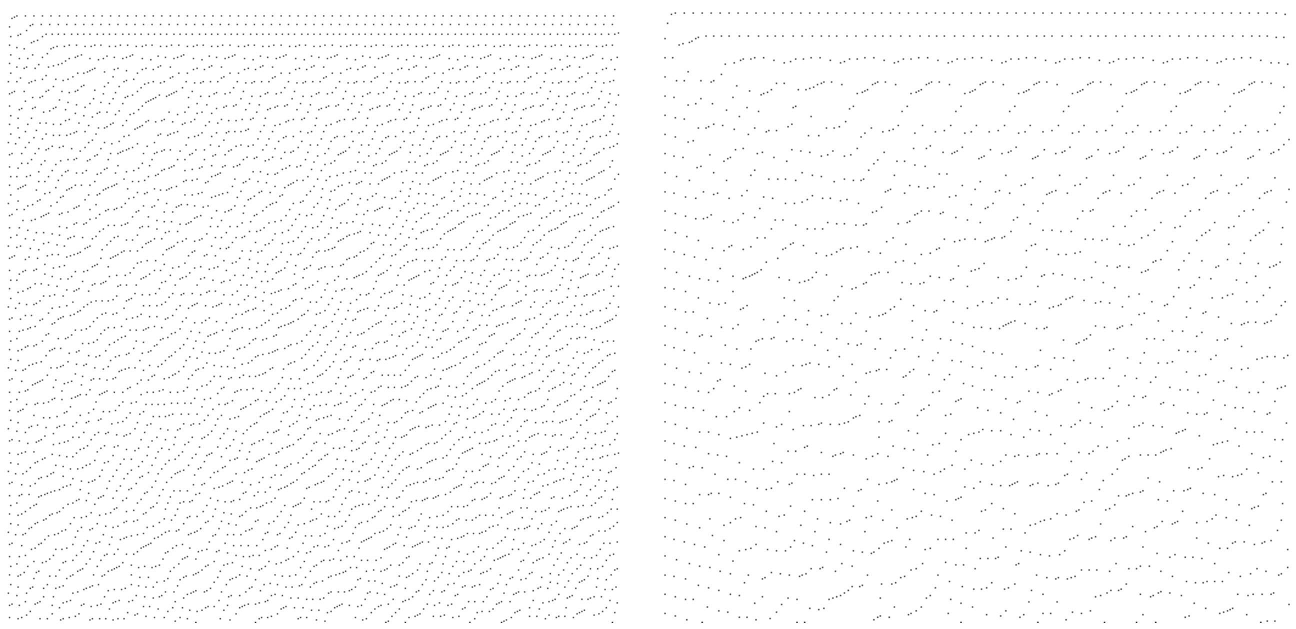
*Figure 1. Error diffusion output for the input levels 250 (left) and 253 (right) for an input range from 0 to 255.*

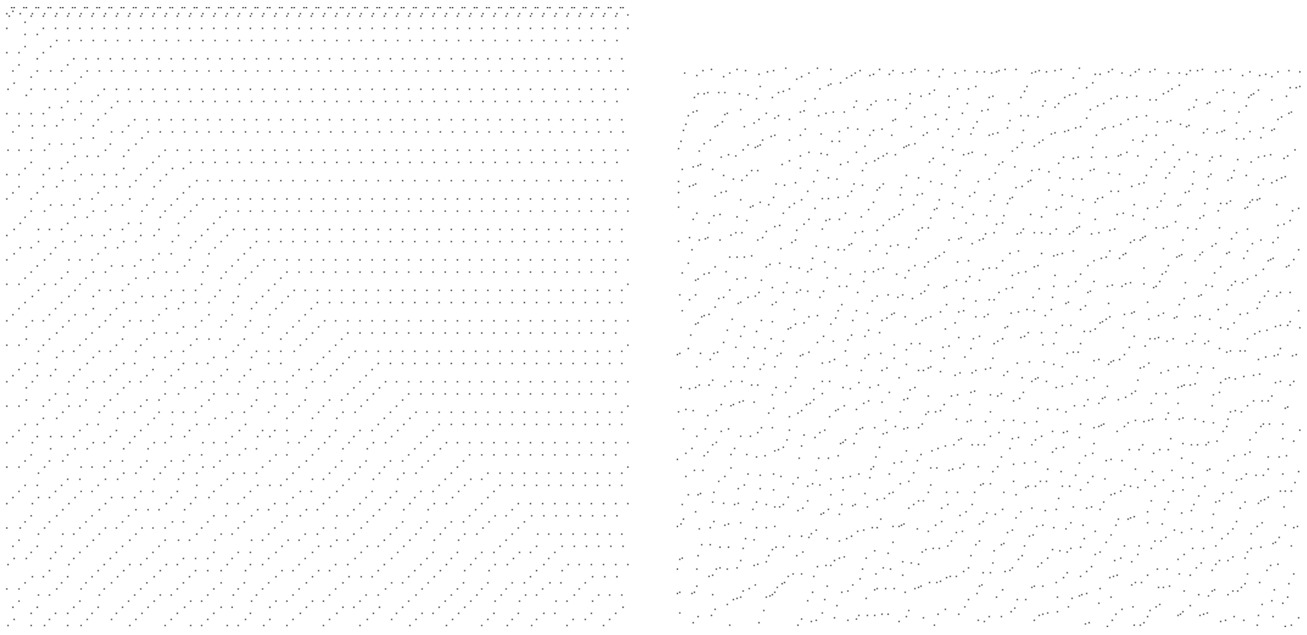

*Figure 2. Error diffusion output for level 253, where the threshold has been randomized using a dither pattern of amplitude 127 ( left ) and white noise of the same amplitude ( right ).*

tion mapping that causes the error diffusion algorithm to produce a different output pattern in shadows and in highlights.[24] On ink-jet printers with an oversized print-dot it is quite common to have a very visible artifact in the highlight regions, but an essentially invisible effect in shadow areas. This discrepancy between shadow and highlight response is exclusively a funtion of the printer calibration and will not be considered in this paper. Several papers by a variety of authors have demonstrated the calibration of error diffusion algorithms.

Figure 2 shows the effect of randomizing the error diffusion threshold. The left half of Figure 2 shows the effect of adding dither pattern of amplitude 127 to the threshold and the right half shows the effect of adding white noise of amplitude 127 to the threshold. Both methods have a clear influence on the dot structure of the result, but neither necessarily should be considered a clear improvement.

Figure 3 shows the effect of using a larger weight distribution matrix for the same highlight value. The binary distribution is distinct from the ones shown in Figure 1 and 2, but it is not clear that any of the distributions should be considered clearly better than the others.

From Figures 1, 2 and 3, it should be clear that an error diffusion algorithm with "better" performance in the highlight and shadow regions would be beneficial. However, no absolute definition of better exists in this context. "Better" oftern refers to user "tastes" and preferences. Consequently, there is no clear metric as to which pulse distribution of Figures 1 and 2 is "better" than the others. In this paper, we will use visual homogenity of the output pulse distribution as a metric for better, acknowledging that some observers might prefer non-homogeneous distributions. The criterion of a homogeneous distribution is correlated to several other criteria that were formulated in the past, e.g.: no low-frequency noise and no clear spikes in the spectrum; a radial frequency spectrum that exhibits blue-noise characteristics, etc.

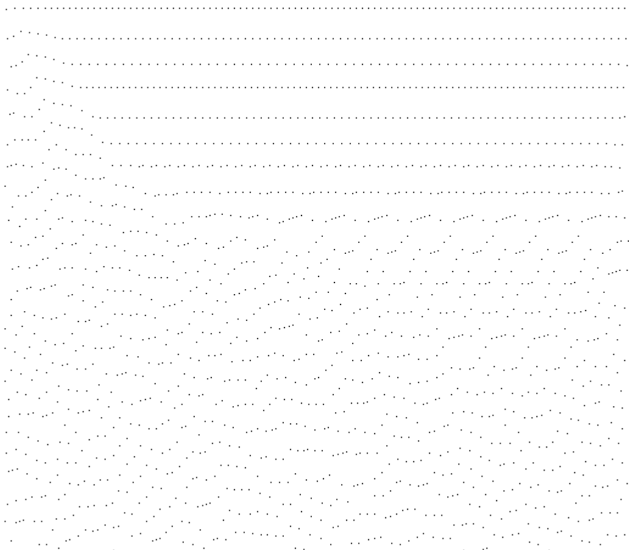

*Figure 3. Effect of using a large error diffusion weight matrix for the input level of 253.*

## 2.2 Threshold Modulated Error Diffusion

In order to produce more homogeneous pulse distribution in the highlight and shadow areas, we make use of an output dependent threshold modulation. Output dependent threshold modulation has been used by Fawcett and Schrack to break up unwanted patterns in the midtone regions and by Levien to create a stronger clustering of output pulses to aide the print-stability of the error diffused images.

It is clear that a threshold variation can have a strong impact on the local pulse distribution. At the same time, the overall graytone reproduction can be maintained, since the threshold modulation has the same effects as a phase modulation of the pulses. If the threshold changes by an amount $\Delta T$, the average "distance" over which this modulation effects the pulses is $o(\Delta T/<I>)$ pixels, where $<I>$ denotes the average input value. This effect can be easily observed in the edge enhanced error diffusion algorithm. The total DC error between input and output in threshold modulated error diffusion is similar to the total DC error in standard error diffusion. In both cases, the total error is a function of the threshold range. For standard error diffusion and dynamic range of $0 \leq$ input $\leq 1$, a threshold of 1/2 leads to a total error of $|e_{total}| \leq 1/2$, a threshold of 50, leads to a total error of $|e_{total}| \approx 50$, in the stable state. For threshold modulation, the same rule applies, bearing in mind the effective range of the threshold variation for the stable state.

## 3 Threshold Imprints

In order to create a more homogeneous pulse distribution in the shadow and highlight areas, threshold modulation has to dicourage "black" pixels in the neighborhood of other "black" pixels in the highlight area, and discourage "white" pixels in the neighborhood of other "white" pixels in the shadow area. This is achieved by raising the threshold (assuming "white" = "1") in the shadow region as soon as a "white" pixel is set, thereby reducing the likelihood of another "white" pixel nearby. The corresponding operation is done for the highlight region, resulting in a lowering of the threshold as soon as a "black" pixel is set, therebye reducing the likelihood of a "black" pixel nearby.

Changing the threshold in the neigborhood of a pixel can be done using a threshold imprint. This is illustrated in Figure 4 for a simple one-dimensional (1-D) case.

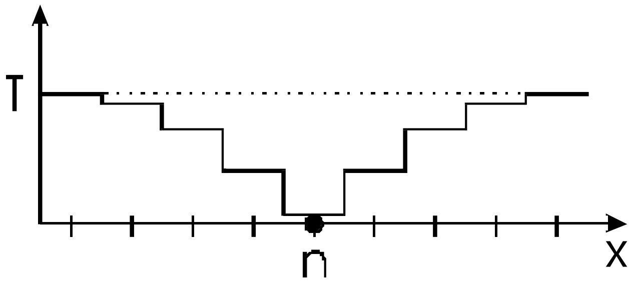

*Figure 4. Setting an output pixel at position n to black (solid circle) causes the threshold (solid line) to be lowered in the neighborhood of the pixel.*

Once a black pixel is set in the output at location (solid circle in Figure 4), the threshold T (solid line) is lowered in the neighborhood ($n \pm \Delta$) of the output pixel. This increases the likelihood of the next pixel to be above the threshold. The reason for using a threshold imprint that also modifies the threshold of pixels already processed will become clear in Section 3.2.

After processing pixel n and adjusting the threshold in the neighborhood, pixel n+1 is processed. This leads to an output decision for pixel n+1 and a subsequent threshold imprint based on that decision. Assuming that pixel n+1 is set to white, the threshold is now modified to decrease the likelihood of a white pixel at location n+2. This is shown in Figure 5, where the dotted line shows the original threshold of Figure 5 and the solid line shows the threshold after pixel n+1 is set to white (open circle). Note, that the threshold adjustment for pixel n+1 is added to the adjustment for pixel n.

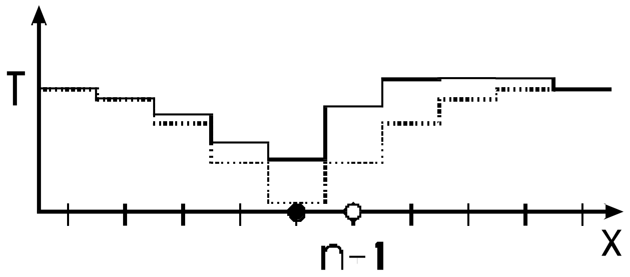

*Figure 5. Setting the output pixel n+1 causes a new threshold imprint that is added to the previous imprints, resulting in the threshold as indicated by the solid line.*

In this way, the threshold is constantly updated as a function of the past output pixels. It is clear, that the amount of the adjustment—or the strength of the threshold imprint—has to be balanced to warranty the correct distribution of the output pixels.

The description so far requires the generation of two-dimensional (2-D) threshold imprints that are a function of the output pixel and the current input value. In order to reduce the design complexity, one can separate the different parameters into the amplitude of the imprint, the method to generate a 2-D imprint from a 1-D imprint, and the shape of the actual imprint, i.e.: the Stamp function.

### 3.1 Threshold Imprint Amplitude

The threshold imprints of the different previously processed pixels superpose each other to form the threshold at any given, as yet unprocessed, pixel. Several restrictions can be imposed on the imprints in order to achieve a homogeneous pulse distribution in the shadows and highlights:

- the threshold imprints should cancel each other out for a homogeneous pulse distribution
- the threshold imprint should disappear in the midtone region

These restrictions are not mandatory, but can help in developing an imprinting scheme. Both requirements lead to an imprint dependency on the local input value (original input value, not the modified input value). The first requirement says that inside an area of brightness 1/4, the imprints of 3 black and one white output pixel should cancel each other out. Although not necessary, this requirement is sufficient to limit the total dynamic range of the threshold. At this point, no determination has been made about the shape and extent of the threshold imprint. For simplicity we assume throughout this paper that there is one threshold imprint function with an imprint amplitude that is a function of the output pixel value and the input value. This separation of imprint form and amplitude separates the free parameters in the threshold imprint and thus allows an easier control of the algorithm. Using $A_W(I)$ to denote the amplitude of the threshold imprint for input value $I$, one can write:

$$A_W(I) = -\frac{I}{255 - I} A_B(I), \qquad (6)$$

where we assume that the input dynamic range is between 0 and 255. Eq. (6) gives the relationsship between the threshold imprint between a white and a black output pixel for the same input value, if one wants to enforce the requirement that the threshold imprints in the equilibrium should cancel each other out. It should be noted, that this requirement is artificial and only used to reduce the number of free parameters for the purpose of this paper.

It is desired that the algorithm behaves symmetrically between black and white (this is a restriction one might want to modify in certain situations). This can be written as:

$$A_W(I) = -A_B(255 - I). \qquad (7)$$

Using Eqs. (6) and (7) one only needs to determine the amplitude of one of the imprints (say for white oputput pulses) for half of the input range.

Assuming one only wants to modify the output pixel distribution in the shadow and highlight area, one would define an imprint amplitude that vanishes in the midtones. For the purpose of this paper, an amplitude of the form

$$A_w(I) = C\,\frac{(I - 127.5)^3}{127.5^3} \qquad \forall I > 127, \qquad (8)$$

where $C$ is a constant.

It should be noted that the amplitude does not need to be defined using a mathematical function, but that “arbitrary” input/output pairs are valid in the description. The main assumption behind Eq. (8) is that the amplitude increases towards the highlight (shadow) area and that it vanishes in the midtones. The actual values of $A_W(I)$ are not calculated inside the algorithm but rather are stored inside a look-up table. The complexity of $A_W(I)$ is therefore not a factor in the execution speed of the algorithm—error weight distribution and stampform (see section 3.3) and size are a factor.

## 3.2 Two-Dimensional Threshold Imprints

Generating a 2-D threshold imprint for every image pixel can be a time consuming process. As a simplification, we will describe the use of a 1-D threshold imprint with a subsequent decay of that imprint into the next image scanline. In order to perform this operation, we assume that the threshold at an individual pixel consists of three different parts, 1) the threshold of error diffusion—normally 1/2, but see also Ref.[19] and [15], 2) a dampened version of the threshold imprints in the previous scanline, and 3) the imprints, $S$, from the actual scanline:

$$T(m,n) = T_{ED} + \sum_j d_j \times T_{imp}(m-1, n-j) + \sum_{j=1}^{n-1} S(j,n). \qquad (9)$$

Here the summation over $S$ describes the cumulative effect of the threshold imprints for the current scanline. $S(j,n)$ denotes the influence that an imprint at location $j$ has on location $n$. Care has to be taken in this summation, to include the correct weighting coefficients of the different imprints along the scanline. The summation over the threshold imprint of the previous scanline, $T_{imp}(m-1, x)$, creates a 2-D effect from the 1-D threshold imprints. In the following we will use a very simple form with $d_j = 0$ for $j \neq 0$, which simplifies the calculation of the effective threshold. The initial threshold of a new scanline is thus the standard error diffusion threshold plus a dampening factor $d_0$ times the imprint threshold of of the pixel exactly above the new pixel.

The reason to distinguish $T(m,n)$ and $T_{imp}(m,n)$ lies in the possibility to introduce additional threshold modulation into $T_{ED}$. One example is the threshold modulation for edge enhancement, which will be shown in chapter 4.1.

The method of creating a 2-D threshold imprint from a 1-D threshold imprint is the reason for using an imprint that also modifies the threshold at pixels that have already been modified, as mentioned in chapter 3. Obviously, one might use either pure trailing or pure leading threshold imprints, but the choice will influence the choice of $d_j$ in Eq. (9).

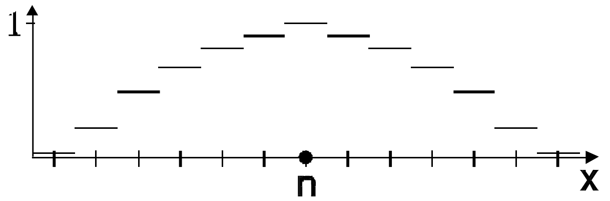

*Figure 6. Example for a stamp function centered around position n. This is the stamp function used throughout this paper.*

## 3.3 Threshold Imprint Form: the Stamp-Function

The previous two sections explained the generation of 2-D threshold imprints without consideration for the actual shape of the imprint. The shape of the imprint can be understood as a stamp that is pressed into the deforming threshold with a strength or amplitude that is a function of the current input intensity. Figure 6 shows a graphic representation of the stamp function used in this paper. The stamp function is centered around location $n$, which in the experiments will be identical with the pixel currently being processed. It should be understood that

Table 1. **Numerical values and relative positions of the stamp function used in this paper.**

| position | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| value | 0.045 | 0.23 | 0.5 | 0.68 | 0.82 | 0.91 | 1 | 0.91 | 0.82 | 0.68 | 0.5 | 0.23 | 0.045 |

the actual choice of the stamp function was arbitrary and that there is no requirement for it to be symmetric around the current pixel, nor is there any requirement for it to cover the number of pixels shown in Figure 6.

The exact form of the stamp function used in this paper is given in table 1.

The stamp function as given in Table 1 is arbitrary and smaller and non-symmetric stamp functions have been used. In all cases, however, there is a balance between the

- error diffusion weights
- stamp function
- dampening factor and
- imprint amplitude

that has to be taken into account when implementing the algorithm.

### 3.4 Threshold Imprint Algorithm

The implementation of the proposed algorithm is straightforward. The threshold comparison that is normaly done with respect to a constant threshold in standard error diffusion, or to an input dependent threshold in edge enhanced error diffusion is changed to an input and output dependent comparison. This is shown in Figure 7, where the original error diffusion algorithm is contained in the broken-line rectangle. In the modified version, the input and output values are used to determine the amplitude of the threshold imprint according to Eqs. (6)-(8). This threshold imprint is then combined with the current threshold (thereby updating the current threshold) and the combination is used as the threshold for the binarization decision.

As can be seen from Figure 7, the algorithm now has a dependency of the error diffusion threshold on both input and output data.

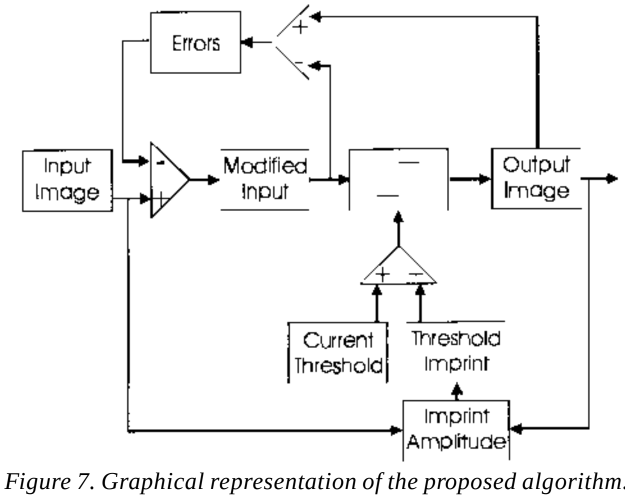

*Figure 7. Graphical representation of the proposed algorithm.*

## 4 Experimental Results

The proposed algorithm has been tested on several images. In all cases, the original weights suggested by Floyd and Steinberg have been used for the algorithm, although other weight distributions could be substituted. Figure 8a shows the result of binarizing some shadow and highligh patches using the standard error diffusion weights as suggested by Floyd and Steinberg. The artifacts are clearely visible in the four patches, with patch values “3”, “5”, “10”, and “250” from top-left to bottom right. The background has a graylevel of “50”.

Figure 8b shows the result of applying the proposed algorithm to the same input pattern. As can be seen in Figure 8b, the artifacts in the dark and light patches are greatly reduced.

Figure 9 shows the result of applying the standard error diffusion algorithm to the image of a building facade. Figure 10 shows the same input image, this time binarized using the proposed algorithm. The stamp-function values of Table 1 were used, the constant C of Eq. (8) was set to C=150, and the dampening factor $d_0$ of Eq. (9) is set to $d_0 = 0.8$. The distribution of the output pixels in the shadow region (e.g., in the balcony area, or under the overhangs) is more homogeneous than in Figure 9.

Figure 11 shows another example of a binarized image, this time an artistic image with a large shadow region. The artifacts caused by the error diffusion algorithm are clearly visible and distracting. Figure 12 shows the result of using the proposed algorithm on the same input. The pulse distribution in the shadow region shows a more homogeneous bahaviour than in Figure 11.

### 4.1 Interactions of Threshold Imprints with other Threshold Modulations

The results given so far, show the effect of the threshold imprints on the homogeneity of the output pulses. The following example shows the combination of the threshold imprints with the threshold modulation for edge enhancement as suggested by Eschbach and Knox. In this case, the two threshold modulations are treated in an additive way, resulting in an effective threshold of

$$T(m,n) = T_{cons.} - \kappa \times I(m,n) + d_0 \times T_{imp}(m-1,n) + \sum_{j=1}^{n-1} S(j,n), \quad (10)$$

where $\kappa$ is the edge enhancement coefficient as defined in ref(10). $T_{cons}$. is the constant component of the threshold, set to 1/2. Figure 13 shows an image processed with the proposed algorithm with no added edge enhancement. Figure 14 shows the result for an edge enhancement of $\kappa = 2$. As can be seen in Figure 14, the pulse distribution in the shadow (under the overhang) and highlight area is not altered. The reproduction of the fine details (in the bricks) is enhanced as expected from the edge enhanced error diffusion algorithm.

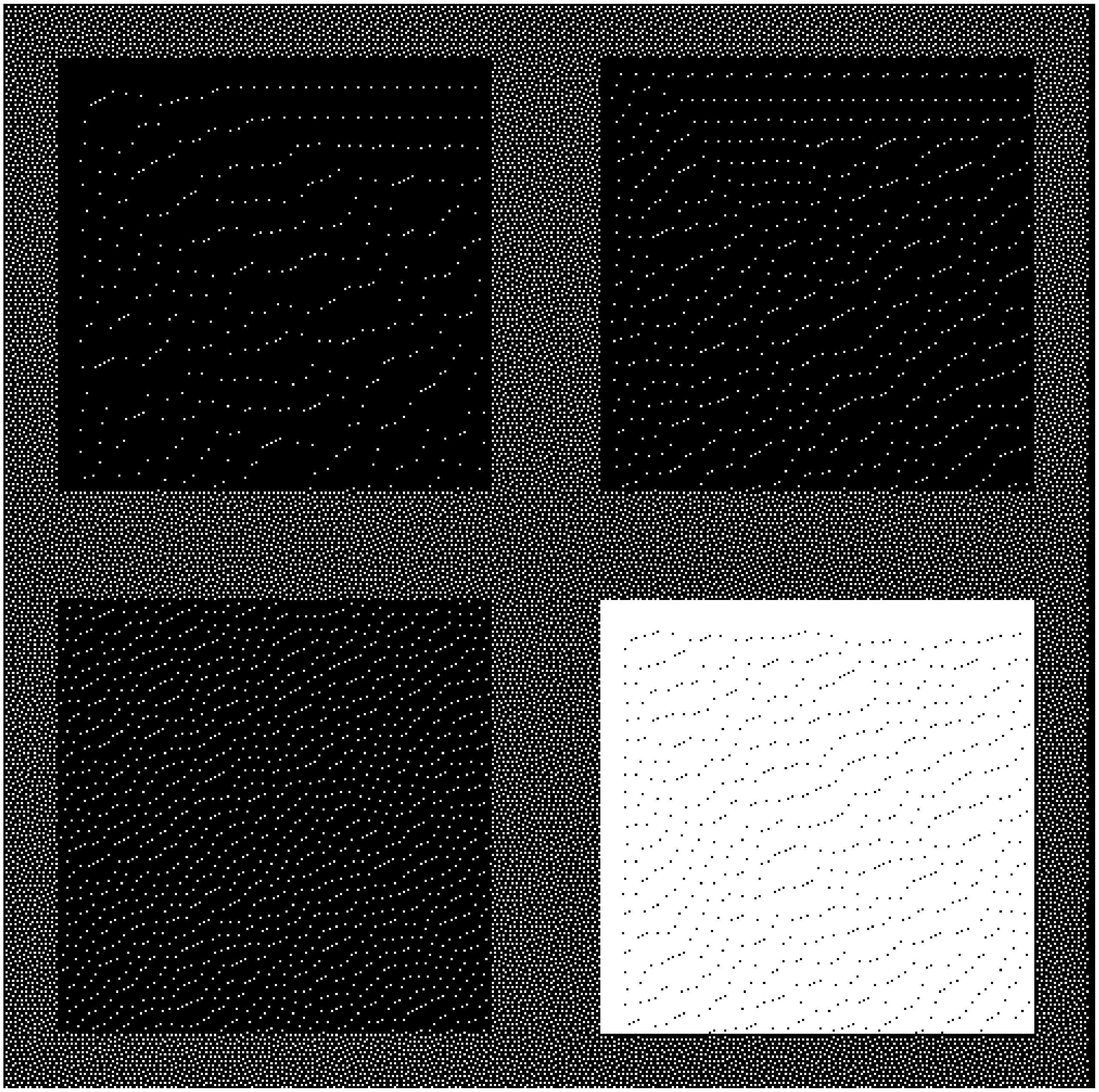

*Figure 8a. Result of applying the original error diffusion algorithm to a set of patches with graylevels “3”, “5”, “10”, and “250” from top-left to bottom right. The graylevel of the background area is “50”. The allowed input range is “0” to “255”.*

## 5. Conclusion

The shadow and highlight bahaviour of error diffusion is an important aspect in the reproduction of document images.

The threshold function in error diffusion can be a powerful tool to influence the local distribution of output pixels. In this paper, the threshold was modified using an output dependent threshold imprint to generate a more homogeneous response in the highlight and shadow areas as compared to standard error diffusion. The implementation of the proposed algorithm is straightforward and only modifies the binary decision making step in error diffusion. Error calculation and error forwarding are identical to the standard error diffusion algorithm.

The proposed threshold modification can be used in conjunction with other threshold modification, such as the edge enhanced error diffusion algorithm and its modifications, or output pattern suppression.

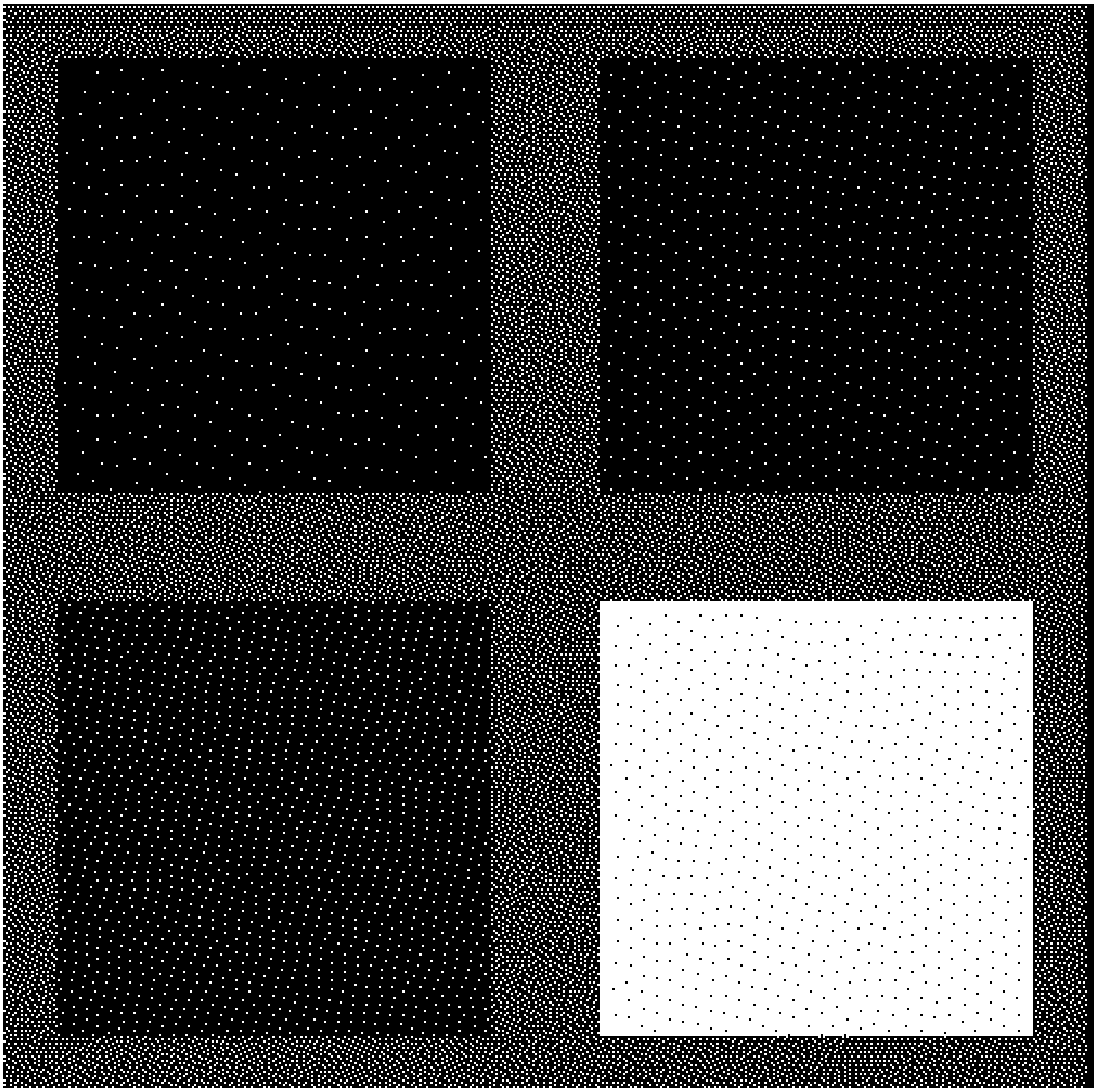

*Figure 8b. Result of applying the proposed algorithm to the same input data as used in Figures 8a.*

## References

1. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey-scale," *Proc. Soc. Inf. Disp.* **17**, 75–77 (1976).
2. C. Billotet-Hoffmann and O. Bryngdahl, "On the error diffusion technique for electronic halftoning," *Proc. Soc. Inf. Disp.* **24**, 253–258 (1983).
3. G. S. Fawcett and G. F. Schrack, "Halftone techniques using error correction," *Proc. Soc. Inf. Disp.*, **27**, 305–308 (1986).
4. J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Comp. Graphics and Image Process ing*, **5**, 13–40 (1976).
5. J. Sullivan, R. Miller and G. Pios, "Image halftoning us-

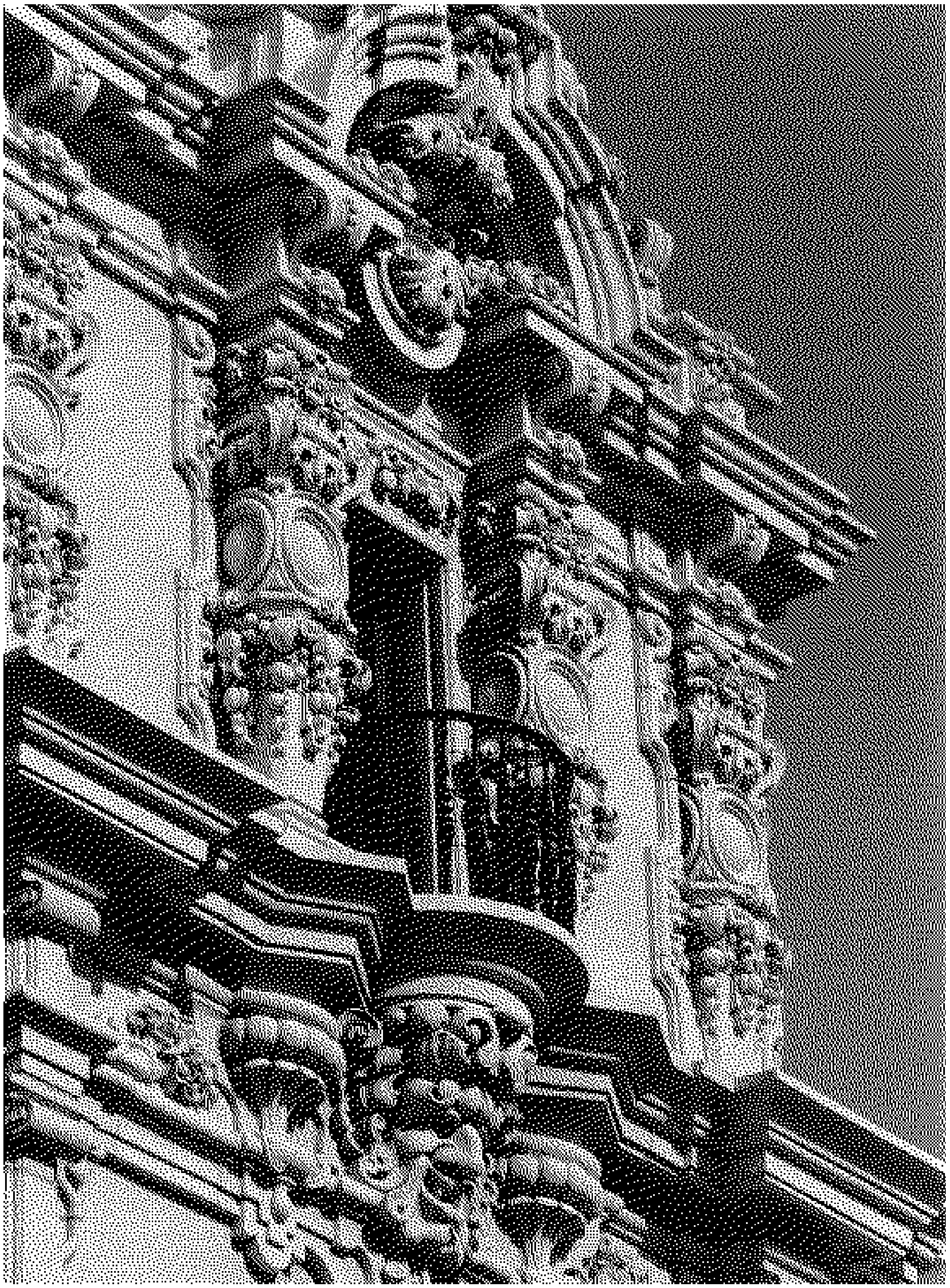

*Figure 9. Binarization using the standard error diffusion algorithm.*

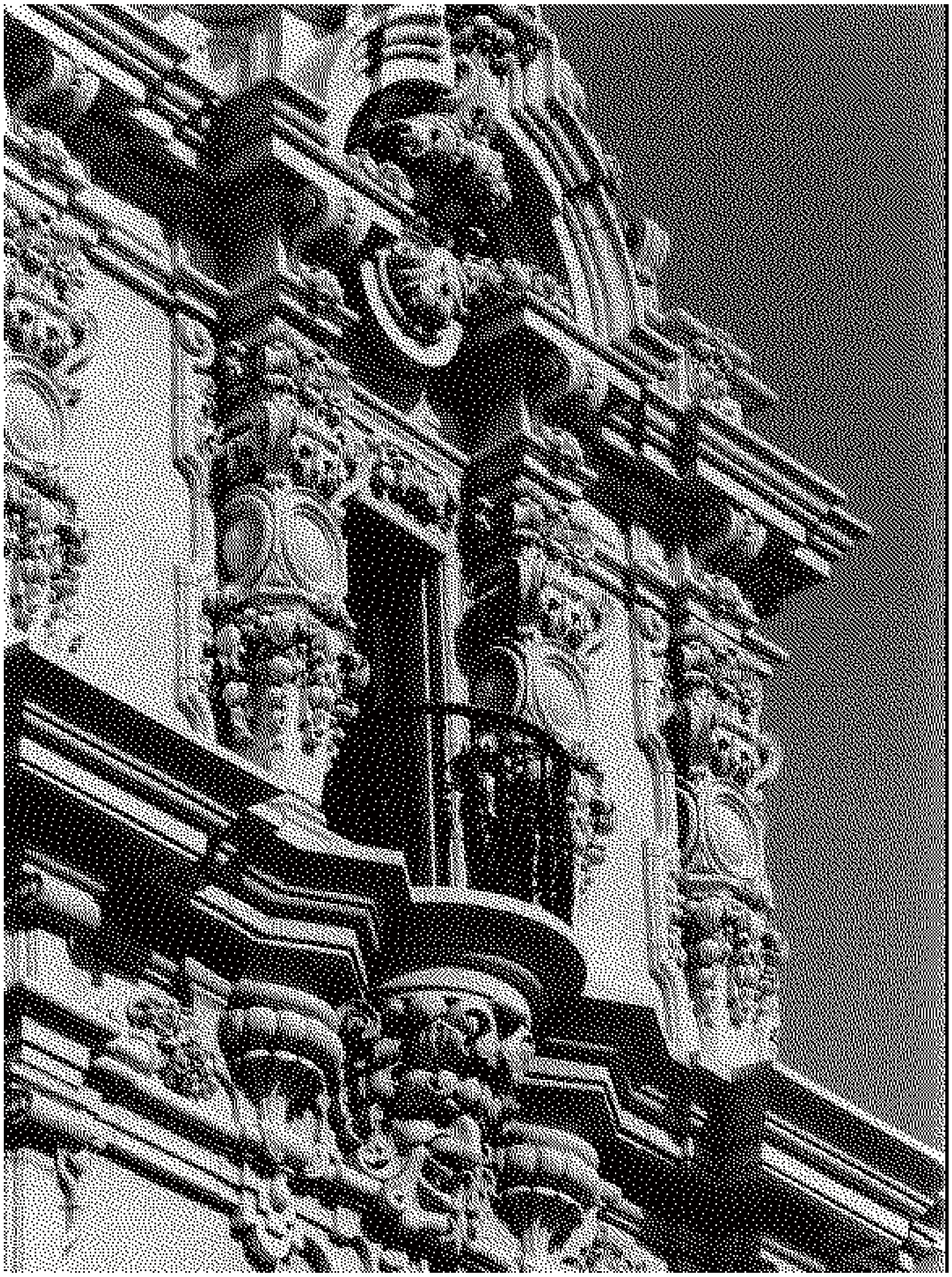

*Figure 10. Binarization using the proposed algorithm.*

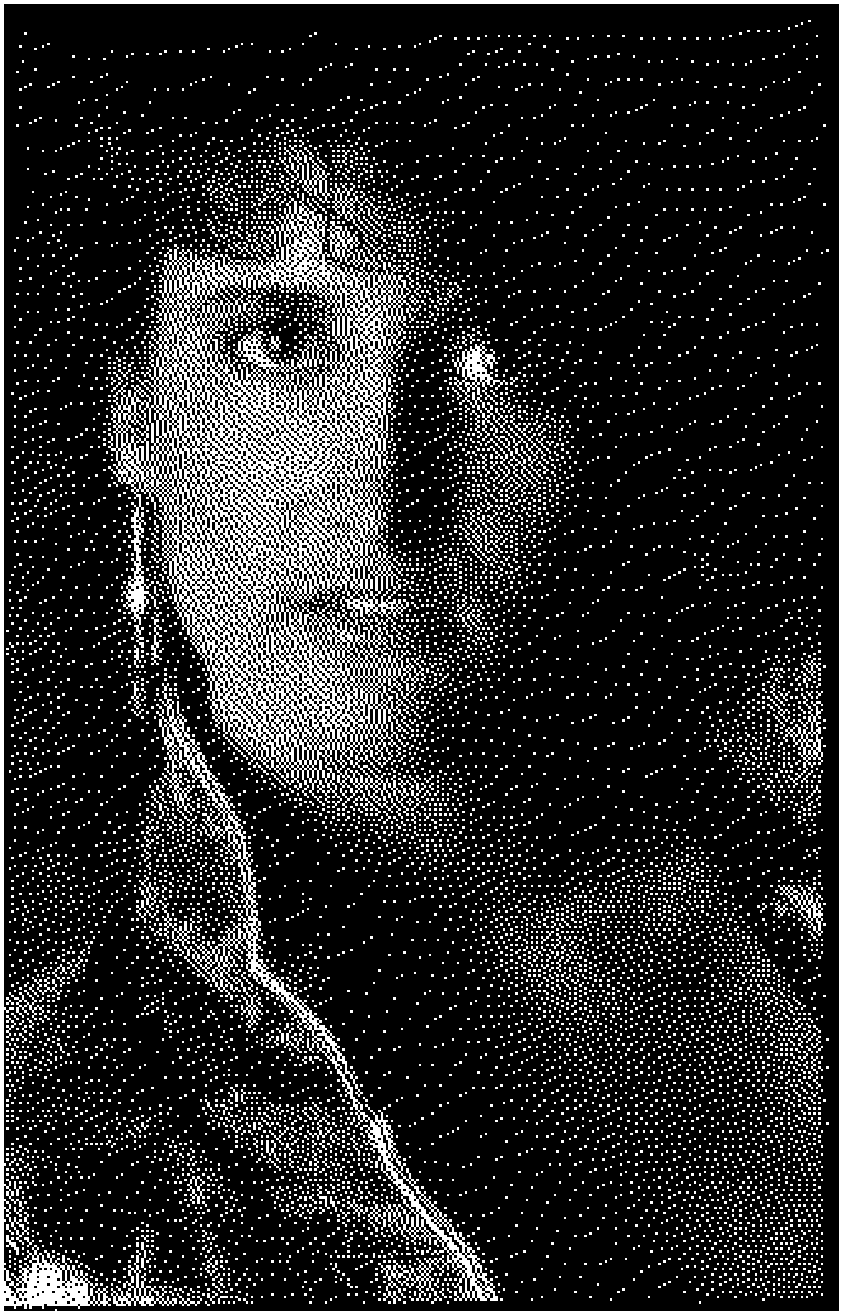

*Figure 11. Result of applying the standard error diffusion algorithm to an artistic image with large shadow areas.*

ing a visual model in error diffusion," *J. Opt. Soc. Am. A*, **10**, 1714–1724 (1993).

6. H. Bowers, "Error diffusion system", US Patent 5,130,823, 14 July 1992.
7. R. Levien, "Output dependent feedback in error diffusion halftoning," in *Recent Progress in Digital Halftoning*, Society for Imaging Science and Technology, 106-109, (1994).
8. T. N. Pappas, "Digital halftoning techniques for printing", in *Recent Progress in Digital Halftoning*, Society for Imaging Science and Technology, 42–45, (1994).
9. S.G. Wang, K.T. Knox and N. George, "Novel centering method for overlap correction in halftoning", in *Recent Progress in Digital Halftoning*, Society for Imaging Sci ence and Technology, 56–60, (1994).
10. R. Eschbach and K. T. Knox, "Error-diffusion algorithm

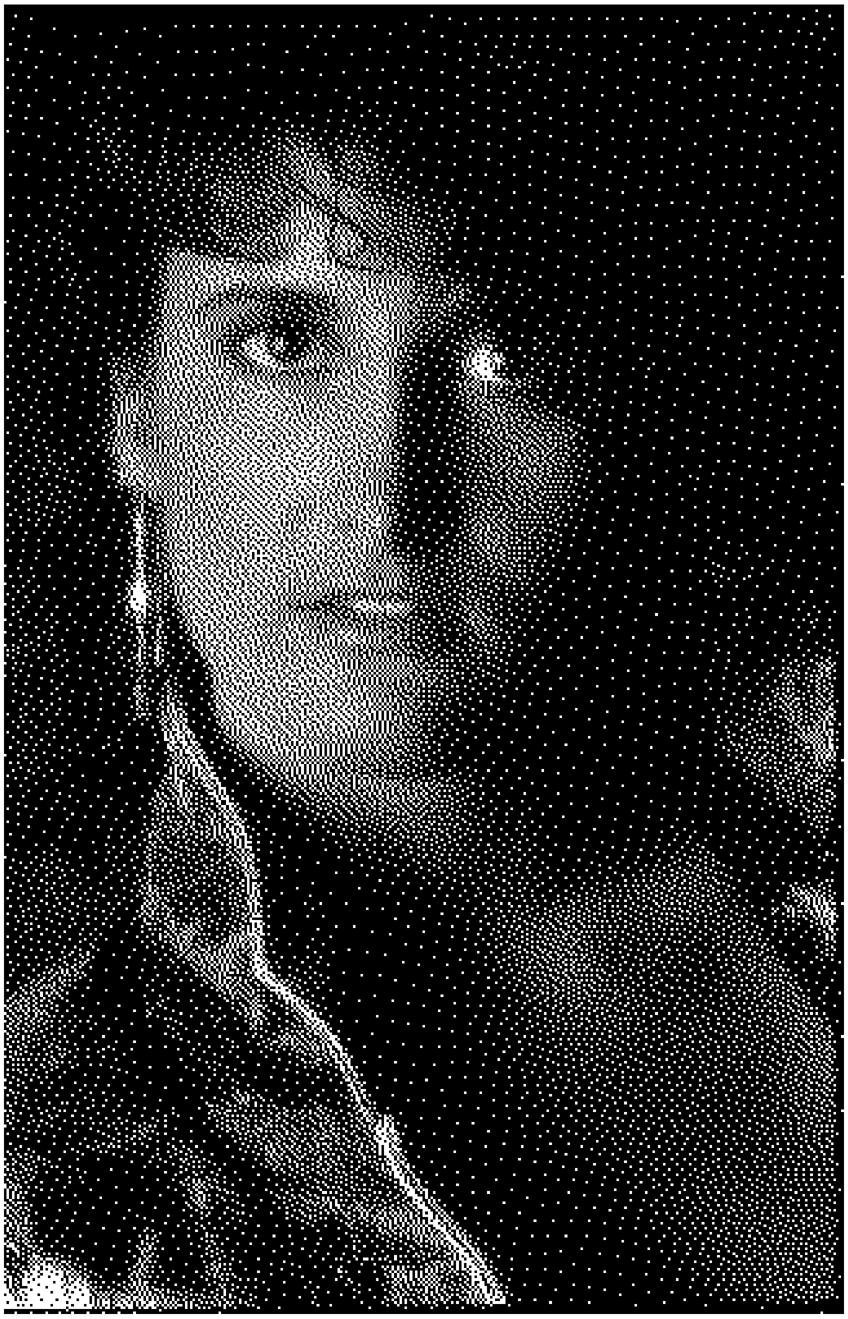

*Figure 12. Result of applying the proposed algorithm to the same input as used in Figure 11.*

with edge enhancement," *J. Opt. Soc. Am. A* **8**, 1844–1850 (1991).

11. *Recent Progress in Digital Halftoning*, Society for Imaging Science and Technology, (1994).
12. Z. Fan, "Analysis of error diffusion", *Proc. IS&T's 44th Annual Conference*, (1991).
13. R. Ulichney, "System for producing dithered images from continuous-tone image data", US Patent 4,955,065, 4 Sept. 1990.
14. G. Goertzel, and G. R. Thompson, "Sysstem for reproducing multi-level digital images on a bi-level printer of fixed dot size," US Patent 4,654,721, 31 Mar 1987.
15. K. T. Knox and R. Eschbach , "Threshold modulation in error diffusion", *Journal of Electronic Imaging*, **2**, (1993) 185–192.
16. R. Eschbach, "Reduction of artifacts in error diffusion

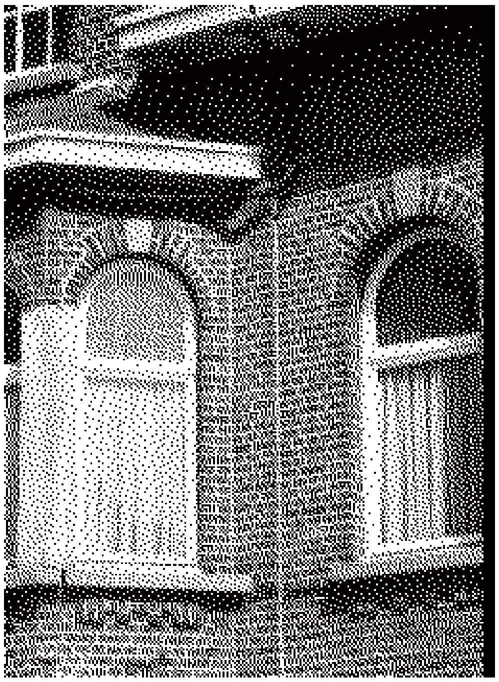

*Figure 13. The proposed algoithm without threshold modulation for edge enhancement.*

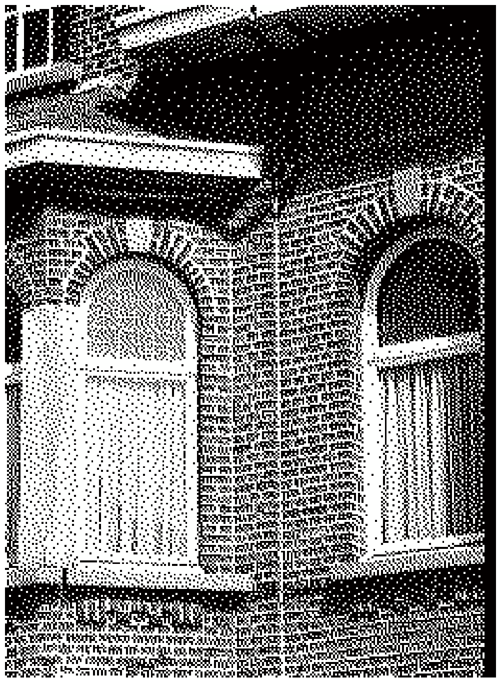

*Figure 14. The proposed algorithm with an additional edge enhancement by a additive input dependent threshold modulation.*

using input dependent weights", *Journal of Electronic Imaging,* **2**, 352–358 (1993).

17. J. A. G. Hale, "Dot spacing modulation for the production of pseudo grey pictures," *Proc. Soc. Inf. Disp.* **17**, 63–74 (1976).
18. M. Broja, R. Eschbach and O. Bryngdahl, "Stability of active binarization processes," *Optics Commun.* **60**, 353–358 (1985).
19. S. Weissbach and F. Wyrowski, "Numerical stability of the error diffusion concept", *Optics Commun.* **93**, 151–155 (1992).
20. S. Weissbach, F. Wyrowski and O. Bryngdahl, "Digital phase holograms: coding and quantization with an error diffusion concept," *Opt. Commun.* **72**, 37–41 (1989).
21. D. Birnbaum, R. Eschbach and K. T. Knox, "Under-compensated error diffusion for multi-level printing," *SPIE* Vol. **1912**, *Color Hard Copy and Graphics Art II,* 482–487 (1993).
22. R. Eschbach and Z. Fan, "Complex-valued error diffusion for off-axis computer-generated holograms," *Applied Optics* **32**, 3130–3136 (1993)
23. J.H. Kim, T.I. Chung, H.S. Kim, K.S. Son, and Y.S. Kim, "A new edge-enhanced error diffusion algorithm based on the error sum criterion", *Journal of Electronic Imaging*, **4**, 172-178 (1995); ***see pg. 105 this publication***.
24. C. J. Rosenberg, "Measurement based verification of an electrophotographic printer dot model for halftone algorithm tone correction", in *Recent Progress in Digital Halftoning*, Society for Imaging Science and Technology, 159–163, (1994).

❋ Previously published in the *Journal of Electronic Imaging,* **6**(2) pp. 348–356, 1997.