Modeling and Simulation of a JBIG2 Compliant Color Printer Pipeline

Niranjan Damera-Venkata and Chase Krumpelman

Abstract

The objective of this work is to develop a design tool for the simulation and testing of a color image printing pipeline. True color images typically have three color planes (red, green and blue) with eight bits of resolution per pixel per image plane. Printing these images involves a binarization process called halftoning which converts each color plane into a binary image for rendering on binary devices such as printers. While the halftoning algorithm is the heart of the printer pipeline, several preprocessing methods are often required to perform basic image manipulations. These include printer dependent color transformations, image scaling/resizing, and image compression. Thus images at various stages in the printer pipeline are subjected to various types of distortions. Simulation of the printer pipeline is of key importance for end to end performance optimization. We propose to model the printer pipeline using dataflow models of computation in the Ptolemy design environment. Further, we plan to simulate typical tradeoffs in a color printer pipeline using JBIG2 compliant image compression.

Index terms– image halftoning, color imaging, JBIG2, printer pipeline

The authors are with the Center of Vision and Image Sciences, Dept. of Electrical and Computer Eng., The University of Texas at Austin, Austin, TX 78712-1084. E-mail: {damera-v,krumpelm}@ece.utexas.edu.

I. Introduction

Digital halftoning is the process of representing continuous-tone (a.k.a grayscale/color) images with a finite number of levels, for the purpose of display on devices with finite reproduction pallettes. The resulting images are called halftones. In most printers, the printer is able to either place a dot on the paper, or not place a dot. This means that the printer is a binary device capable of reproducing only two levels where the presence of a dot on the paper may be indicated by the level -1, and the absence of a dot may be indicated by the level $+1$. In other applications, such as display on monitors, the levels available are usually more than two, but finite. In all cases, the goal of digital halftoning is to produce, via a clever distribution of dots, the illusion of continuous tone.

A practical printer or fascimile pipeline includes several preprocessing and postprocessing methods such as printer dependent color transformations [1], image scaling/resizing algorithms [2], and image compression algorithms such as JPEG or JBIG [2], apart from the halftoning algorithm. Recent standardization efforts for bi-level images and text compression have led to the new JBIG2 standard [3], [4] as the protocol for halftone and fascimile compression. Fig. 1(a) and (b) show a typical color printer pipeline using JBIG2-compliant compression.

The encoder part of the pipeline, shown in Fig. 1(a) halftones the image after resizing to a required output resolution. Color printers typically use a printer dependent color space [1] to represent color images. The main components of the encoder stage of the printer pipeline are the halftoning algorithm and the JBIG2 halftone coder. The halftoning algorithm is printer dependent. For the purpose of this work we will use vector color error diffusion [5], [6] as the halftoning algorithm. The JBIG2 halftone coder converts the halftone into a JBIG2 compliant bitstream. The JBIG2 standard offers a lot of freedom in designing the encoder [3]. Some of the previously reported JBIG2-compliant halftone coders include [4] and [7]. The decoder, shown in Fig. 1(b) simply consists of a JBIG2 decoder (which is typically arithmetic decoding followed by table lookup), an image resizing operation and the printer's local halftoning engine.

The end to end simulation of the printer pipeline is of key importance in designing JBIG2 encoders [8] and halftoning algorithms. Typical tradeoffs such as rate vs. visual distortion and noise vs. frequency distortion can then be explored via simulation [8]. The principal goal of this work is to develop an end to end simulation of the printer pipeline.

II. Halftoning Algorithms

A. Common halftoning methods

Commonly used halftoning methods fall into the following three categories.

1. **Clustered dot dither:** The continuous tone image is quantized by a periodic array of thresholds. The thresholds are organized to promote the formation of dot-clusters, which are robust to ink spread in printing. Newspaper printing and many inkjet printers use clustered dot dither.
2. **Dispersed dot dither:** The continuous tone image is quantized by a periodic array of thresholds, but with the goal of keeping the dots from clustering. This method leads to better halftones when dots are precisely rendered, e.g. on monochrome monitors. An analysis of clustered and dispersed dot dither halftoning is presented in [9].
3. **Error diffusion:** Error diffusion [10] developed by Floyd and Steinberg diffuses the quantization error over the neighboring continuous-tone pixels. That is, the image would be scanned, and the unprocessed pixels in a local neighborhood, around the current pixel, would be modified. Anastassiou [11] and Bernard [12] showed that error diffusion halftoning is really two-dimensional delta-sigma modulation. Delta-sigma modulation is a popular method for A/D conversion in digital audio that also employs feedback. Error diffusion feeds back a filtered version of the quantization error to the input. The design of the error filter is the key to high quality error diffusion halftoning methods. Error diffusion halftones have significantly improved quality over the clustered and dispersed dot dither, because it is free from periodic artifacts and shapes the quantization noise to the high frequencies where the human eye is least sensitive. High frequency noise is referred to as "blue-noise", and hence, error diffusion is an example of blue noise halftoning [13]. Error diffusion is used in high quality inkjet printers and displays.

B. Vector Color Error Diffusion

Akarun, Yardimici and Cetin [5] extended grayscale error diffusion to color images, which we refer to as vector color error diffusion. Our pipeline will use the fixed filter version of vector color error diffusion [6]. Fig. 2 shows a block diagram of vector color error diffusion halftoning. If we consider halftoning red-green-blue (RGB) images, then the quantizer output for each color channel at any pixel is exactly one element from the discrete set $\mathcal{O} = \{-1, 1\}$. Here -1 represents a red, green or blue dot, depending on the color channel, whereas 1 represents the absence of a dot for that color channel.

The quantizer $\mathbf{Q}(\cdot)$ is defined by

$$\mathbf{Q}(\vec{\mathbf{u}}) = \begin{pmatrix} Q(u_1) \\ Q(u_2) \\ Q(u_3) \end{pmatrix} \tag{1}$$

$$Q(u_i) = \begin{cases} 1 & u_i \geq 0 \\ -1 & u_i < 0 \end{cases} \tag{2}$$

where $\vec{\mathbf{u}}$ is a column vector and u_i, $i = 1, 2, 3$, represents the red, green and blue components of the color vector to be quantized.

The filter in the feedback loop has matrix valued coefficients. The filter operates on the quantization error sequence $\vec{\mathbf{e}}(\vec{\mathbf{m}})$ to produce the feedback signal sequence according to

$$\mathcal{H}\vec{\mathbf{e}}(\vec{\mathbf{m}}) = \sum_{\vec{\mathbf{k}} \in \mathcal{S}} \tilde{\mathbf{H}}(\vec{\mathbf{k}})\vec{\mathbf{e}}(\vec{\mathbf{m}} - \vec{\mathbf{k}}) \tag{3}$$

where $\vec{\mathbf{m}}$, $\vec{\mathbf{k}}$ are two-dimensional vectors, $\tilde{\mathbf{H}}(\cdot)$ is a 3×3 matrix valued sequence, and $\mathcal{S}$ is the filter support. Efficient parallel implementations of the error filter $\mathcal{H}$ suitable for parallel embedded digital signal processors exist [14]. Fig. 4(b) shows an example halftone obtained from the original *peppers* image of Fig. 4(a).

III. The JBIG2 standard

A. Overview of the JBIG2 standard

JBIG2 is an international standard for lossy and lossless bi-level image compression. JBIG2 is the first international standard that provides for lossy compression of bi-level images. JBIG2 will be useful for fascilmile, print spooling, document storage and archiving, images on the World Wide Web, wireless data transmission and teleconferencing. JBIG2 allows both quality-progressive as well as content progressive encoding. A typical JBIG2 encoder decomposes the input bi-level image into several regions or segments (usually based on content), and each of the image segments is separately coded using a different coding method. Thus text and halftone information can be coded separately with JBIG2. An excellent overview of JBIG2, including compliant bitstream specification, coding modes and suggested lossy preprocessing and postprocessing methods is given in [3].

Fig. 3 shows a typical JBIG2 halftone codec. JBIG2 makes no *a priori* assumptions about particular halftone types in defining a compliant bitstream. This means that a halftone coded with JBIG2, can be decoded even though there is *no information in the bitstream* indicating the type of halftoning used at the encoder. A typical JBIG2 lossy encoder converts error halftones back to grayscale (a.k.a. inverse halftoning) and the grayscale values are transmitted. In this method the bi-level image may be divided into pixel blocks of m_b rows and n_b columns. The grayscale image can be any function of the binary pixel values in the corresponding $m_b \times n_b$ block. Usually a simple average of pixels in the block is used [4]. The lower resolution grayscale image is then Gray coded, and the bitplanes are coded using context-based arithmetic coding as in JBIG1 [15]. The decoded grayscale values at the decoder are then used as indices into a halftone bitmap dictionary. The dictionary of binary patterns (bitmaps) is itself transmitted to the decoder using lossless JBIG1 based coding methods. The JBIG2 decoder simply decodes the transmitted indices and uses them to look up the matching bitmap from the pattern dictionary. JBIG2 allows for lossy postprocessing of transmitted bitstream without specifying how this is to be done. This is to allow the freedom at the decoder to tune the reconstructed image to a particular device [3]. This gives the decoder the freedom to use patterns other than those in the pattern dictionary for reconstructing the halftone.

IV. Impact of work: Pipeline Optimizations

Previous JBIG2 compliant lossy halftone coding methods [4], [7] only consider JBIG2 encoder improvement. In [7] the authors consider the coding of stochastic halftones with JBIG2. Both these encoders, however are not optimized for a particular type of halftoning at the decoder. Often in print spooling, the designer knows the algorithm used in the decoder. The encoder can thus be optimized for a particular type of halftoning scheme, and rendering resolution used at the decoder. Such an optimization was explored for a JPEG-compliant printer pipeline in [2]. The authors optimize discrete cosine transform (DCT) quantization tables and huffman coding tables for a particular halftoning algorithm including image scaling distortions. Because JPEG does not allow explicit bit-allocation, the encoder/decoder combination was not rate-distortion optimal. In the context of JBIG2 one can optimize the encoder to take into account degradations introduced by the decoder in a rate distortion sense. This means that the encoder will use less bits in regions/frequencies where halftoning or resizing introduces a lot of distortion [8]. End to end optimization of the printer pipeline is not restricted to JBIG2 encoder optimization. Wong [16] optimizes error diffusion to produce halftones that are more compressible using lossless compression. Such algorithms are suitable for printer pipelines employing JBIG2 lossless encoding. Our printer pipeline simulator will incorporate metrics to judge end to end performance. We will quantify encoding rate and its relationship with perceptual linear frequency distortion and noise. Thus the effect of changing a single component of the pipeline may be judged on the entire system [17].

V. Pipeline Modeling

Fig. 1 shows a generic printer pipeline with simple feedforward dataflow semantics [18]. In our implementation of this pipeline in Ptolemy, we will focus on the image scaling, halftoning, JBIG2 compression, and JBIG2 decoding blocks. The color transformation is printer specific and will be left as a spaceholder for application-specific customization of our generic model.

On each firing, the image scaling block will take in a single token consisting of a c-element vector pixel where c is the number of color planes and output p c-element vector tokens where p is an integer interpolation factor. Internally, the image scaling block will be an interpolator with upsampling factor p. The halftoning block will contain error-diffusion halftoning (Fig. 2) using vector synchronous data flow (SDF) [18]. On each firing, the halftoning block will take in a vector token containing the continuous c color plane values of a pixel, and output a vector token containing the quantized c color plane values of a pixel. The JBIG2 compression block will take n tokens corresponding to the size in pixels of the interpolated image and output one vector equivalent to the entire encoded bitstream containing indices and bitmaps. The JBIG2 decoder will take in one token, the encoded bitstream vector, and output the decoded image. Internally, the JBIG2 encoder and decoder will both use dynamic data flow (DDF) [18] for the data-dependent coding and decoding. All blocks will be implemented as Ptolemy stars written in ptlang.

References

[1] R. W. G. Hunt, *The Reproduction of Color.* Fountain Press, 5th ed., 1995.

[2] R. A. V. Kam, P. W. Wong, and R. M. Gray, "JPEG-compliant perceptual coding for a grayscale image printing pipeline," *IEEE Trans. Image Processing*, vol. 8, pp. 1–14, Jan. 1999.

[3] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W. J. Rucklidge, "The emerging JBIG2 standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, pp. 838–848, Nov. 1998.

[4] B. Martins and S. Forchhammer, "Halftone coding with JBIG2," *IEEE Trans. Image Processing.* to appear.

[5] L. Akarun, Y. Yardimici, and A. E. Cetin, "Adaptive methods for dithering color images," *IEEE Trans. Image Processing*, vol. 6, pp. 950–955, July 1997.

[6] N. Damera-Venkata and B. L. Evans, "Design and analysis of vector color error diffusion halftoning systems," *IEEE Trans. Image Processing*, submitted.

[7] M. Valliappan, B. L. Evans, D. A. D. Tompkins, and F. Kossentini, "Lossy compression of stochastic halftones with JBIG2," *Proc. IEEE Conf. Image Processing*, vol. 1, pp. 214–218, Oct. 1999.

[8] N. Damera-Venkata and B. L. Evans, "Optimized JBIG2-compliant error diffused halftone coding," in preparation.

[9] T. Rao, G. Arce, and J. Allebach, "Analysis of ordered dither for arbitrary sampling lattices and screen periodicities," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1981–1999, Nov. 1981.

[10] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," *Proc. Soc. Image Display*, vol. 17, no. 2, pp. 75–77, 1976.

[11] D. Anastassiou, "Error diffusion coding for A/D conversion," *IEEE Trans. Circuits and Systems*, vol. 36, pp. 1175–1186, Sept. 1989.

[12] T. Bernard, "From Σ-Δ modulation to digital halftoning of images," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 2805–2808, May 1991.

[13] R. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, pp. 56–79, Jan. 1988.

[14] N. Damera-Venkata and B. L. Evans, "Parallel implementation of multifilters," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, June 2000. to appear.

[15] JBIG, "Progressive bilevel image compression," 1993. ISO/IEC International Standard 1154.

[16] P. W. Wong, "Entropy-constrained halftoning using multipath tree coding," *IEEE Trans. Image Processing*, vol. 6, pp. 1567–1579, Nov. 1997.

[17] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image quality assessment based on a degradation model," *IEEE Trans. Image Processing*, May 2000. to appear.

[18] B. L. Evans, *Class Notes for EE 382C: Embedded Software Systems.* Spring 2000. The Univerity of Texas at Austin.

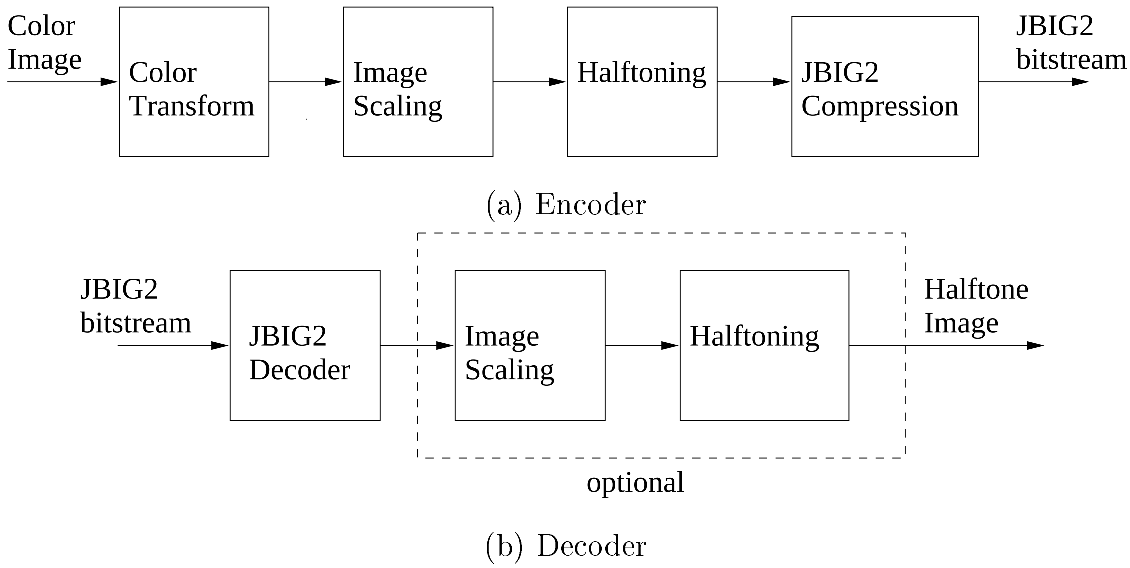


Fig. 1. System block diagram for a color image printing pipeline using JBIG2 compliant halftone coding.

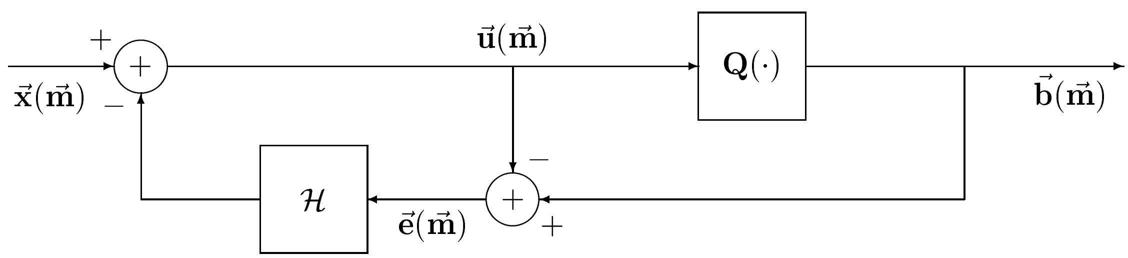


Fig. 2. System block diagrams for vector color error diffusion halftoning where $\mathcal{H}$ represents a fixed 2-D nonseparable FIR error filter with matrix valued coefficients.

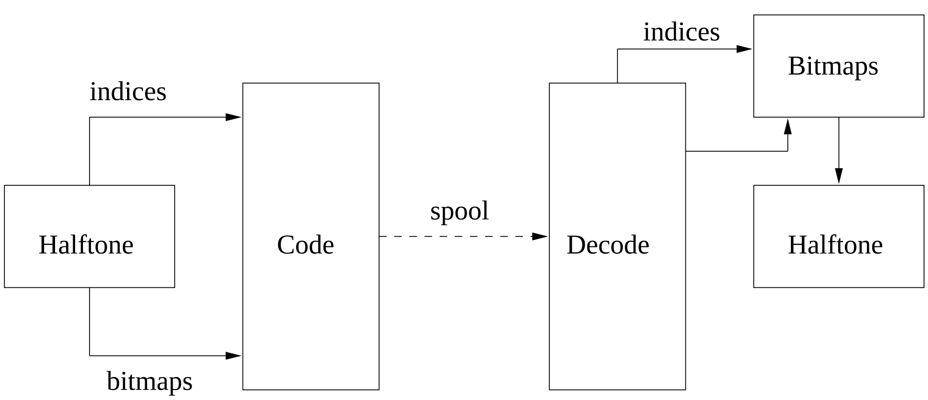
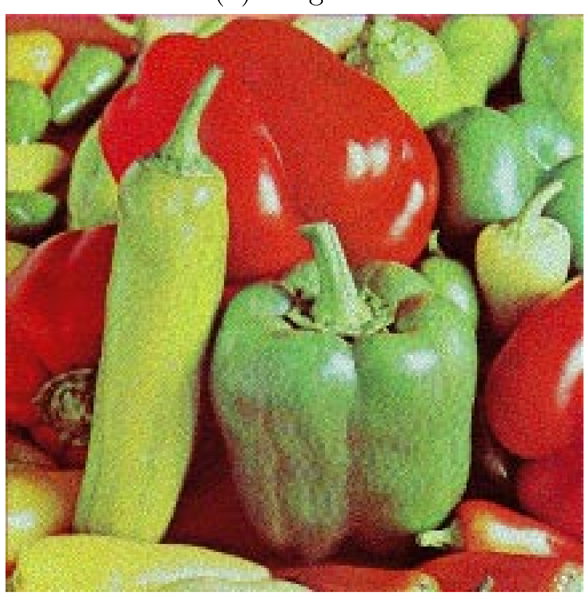


Fig. 3. Block diagram illustrating halftone coding with JBIG2.

(a) Original



(b) Halftone

Fig. 4. Halftoning with color error diffusion.