Pulse-Width Modulation for Rendering Color Text and Graphics for Laser Printers

Jincheng Huang, Onur Guleryuz, Anoop K. Bhattacharjya, and Joseph Shu
Epson Palo Alto Laboratory
Palo Alto, California

Abstract

We present techniques for rendering high-quality color text and graphics on a laser printer using pulse-width modulation (PWM). Multilevel tones for these printers are generated using a combination of screening and pulse-width modulation signals that switch a laser source to render an image on an electrically-charged surface.

In the past, PWM techniques have been used to improve the rendering of text/graphics in saturated primary tones, however, rendering gray or arbitrary colors using these techniques typically leads to noisy text/graphics boundaries, primarily due to the inability of a laser printer to print weak or isolated pulses at the boundaries.

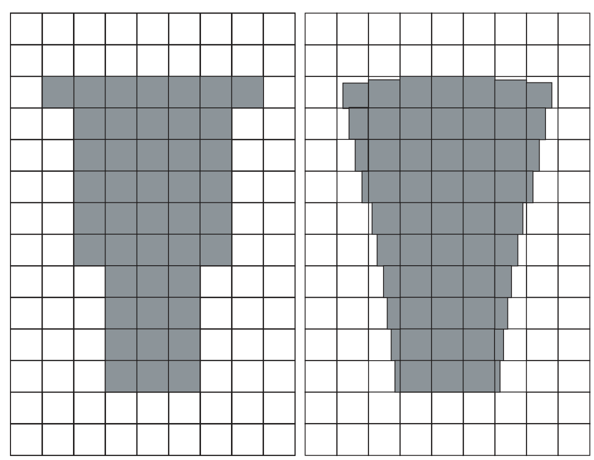
In this paper, rendering of all graphics/text tones is achieved by using a fast chain-code lookup table to determine the antialiasing required at a given pixel location. This table is precomputed using a wavelet-based boundary denoising algorithm. We present two techniques that use the antialiasing information to modify the output of the screening algorithm in order to generate pulse patterns that improve the stability of rendering at text/graphics boundaries. The presented techniques lead to superior text/graphics quality and smooth boundaries.

1. Introduction

It is always desirable to print text/graphics objects with smooth boundaries. However, the limitations of image acquisition devices produce data with jagged text/graphics objects. Therefore, many post-processing algorithms have been proposed to improve the quality.

Pulse-width modulation (PWM) techniques have been used to improve the rendering of text/graphics in saturated primary tones. PWM takes advantage of the high addressibility of a printer in order to render text/graphics at higher resolutions. Crean *et al* [1] disclosed a method to render image pixels at near-saturation levels. This technique applies an anti-aliasing filter to text/line art regions, identifies the pixels being modified, and renders the identified pixels. Zeck *et al* [2] describe a technique to encode higher resolution edge information in lower resolution image information. Two different rendering methods are used for the high-resolution edge and low-resolution contone.

One way to handle gray or arbitrary color text/graphics is through the use of halftones. However, halftoning can introduce jaggedness depending on the halftone method used. Figure 1 (a) shows a gray text/graphics object and Figure 1 (b) shows the object after smoothing. Figures 2 (a) and (b) are the schematic representations of the results of halftoning the objects in Figures 1 (a) and (b), respectively. As seen from the figures, halftoning artificially creates uneven boundaries. Furthermore, since PWM signals switch a laser source to render an image on an electrically-charged surface, a weak signal can fail to attract toner particles. Hence, the laser printer can fail to print weak isolated PWM signals. Examples of such weak signals can been seen in Figure 2 (b) (enclosed by ellipses). This phenomenon can be problematic when printing objects such as text or thin lines. If the printing device fails to print the isolated pulse, the object's appearance is degraded.



(a) (b)

Figure 1: (a) A Jagged Gray-scale Text/Graphics Object; (b) The Text/Graphics Objects after Smoothing.

In this paper, we propose a new algorithm to render color text/graphics. The algorithm consists of two parts:

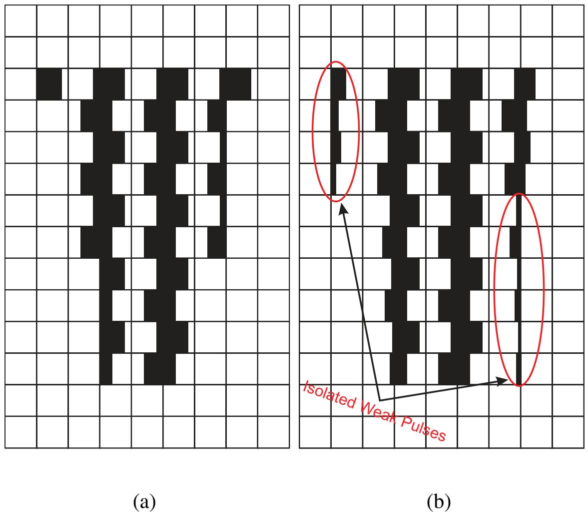


Figure 2: (a) PWM Output for the Jagged Object in Figure 1 (a) using Conventional Halftone; (b) PWM Output for an Smoothed Object as in Figure 1 (b) using Conventional Halftone.

(1) denoising in image domain, and (2) rendering/smoothing in PWM domain. The text/graphics object is smoothed by using a fast chain-code lookup table to determine the antialiasing required at a given pixel location. This table is precomputed using a wavelet-based boundary denoising algorithm. Two rendering techniques use the antialiasing information to modify pulse patterns that improve the stability at the boundaries of text/graphics objects. Hence, these techniques lead to superior text/graphics quality and smooth boundaries.

The paper is organized as follows. Section 2 describes a wavelet-based denoising algorithm to smooth text/graphics boundaries. Section 3 presents two approaches to render text/graphics data in the PWM domain. Finally, Section 4 concludes the paper.

2. Wavelet-based Denoising/Antialiasing

The denoising algorithm views the jagged boundary as a noisy version of an underlying smooth boundary. The pixels on the boundary of a text/graphics object are traced and those that make up the local boundary segment are determined. Traced local boundary segments are parameterized in terms of pixel coordinates, these coordinates are denoised/smoothed using an overcomplete wavelet transform, and finally, a smoothed boundary segment is constructed using the denoised coordinate data. The overview of the algorithm is given in this section. Please see [3] for more details.

The boundary trace is carried out for each text/graphics pixel that lies on the boundary of a text/graphics object. A text/graphics pixel belongs to the boundary of a text/graphics object if at least one of its 4-connected/8-connected pixel neighbors does not belong to a text/graphics object.

The process of the boundary denoising is illustrated in Figure 3. Prior to coordinate decomposition, the boundary segment is first upsampled to form an upsampled boundary segment. The upsampled boundary segment is parameterized by establishing an orthogonal (X, Y) coordinate system. Each pixel is represented in terms of its coordinates. The formed (X, Y) coordinates are considered as two independent sequences and are denoised using an overcomplete wavelet transform to form a new sequence $(\tilde{X}, \tilde{Y})$ of coordinates. The difference of the denoised coordinates and the original coordinates are determined as the differential $(d_x, d_y) = (\tilde{X}, \tilde{Y}) - (X, Y)$. The rendering stage only requires the differentials that correspond to the current pixel for which the local boundary segment is traced.

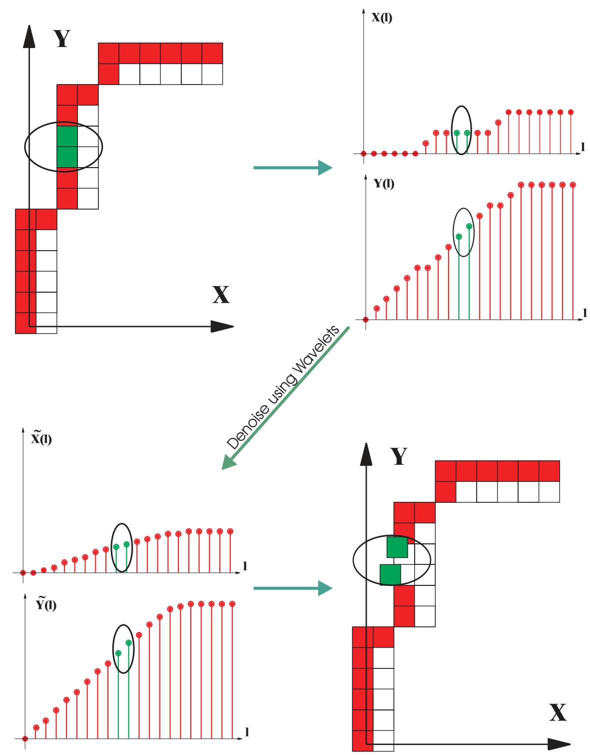
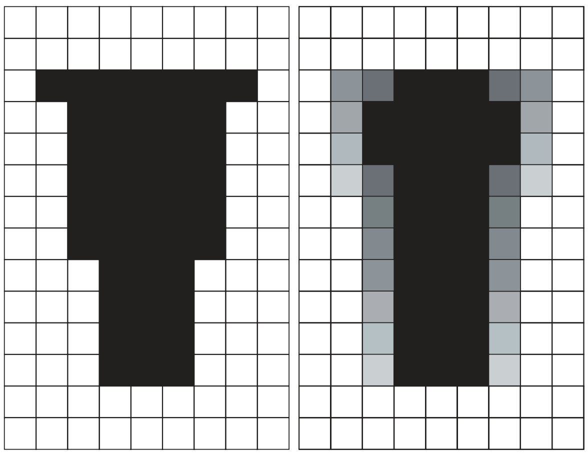


Figure 3: Wavelet-based Denoising.

After the new coordinates for the text/graphics are constructed, a rendering stage follows. New values for each color plane are calculated. $Alpha$, which represents the percentage of foreground blending into background or vice versa, is also determined. The rendering algorithm blends text/graphics pixels smoothly into the background. $Alpha$ has values range from 0 to 1; any value between 0 and 1 (excluding 0 and 1) indicates a pixel is a mixture of background and foreground. Figure 4 (b) is an example of $Alpha$ for the object in Figure 4 (a) after denoising. Black represents the pixels in the original text/graphics object; white represents background; and gray represents values between 0 and 1, where the darker pixel, the higher alpha value.



(a) (b)

Figure 4: (a) A Jagged Text/Graphics Object; (b) Alpha Plane Generated after Wavelet-based Denoising and Rendering.

In order to speed up in high-resolution printing, the boundary denoising is carried out off-line and only the results are stored in a LUT. The LUT is accessed using the index constructed from the chain-code of the traced boundary segment at run-time. All of the computationally complex operations are conveniently represented in lookup tables (LUTs) for fast and computationally simple operation. The traced boundary is used to construct a chain-code which addresses the LUTS containing the results for subsequent stage. Furthermore, memory requirements for the algorithm and the sizes of the LUTS can be precisely controlled by adjusting the size of the pixel neighborhood in the construction of the boundary segment.

3. PWM Rendering for Text/Graphics

As mentioned early, halftone process can create jaggedness for printing. In this section, we describe two approaches to manipulate the PWM data to maintain the smoothness of text/graphics objects. In the first approach, halftoning and denoising are implemented in parallel, where as those two processes are in serial in the second approach.

3.1. Approach I: Parallel Implementation

In this implementation, antialiasing and halftoning are in parallel as shown in Figure 5. The output of those two modules are combined in PWPA (Pulse Width and Position Adjustment) module according to $Alpha$.

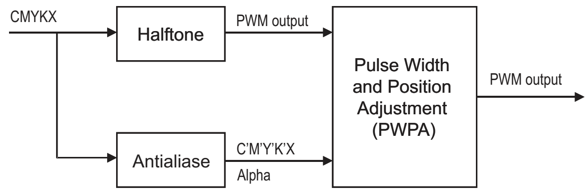


Figure 5: Block Diagram for Approach I.

In Figure 5, the input data to the halftoning and antialiasing modules are CMYKX. CMYK are the values for the four different color planes (Cyan, Magenta, Yellow, and Black). X is the attribute plane which indicate whether a pixel is text, graphics, or photo. Original image data (CMYK) are halftoned and PWM data are generated. According X, templates for all text/grpahics objects are generated. Antialiasing process smoothes the object boundaries and re-generates image data (C'M'Y'K') and an Alpha plane.

PWPA module takes the outputs from halftoning and antialising, and modifies PWM data accordingly. If $Alpha$ is 0 or 1, PWM data remains unchanged. If a pixel is modified, in which $Alpha$ is not 0 and 1, PWM data at that location is replaced by new values. The values taken from a supplied gamma table, in which the relationship of pulse width and input is monotonic.

If a pixel lies on the boundary of the alpha plane (in which one of its neighbors is a background pixel), pulse position is re-adjusted. The reason is that the partial pulses in an isolated environment is not likely be printed out by laser printer. The adjustment follows the following rules. If the left and the right of the pixel are background pixels, the pulse is placed at the center. The outer-most pixel on the left boundary shifts its position to the right and its right non-background neighbor pixel to the left. The pulse of the outer-most pixel on the right boundary is aligned to the left and its left non-background neighbor to the right. An example of the final PWM output from Approach I is shown in Figure 6. Black indicates the original PWM data, where gray is the pixel modified by PWPA.

3.2. Approach II: Serial Implementation

Figure 7 is the block diagram of serial implementation. In this implementation, halftoning follows antialiasing. Antialiasing is performed to smooth the boundaries of text/graphics objects contained in the data, and generate antialiased data and $Alpha$ plane. The antialiased data are halftoned. BSPA

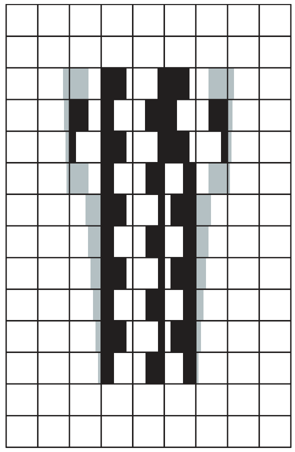


Figure 6: *PWM Output from Scheme I.*

(Boundary Smoothing and Position Adjustment) module modifies PWM data on the boundary of text/graphics objects according to $Alpha$.

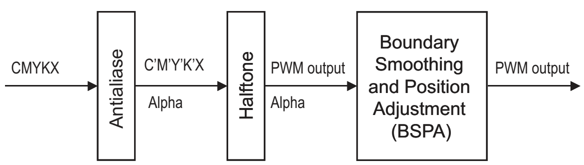


Figure 7: *Block Diagram for Approach II.*

An example of PWM data representation of a smoothed object after halftoning is shown in Figure 2 (b). Isolated pulses at the boundary can be seen. Those pulses probably will not be printed out. To circumvent the problem, the PWM data on the text/graphics boundary are needed to re-align. The same rules are used as described in the previous section in which the outer-most left one to the right and the outer-most right one to the left and their non-background adjacent pixels move closer to them. So the pulses on the boundary cluster and are stronger to survive when printed. The re-aligned data is given in Figure 7 (a).

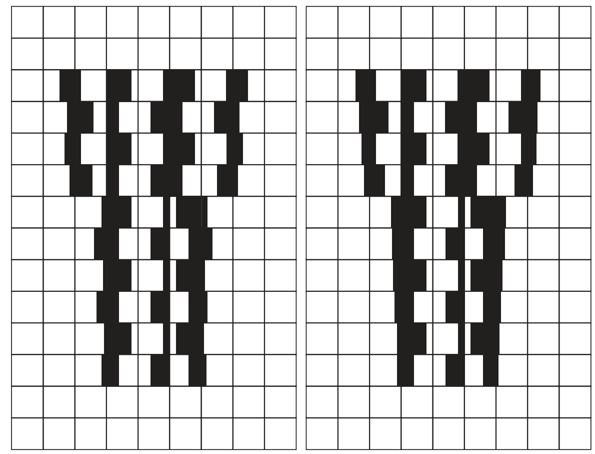
Since halftone can introduce jaggedness as seen in Figure 7 (a), additional smoothing is introduced in PWM data domain. Vertical text/graphics boundary is first detected using $Alpha$ information and then low-pass filtering is applied to the boundary (in PWM domain). Let $p(x, y)$ be the pulse at location (x, y), and p_1, p_2, p_3 denote $p(x, y-1)$, $p(x, y)$, $p(x, y+1)$, respectively. If a vertical boundary is found and $p_1 > p_2 > p_3$, the new pulse widths (p'_1, p'_2, p'_3) are calculated as:

$$p'_1 = \frac{2}{3} \times p_1 + \frac{1}{3} \times p_2, \quad (1)$$

$$p'_2 = \frac{1}{3} \times (p_1 + p_2 + p_3), \quad (2)$$

$$p'_3 = \frac{1}{3} \times p_2 + \frac{2}{3} \times p_3. \quad (3)$$

Figure 8 (b) is an example of PWM output after boundary-smoothing and position adjustment.



(a) (b)

Figure 8: *PWM Output of Approach II: (a) after Position Adjustment; (b) after Boundary Smoothing and Position Adjustment.*

4. Conclusion

The wavelet-based antialiasing algorithm blends the foreground (text or graphics objects) with the background. Hence, it is more tolerant to registration error. The two PWM rendering algorithms render PWM data on the boundary to improve the stability of the boundary pixels. These techniques lead to smooth boundaries and superior text/graphics quality.

References

[1] P. A. Crean and R. P. Loce, "Method for Generating Rendering Tags to Facililate the Printing of Antialiased Images," U.S. Patent 6,114,461.

[2] N. W. Zeck and T. A. Henderson, "Method of Encoding High Resolution Edge Position Information in Continuous Tone Image Information," U.S. Patent 6,020,979.

[3] O. G. Guleryuz and A. K. Bhattacharjya, "Fast Text/Graphics Resolution Improvement using Wavelet-based Denoising and Chain-Code Table Lookup" to appear in IEEE Int'l Conf. on Image Proc. (ICIP2002), Rochester, New York, Sept. 22–25, 2002.