Efficient Document Rendering with Enhanced Run Length Encoding

Guotong Feng[a] and Charles A. Bouman[b]

[a] Ricoh Innovations Inc., Menlo Park, CA
[b] Purdue University, West Lafayette, IN

ABSTRACT

Document imaging and transmission systems (typically MFPs) require both effective and efficient image rendering methods that support standard data formats for a variety of document types, and allow for real time implementation. Since most conventional raster formats (e. g. TIFF, PDF, JPEG) are designed for use with either black and white text, or continuous-tone images, more specialized rendering methods are often required for representing mixed content documents. The baseline TIFF format supports a few binary compression options: PackBits, CCITT G3 and G4. Conventionally, halftoning algorithms, such as error diffusion, can be used to create a binary representation of a document image in the TIFF format. However, PackBits, CCITT G3 and G4 compression generally do not produce desired compression on halftone images.

In this paper, we propose an efficient error diffusion algorithm optimized for PackBits compression. This method, which we refer to as POED (PackBits optimized error diffusion), is a form of threshold modulation error diffusion which takes advantage of the byte-oriented run length structure of PackBits compression by encouraging repetition of bytes in the resulting binary image. To maintain the sharpness of text, a binary segmentation algorithm is provided to switch off the adaptive error diffusion procedure and switch on the Floyd Steinberg error diffusion procedure in text regions. The POED method with PackBits compression yields higher compression ratios than the conventional error diffusion method, while maintaining desirable visual quality with low computational and memory requirements. We show experimental results to compare our method with the Floyd Steinberg error diffusion method.

Keywords: Error diffusion, compression, segmentation, halftoning, run length encoding, compound documents

1. INTRODUCTION

Over the past few years, multifunction products (MFPs) have been increasingly used in many document imaging and distribution environments due to their low cost and high productivity. The typical document sending functions of MFPs such as scan-to-email, scan-to-fax and scan-to-folder are designed to automatically scan documents, convert them to desired file formats, and route them to desired destinations using existing communication protocols. These functions generally need to support both a variety of document types (e. g. text, mixed and picture) and standard data formats (e. g. TIFF, MTIFF, PDF and JPEG) with reasonably low memory usage and high processing speed.

Since most conventional raster formats are designed for use with either black and white text, or continuous-tone images,[1] more specialized rendering methods are often required for representing mixed content documents, which typically include a mixture of text, graphics, halftone regions and image data. TIFF is a widely used format[2] in MFPs for document transmission. However, the baseline TIFF format only supports a few binary compression options, CCITT G3/G4 (fax compression)[3,4] and PackBits compression, a simple byte-oriented run length scheme.[2] Conventional halftoning algorithms, such as error diffusion,[5] can be used to create a binary representation of a continuous-tone image. Such an image can be subsequently compressed using PackBits, G3 and G4 compression. However, CCITT G3/G4 is poorly suited to the compression of halftone images,[6] and particularly poorly suited to the compression of images halftoned using error diffusion. In fact, CCITT G3/G4 compression can actually expand the file size of error diffused halftones. This is problematic because error diffusion is a method for producing high quality binary representations when the sampling resolution is limited (e. g. 300 dpi). Alternatively, PackBits compression is a run length encoding algorithm which is more effective

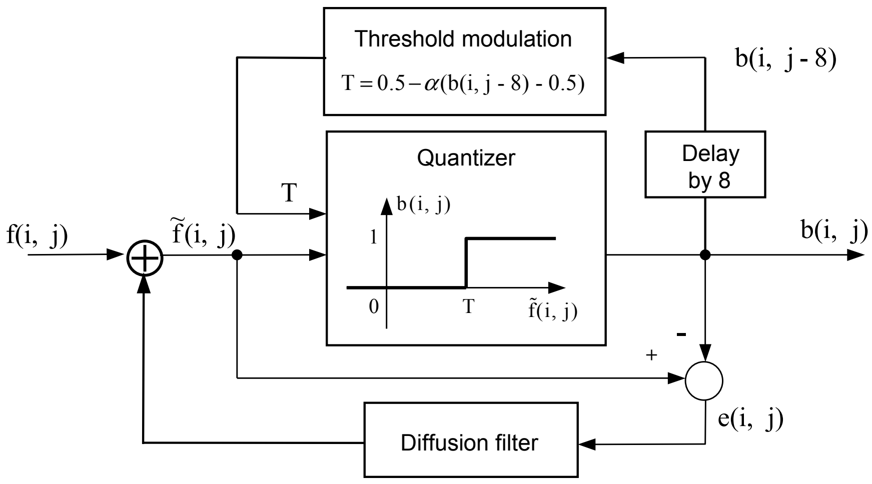


Figure 1. Illustration of threshold modulation error diffusion used in POED.

at compressing error diffused halftones than G3/G4. However, the size of PackBits compressed files is still larger than is desirable for fast document transmission.

In this paper, we propose a method called PackBits Optimized Error Diffusion (POED) for producing efficient binary representations of documents with enhanced run length encoding efficiency. The POED method is a modified form of error diffusion which is designed to produce halftones that can be more efficiently compressed using PackBits compression, but preserve much of the desirable quality of conventional error diffusion. This approach is particularly suitable for documents that are scanned and transmitted as Email attachments.

The PackBits compression is a simple byte-oriented run-length compression scheme in which all repeated bytes are encoded as one or more replicate runs. The POED method includes an adaptive error diffusion algorithm which takes advantage of the byte-oriented run-length structure of PackBits to encourage repetition of bytes in the resulting binary image. To maintain the text quality, a simple binary segmentation algorithm[7] is combined with the error diffusion algorithm to control the separate processing of text regions and background regions. Both the error diffusion and segmentation algorithm can be optimized to produce the best rate-distortion tradeoff.

The rest of the paper is organized as follows. In Section 2, a threshold modulation error diffusion algorithm is proposed. Section 3 describes the entire POED procedure that combines the binary segmentation algorithm and the threshold modulation error diffusion algorithm. In Section 4, a parameter optimization procedure is developed. In Section 5, the experimental results of the proposed POED method are shown and compared with the conventional error diffusion method. Finally, Section 6 summarizes the conclusions of the paper.

2. THRESHOLD MODULATION ERROR DIFFUSION

Figure 1 illustrates a threshold modulation error diffusion algorithm that is the critical part of the PackBits Optimized Error Diffusion method. Notice that the figure without the threshold modulation module is exactly the diagram of conventional error diffusion. In order to enforce a periodic pattern in the binary output, the threshold modulation technique is applied to the conventional error diffusion algorithm. In particular, the quantizing threshold of each pixel is modulated by the binary output that occurred 8 pixels earlier in the scan order of the error diffusion. The 8 pixel delay corresponds to the 8 bits in a single byte, so this adaptive threshold modulation results in byte sequences that tend to repeat.

Note that in Fig. 1, $f(i,j)$ is the normalized pixel value of the grayscale image input, and $b(i,j)$ is the pixel value of the binary image output. $b(i,j-8)$ is the value of the binary output that occurred 8 pixels earlier in the scan order. The threshold modulation is then given by $T = 0.5 - \alpha(b(i,j) - 0.5)$ where α is a parameter that is in $[0,1]$. In particular, if $b(i,j-8) = 1$, then the threshold T is less than 0.5, and if $b(i,j-8) = 0$, then the threshold is greater than 0.5. Intuitively, a smaller threshold encourages a "1" value at the output of quantizer, while a larger threshold encourages a "0" value at the output. So $b(i, j)$ tends to repeat the value of $b(i, j - 8)$.

Fig. 2 shows an example result for comparison of the threshold modulation error diffusion and the conventional error diffusion. As shown in the example, the threshold modulation error diffusion algorithm creates a periodic pattern in the gray level background. This typically reduces the PackBits compressed image size over conventional Floyd Steinberg error diffusion, while maintaining decent overall visual quality. Notice that in the text regions of the example, the threshold modulation error diffusion process is switched off and the Floyd Steinberg error diffusion process is switched on. This separate processing is done by using a binary segmentation bitmap, which we will describe in the next section.

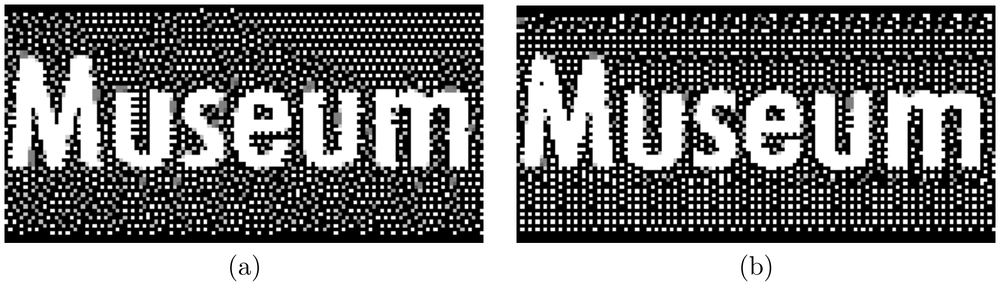


Figure 2. Example of threshold modulation error diffusion. (a) Haltoned by Floyd Steinberg error diffusion; (b) Halftoned by threshold modulation error diffusion.

3. PACKBITS OPTIMIZED ERROR DIFFUSION

While the threshold modulation error diffusion algorithm performs well in the background region, it can also produce some undesirable noise around the text edges. Consequently, it is necessary to switch off the threshold modulation error diffusion in the text regions, and to switch on the conventional error diffusion in these regions. For this purpose, we apply a simple binary segmentation algorithm to control the switching process. Fig. 3 illustrates the overall structure of the PackBits optimized error diffusion algorithm. The binary segmentation algorithm separates the input image into two regions, text and non-text. In the text region, The Floyd Steinberg error diffusion coefficients are applied and the threshold modulation is turned off, while in the non-text region, the threshold modulation is turned on and a different set of error diffusion coefficients (w_0, w_1, w_2 and w_3) are applied. All these weights are optimized for the best tradeoff of bit rate and distortion. In addition, the combined weights of both error diffusion processes are adjusted by a parameter β in $[0, 1]$, which is also optimized.

Fig. 4 shows how the binary segmentation algorithm works. The input grayscale image is first smoothed by using a 5×5 Gaussian low pass filter with the standard deviation σ to remove noise. A local activity measure is then computed for the filtered output using the following formula.

$$l(i,j) = \sqrt{\frac{1}{8}\sum_{m=-1}^{1}\sum_{n=-1}^{1}|g(i,j) - g(i-m, j-n)|^2}\ , \tag{1}$$

where $g(i, j)$ is the filtered output value at pixel (i, j) from the Gaussian low pass filter. A binary threshold block compares the value of $l(i, j)$ to a predefined threshold, T_b, at each pixel and creates the binary segmentation map. If the local difference is larger than the threshold, the output is set to "1", and the pixel is classified as non-text; otherwise, the output is set to "0" and the pixel is classified as text.

4. PARAMETER OPTIMIZATION

The POED method is optimized by searching for the value of the vector $\theta = [\beta, w_0, w_1, w_2, w_3]$ where each element falls in $[0, 1]$ and the value of α is fixed. Note that the four error diffusion weights are also normalized

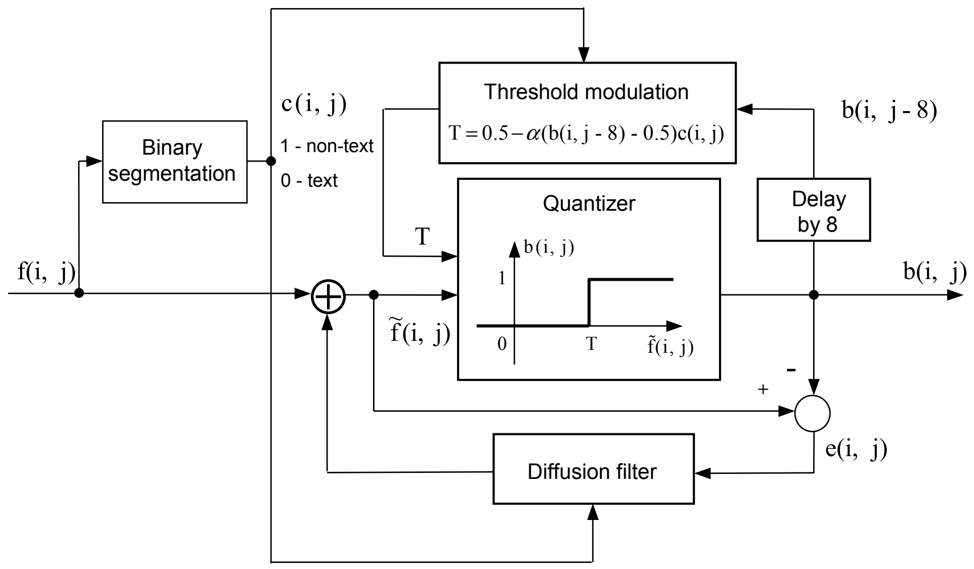


Figure 3. Overall structure of PackBits Optimized Error Diffusion (POED).

to sum to 1 before they are used in the error diffusion algorithm. The search strategy is designed to minimize the cost function defined as:

$$C = B + \lambda D \ , \tag{2}$$

where B is the PackBits compressed file size in Kbytes, D is the distortion measure in mean square error, and λ is a predefined weight that controls the balance between B and D. We used 30 as the value of λ in all our experiments. The distortion measure D is computed as follows. First, an error image is taken by comparing the grayscale input image and the halftoned image. Next, the error image is processed by using a 5×5 Gaussian low pass filter. Finally, the mean square error is computed by taking the average of the squared values of all the pixels in the filtered error image.

The optimization of θ is done by sequentially perturbing the elements of the vector θ using an iterative coordinate decent (ICD) strategy. The starting point is always set at $[1.0, 0.4375, 0.0625, 0.3125, 0.1875]$ where the four error diffusion weights are exactly the Floyd Steinberg error diffusion coefficients. Each perturbation is made using a step size of $\pm\delta$ which is initialized to the value $\delta = 0.2$. The decision to perturb each component by $\pm\delta$ or to leave it unchanged is made based on the value of the mean square error computed for the documents in the training set. If the mean squared error does not decrease with the perturbation of every element of the parameter vector, then the size of the perturbation is automatically reduced using the formula $\delta \leftarrow \delta/2$. Coordinate descent optimization is stopped when the perturbation value is less than 0.02.

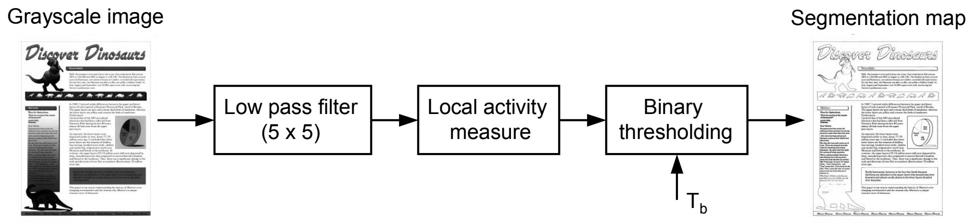
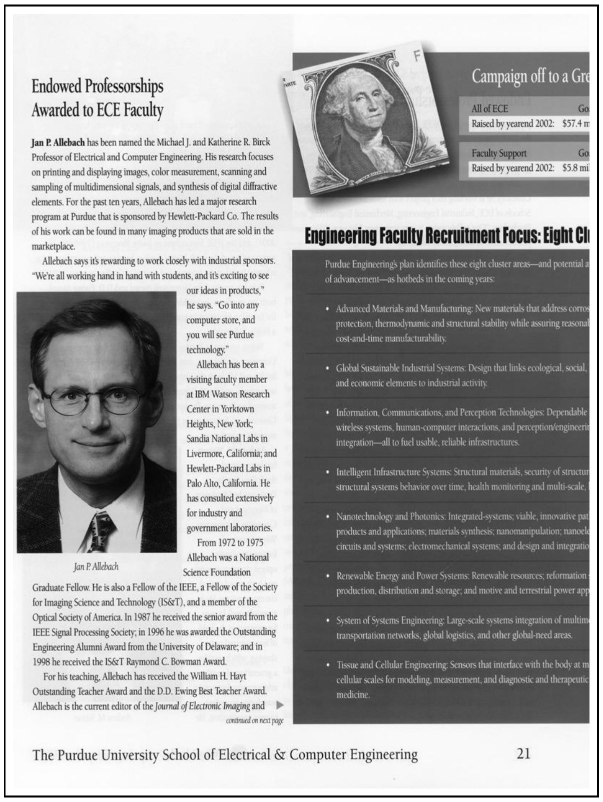


Figure 4. Diagram of binary segmentation used in POED.



Endowed Professorships
Awarded to ECE Faculty

Jan P. Allebach has been named the Michael J. and Katherine R. Birck Professor of Electrical and Computer Engineering. His research focuses on printing and displaying images, color measurement, scanning and sampling of multidimensional signals, and synthesis of digital diffractive elements. For the past ten years, Allebach has led a major research program at Purdue that is sponsored by Hewlett-Packard Co. The results of his work can be found in many imaging products that are sold in the marketplace.

Allebach says it's rewarding to work closely with industrial sponsors. "We're all working hand in hand with students, and it's exciting to see our ideas in products," he says. "Go into any computer store, and you will see Purdue technology."

Allebach has been a visiting faculty member at IBM Watson Research Center in Yorktown Heights, New York; Sandia National Labs in Livermore, California; and Hewlett-Packard Labs in Palo Alto, California. He has consulted extensively for industry and government laboratories.

Jan P. Allebach

From 1972 to 1975 Allebach was a National Science Foundation Graduate Fellow. He is also a Fellow of the IEEE, a Fellow of the Society for Imaging Science and Technology (IS&T), and a member of the Optical Society of America. In 1987 he received the senior award from the IEEE Signal Processing Society; in 1996 he was awarded the Outstanding Engineering Alumni Award from the University of Delaware; and in 1998 he received the IS&T Raymond C. Bowman Award.

For his teaching, Allebach has received the William H. Hayt Outstanding Teacher Award and the D.D. Ewing Best Teacher Award. Allebach is the current editor of the *Journal of Electronic Imaging* and ▶

continued on next page

Campaign off to a Gre

All of ECE	Go
Raised by yearend 2002:	$57.4 m
Faculty Support	Go
Raised by yearend 2002:	$5.8 mi

Engineering Faculty Recruitment Focus: Eight Cl

Purdue Engineering's plan identifies these eight cluster areas—and potential a of advancement—as hotbeds in the coming years:

- Advanced Materials and Manufacturing: New materials that address corros protection, thermodynamic and structural stability while assuring reasonal cost-and-time manufacturability.
- Global Sustainable Industrial Systems: Design that links ecological, social, and economic elements to industrial activity.
- Information, Communications, and Perception Technologies: Dependable wireless systems, human-computer interactions, and perception/engineerin integration—all to fuel usable, reliable infrastructures.
- Intelligent Infrastructure Systems: Structural materials, security of structur structural systems behavior over time, health monitoring and multi-scale,
- Nanotechnology and Photonics: Integrated-systems; viable, innovative pat products and applications; materials synthesis; nanomanipulation; nanoel circuits and systems; electromechanical systems; and design and integratio
- Renewable Energy and Power Systems: Renewable resources; reformation production, distribution and storage; and motive and terrestrial power app
- System of Systems Engineering: Large-scale systems integration of multim transportation networks, global logistics, and other global-need areas.
- Tissue and Cellular Engineering: Sensors that interface with the body at m cellular scales for modeling, measurement, and diagnostic and therapeutic medicine.

The Purdue University School of Electrical & Computer Engineering 21

Figure 5. Test image scanned at 300dpi with size of 2399 × 3224.

5. EXPERIMENTAL RESULTS

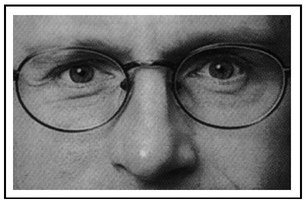
In this experiment, we used a scanned document as the test image to evaluate the performance of the POED method by comparing it to the conventional error diffusion method. The test image is shown in Fig. 5. We used another scanned document as the training image for parameter optimization. Both images were scanned at 300 dpi and 24 bits per pixel (bpp), and stored in TIFF format. The grayscale image of each document was obtained by taking the grayscale mode of the color image in Adobe Photoshop.

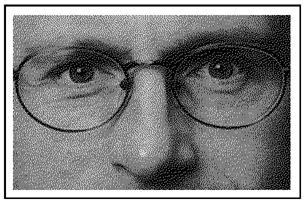
Table 1 lists all the parameters that are used by the POED algorithm. We choose these parameters because we found that they worked well for a wide variety of documents. Notice that the four error diffusion weights are slightly different at different α and β values, and are close to the standard Floyd Steinberg error diffusion weights.

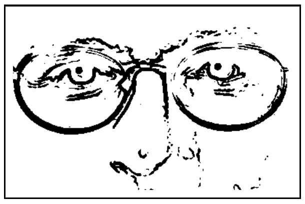
Table 1. Summary of parameters used in POED.

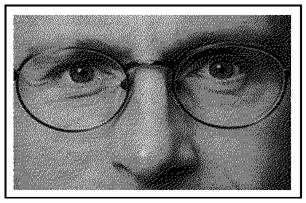
Parameter	σ			T_b		
Value	1.5			0.025		
Parameter	α	β	w_0	w_1	w_2	w_3
Value	0.1	0.9000	0.4889	0.0688	0.2973	0.1450
Value	0.2	0.8800	0.4799	0.0737	0.2790	0.1674
Value	0.4	0.9300	0.5058	0.0954	0.1936	0.2052
Value	0.6	0.9250	0.4403	0.0622	0.3109	0.1866

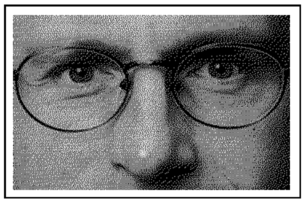
Fig. 6 shows a comparison of the standard Floyd Steinberg error diffusion method and the proposed PackBits optimized error diffusion method in terms of the PackBits compression performance. Fig. 6(c) shows the binary segmentation result created in the POED process (in Fig. 4). The POED results in Fig. 6(d), Fig. 6(e), Fig. 6(f) and Fig. 6(g) show decreasing overall quality and increasing PackBits compression ratios as the value of α is increased. Compared to the Floyd Steinberg error diffusion result in Fig. 6(b), the POED result in Fig. 6(f) with $\alpha = 0.4$ achieves a much higher compression ratio while still maintaining good visual quality.

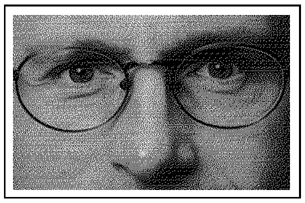

(a) Original grayscale


(b) Floyd Steinberg


(c) Binary segmentation


(d) POED ($\alpha = 0.1$)


(e) POED ($\alpha = 0.2$)


(f) POED ($\alpha = 0.4$)

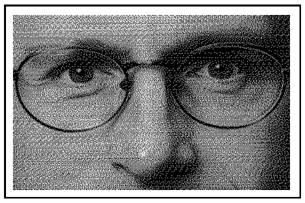

(g) POED ($\alpha = 0.6$)

Figure 6. Comparison of Floyd Steinberg error diffusion and PackBits optimized error diffusion: (a) Original grayscale portion A (uncompressed binary image file size 967 Kbytes); (b) Floyd Steinberg error diffusion result (PackBits compressed size 894 Kbytes, compression ratio 1.08:1); (c) Binary segmentation map created in POED; (d) POED result ($\alpha = 0.1$, PackBits compressed size 661 Kbytes, compression ratio 1.46:1); (e) POED result ($\alpha = 0.2$, PackBits compressed size 527 Kbytes, compression ratio 1.83:1); (f) POED result ($\alpha = 0.4$, PackBits compressed size 419 Kbytes, compression ratio 2.31:1); (g) POED result ($\alpha = 0.6$, PackBits compressed size 381 Kbytes, compression ratio 2.54:1).

Fig. 7 shows another comparison of the standard Floyd Steinberg error diffusion method and the POED method for the same test image. The results in this figure show similar performance to Fig. 6. Furthermore, the text regions in all the POED results retain sharpness that is very close to the Floyd Steinberg error diffusion result. This is just because the binary segmentation algorithm provides an adaptive control on the error diffusion processes at a pixel level. Notice that the binary segmentation map in Fig. 7(c) contains larger strokes which are used to protect the text from being damaged by the threshold modulation error diffusion process.

Fig. 8 shows one more comparison of the standard Floyd Steinberg error diffusion method and the POED method for the same test image. The results in this figure are consistent with those in Fig. 6 and Fig. 7.

6. CONCLUSIONS

We have proposed a binary representation method for scanned documents that is suitable for the baseline TIFF PackBits compression. This method includes an efficient threshold modulation algorithm that encourages a periodic pattern in the binary output. The method applies a simple binary segmentation algorithm to control the error diffusion processes in order to maintain the sharpness of text. The POED method with the PackBits compression yields significant higher compression ratios than the conventional error diffusion method, while maintaining desirable visual quality with low computational and memory requirements.

ACKNOWLEDGMENTS

This work was supported by Hewlett-Packard.

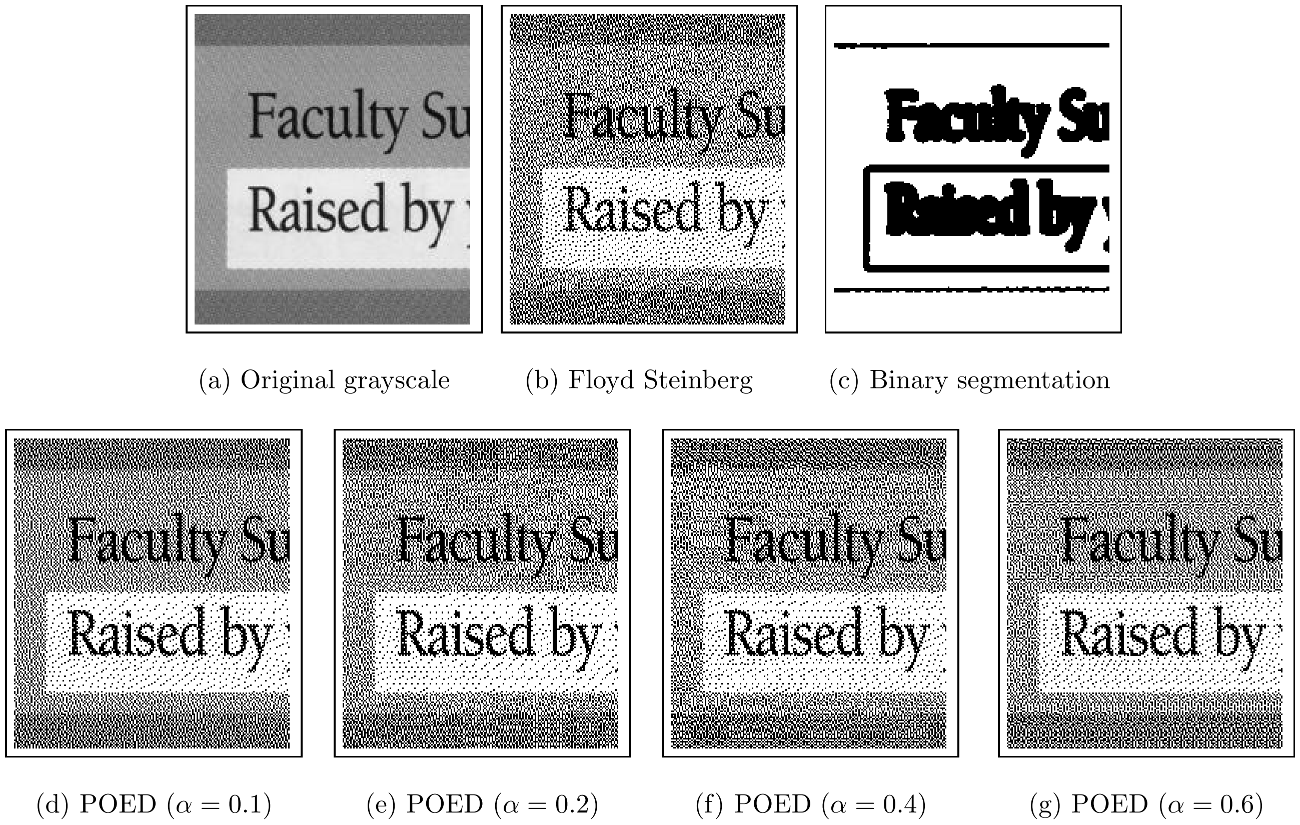
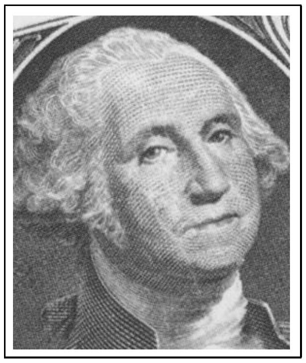

(a) Original grayscale
(b) Floyd Steinberg
(c) Binary segmentation
(d) POED ($\alpha = 0.1$)
(e) POED ($\alpha = 0.2$)
(f) POED ($\alpha = 0.4$)
(g) POED ($\alpha = 0.6$)

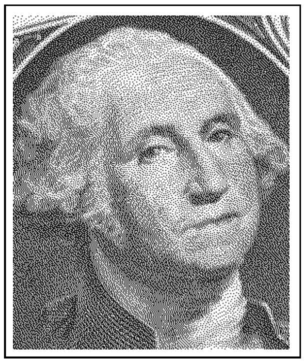
Figure 7. Comparison of Floyd Steinberg error diffusion and PackBits optimized error diffusion: (a) Original grayscale portion B (uncompressed binary image file size 967 Kbytes); (b) Floyd Steinberg error diffusion result (PackBits compressed size 894 Kbytes, compression ratio 1.08:1); (c) Binary segmentation map created in POED; (d) POED result ($\alpha = 0.1$, PackBits compressed size 661 Kbytes, compression ratio 1.46:1); (e) POED result ($\alpha = 0.2$, PackBits compressed size 527 Kbytes, compression ratio 1.83:1); (f) POED result ($\alpha = 0.4$, PackBits compressed size 419 Kbytes, compression ratio 2.31:1); (g) POED result ($\alpha = 0.6$, PackBits compressed size 381 Kbytes, compression ratio 2.54:1).

REFERENCES

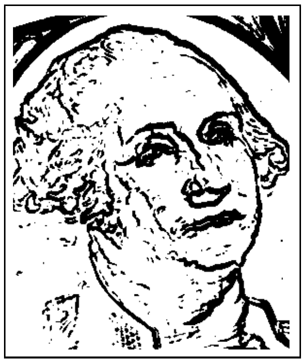
1. G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electronics* **38**(1), pp. 18–34, 1992.
2. "TIFF Revision 6.0," *Aldus Corporation, Seattle, WA* , June 3, 1992.
3. CCITT Recommendation T.4, "Standardization of Group 3 facsimile apparatus for document transmission," **VII-Fascicle VII.3**, pp. 21–47.
4. CCITT Recommendation T.6, "Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus," **VII-Fascicle VII.3**, pp. 48–57.
5. R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial greyscale," *Journal of the Society for Information Display* **17**(2), pp. 75–77, 1976.
6. S. J. Urban, "Review of standards for electronic imaging for facsimile systems," *Journal of Electronic Imaging* **1**, pp. 5–21, Jan. 1992.
7. G. Feng, M. G. Fuchs, and C. A. Bouman, "Image rendering for digital fax," in *Proc. of the SPIE/IS&T Conference on* Color Imaging: Device-Independent Color, Color Hardcopy, and Applications VII, pp. 504–512, (Santa Clara, CA), January 2003.



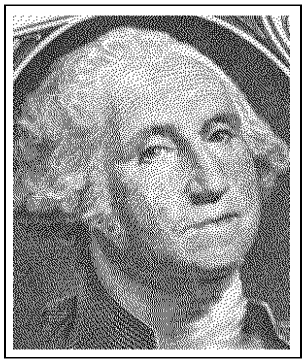
(a) Original grayscale



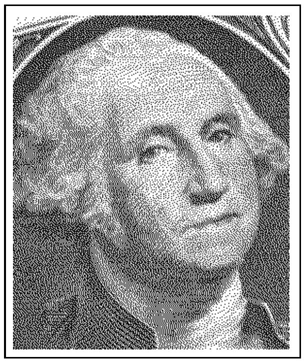
(b) Floyd Steinberg



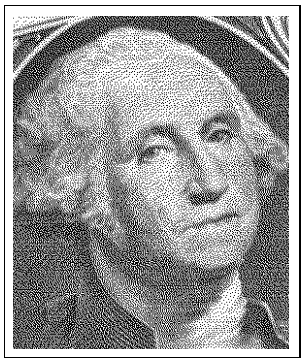
(c) Binary segmentation



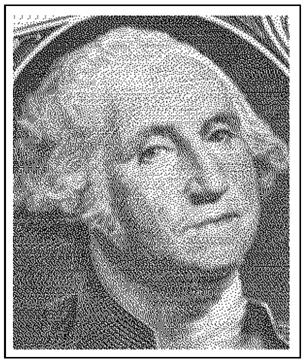
(d) POED ($\alpha = 0.1$)



(e) POED ($\alpha = 0.2$)



(f) POED ($\alpha = 0.4$)



(g) POED ($\alpha = 0.6$)

Figure 8. Comparison of Floyd Steinberg error diffusion and PackBits optimized error diffusion: (a) Original grayscale portion C (uncompressed binary image file size 967 Kbytes); (b) Floyd Steinberg error diffusion result (PackBits compressed size 894 Kbytes, compression ratio 1.08:1); (c) Binary segmentation map created in POED; (d) POED result ($\alpha = 0.1$, PackBits compressed size 661 Kbytes, compression ratio 1.46:1); (e) POED result ($\alpha = 0.2$, PackBits compressed size 527 Kbytes, compression ratio 1.83:1); (f) POED result ($\alpha = 0.4$, PackBits compressed size 419 Kbytes, compression ratio 2.31:1); (g) POED result ($\alpha = 0.6$, PackBits compressed size 381 Kbytes, compression ratio 2.54:1).