Hard Edge Solution for Digital Proofing

Lucien Xu, Patrick Thomas and Bruce Brier*

Keywords: Compression, Image, Proof, Raster

Abstract: Digital proofing technology is getting wider use nowadays, particularly in the direct to plate workflow where it is necessary. Comparing digital proofing output to that of analog proofing, such as AgfaProof, contone quality decent, even at 300 dpi. However, digital proofs lack the sharp (hard) edges necessary for text and line art. In order to obtain acceptable hard edge quality, higher resolution digital proofing devices such as a 2400 dpi printer, are required. However, for the RIP twice scenario, 2400 dpi pages have 64 times more data than 300 dpi to store and pass through network. It is not practical at all in implementation. In this paper, we proposed a PostScript hard edge solution to compress a 2400 dpi page to the data size comparable to that of 300 dpi. By applying it to a high resolution digital proofing device, this PostScript hard edge solution can support a proof quality as good as the same resolution analog proof but with a data size of at most 1/64. The performance of the implementation is linear related to that of the corresponding Run Length Encoding and file saving.

Introduction

Analog proofing technology such as using AgfaProof or Matchprint is commonly used in conventional pre-press and press workflow. Digital proofing technology is necessary to support proofing through various digital proofing printers in the direct to plate workflow. Figure 1 shows a typical direct to plate workflow with a RIP twice digital proof path. A Direct Digital Color Proof device such as Polaroid PolaProof can be used to have direct proof in addition to the proofing path on Figure 1, but the issue of the DDCP is out of the scope of this paper.

Commonly used printers usually have their own RIP as the second RIP shown in Figure 1 that normally takes composite CMYK color data as input. The second RIP could behave very differently from the first RIP which generates the raster images for direct to plate. A RIP twice scenario guarantees the main RIP's behave dominating that of the second RIP. This RIP twice scenario has been successfully implemented in the AGFA product **SelectProof**.

---

* Bayer, AGFA division, 200 Ballardvale St., Wilmington, MA 01887

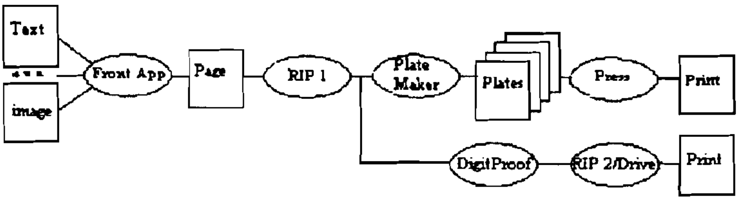

Figure 1. Pre-press/Press workflow dialog

Either one bit (screened) or 8 bit (no screening) rasters can be composed to different resolution proofs via digital proof in the RIP twice scenario. In general, 8 bit rasters give better proofing quality than one bit can support. The descreening process in one bit workflow adds moires and blur on edges. The moires is due to the mismatch between screening and descreening masks, while the edge blurring is caused by the average effect of the descreening. Except the moire, 150 dpi one bit descreened image has about the same spatial resolution and color quality as that of an 8 bit since the screen ruling is usually about 150 lpi. In general, the higher the descreening resolutions, the sharper the hard edges on the image. But the color quality can be obtained when the descreening resolution matches the screening ruling. Since no descreening is needed, 8 bit rasters can support the better color quality and sharper hard edges. In principal, using 8 bit RIPing in higher spatial resolution such as 1200 dpi or 2400 dpi can produce competitive color and hard edge quality digital proof, comparable to that of conventional analog proof. However, the data size in this approach will be dramatically increased from its one bit (screened) version.

For instance, an 8 up job is RIPed in 2400 dpi one bit with the raster image file size of 32x22x2400/8x2400/8x4 = 253 Mb. This size of raster image file is big but still workable. When RIPing the same page in 8 bit, the raster image file size will go up to 253x64 = 16220 Mb. This size of data is very hard to store on disk in the digital proofing workflow unless this data can be compressed to about the size of its one bit (screened) version that means a 64 times compression rate is expected.

Image compression approaches

Lossness image compression approaches, such as JPEG are not suitable for raster image compression. This is because lossness image compression algorithms trade compression ratio with the loss of high frequency image information that mostly contribute to edges on raster images.

Commonly used lossless image compression approaches are RunLength and LZW encoding. Their compression ratio is very much image type and content dependent. Typically, for an image (document page) mixed with tint, line art, text and contone images, the compression ratio largely depends on the percentages of different type of images. Table 1 lists a set of compression ratios for some typical images.

Image type	RunLength (QuarkXPress)	LZW (Tiff)	PostScript (QuarkXPress)
Tint / Line art	1:64	1:128	1:240
Text	1:300	1:100	1:300
Contone	1:1	1:1.5	1:1

Table 1. Compression ratios for typical image types

A test page with 16 color bars made from QuarkXPress is chosen as a typical tint / line art image. A test page with two columns text in black is chosen as a typical test image. Finally, a test page with a 300 dpi contone image is selected as contone image. The size of uncompressed tiff file is used as the reference. The first column shows the compression ratio of the PostScript file size saved from QuarkXPress that includes the corresponding tiff files. The second column shows the compression ration of LZW compressed tiff file from the corresponding uncompressed tiff file. The last column gives the compression ration of the original PostScript files over the corresponding tiff files. Apparently, contone images can be compressed very little. The RunLength encoding compresses text image more than LZW does while LZW compresses tint / line art image more than RunLength does. However, PostScript gives the best compression ratios in general. To depict the PostScript solution, two issues have to be resolved: contone images have to be compressed at least 64 times, and a image segmentation is required to distinguish different image types.

Consider the fact that commonly used screen rulings are less than 300 lpi (usually 133, 150 and 175), 2400 dpi 8 bit RIPed raster contone images have no more than 300 dpi spatial resolution. Therefore, the 2400 dpi 8 bit raster image can be sub-sampled to 300 dpi without any visual resolution loss. This sub-sampling gives a data compression ratio of 1:64 (for 300 dpi, 1:256 for 150 dpi) .

Depend on where in the pre-press / press workflow to apply this PostScript solution, if RIP information can be accessed, there no additional work is required. Otherwise, a image analyzer can be designed to segment a document page into different type of images.

PostScript solution

Figure 2 shows a diagram for the PostScript hard edge solution. It can be summarized as:

1. The data flow of the CMYK rasters is in page (channel) sequence. In this way, additional compression can be achieved for unbalanced data between channels. For example, the C channel does not need to be encoded at all if there is nothing but background.
2. Each separated page is encoded into an EPS file.
3. Each separated page is given in image band, 200 rows every band for example.
4. The image analyzer in Figure 3 groups the current image band into margin, tint, text and contone image boxes. Each image box contains a clip path (a bounding box for rectangular) and data.
5. The margin boxes are skipped. Additional compression can be achieved.
6. The tint image boxes are encoded in PostScript clip path and filling value.
7. The text image boxes are encoded in clip path and RunLength code.
8. The contone image boxes are encoded in clip path and sub-sampled code in 300 dpi.

The following EPS diagram can fit in different workflow. For example:

1. For a raster workflow, a simple mapping mechanism can be used to map those encoded box data back to raster image data.
2. For a PostScript workflow, a PostScript wrapper can be used to wrap all EPS's to a PostScript printing job.
3. For a PDF workflow, a PDF wrapper can be used to wrap all EPS's to a PDF printing job.

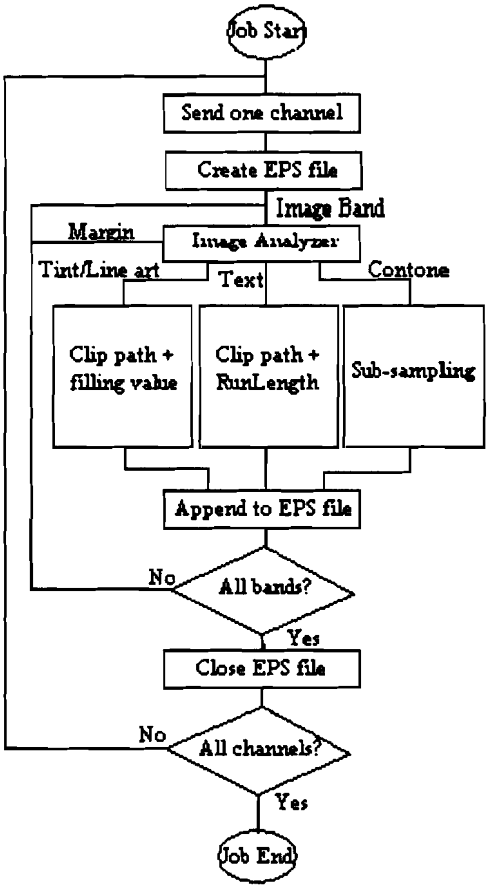

Figure 2. Diagram of the PostScript hard edge solution

Test system

A prototype of the PostScript hard edge solution defined above is implemented in Visual C++ 4.0 on MS/Windows NT workstation. An image analyzer is designed to segment the test page into margin, text or contone image boxes. Then the margin image boxes are skipped, the test boxes are encoded in Hex-code RunLength, while the contone image boxes are sub-sampled in 8x8 pixels.

A test document page is constructed in the size of 8.5x11 inches. A 300 dpi contone image is on about half of the test page, and the other half is filled with 12 point black text. This test document page is RIPed to 8 bit tiff file with 2400 dpi resolution. These raster tiff images are the input of the test system.

The output EPS files from the test system are converted to PDF via Adobe **Distiller** and printed out via Adobe **Exchange** to AGFA Avantra 44 in 2400 dpi and Polaroid DryJet in 600 dpi. The original QuarkXPress page is also output directly to Avantra and DryJet as references.

Test result and discussion

Table 2 shows the test result for the black channel. The first two columns shown the result of the visual comparison between the direct output film and the test output film. They are both in 2400 dpi, Positive and Right Reading. There is no noticeable difference between the contone images on both films, and no difference in the shapes between the test images on both films. The third and fourth column show the same visual result for DryJet prints that are in 600 dpi. The compression ratio can be calculated from the last column apparently as 3.7:538.6 or equivalent to 1: 145 which exceeds the expectation of 1:64.

	Avantra Contone	Avamtra Text	DryJet Contone	DryJet Text	File size
Direct output	Not noticeable	No difference	Not noticeable	No difference	538.6 M*
Test system output	Not noticeable	No difference	Not noticeable	No difference	3.7 MB** (13.6 MB)

* Uncompressed tiff file size of 2400 dpi black raster
** The number is adjusted by dividing 2x8 times to discount the two byte hex-code and bit pack for bi-level test in the test system. The braced number is the size of the EPS file.

Table 2. Test result

In this test, the contone image is sub-sampled in 8x8 pixels. If the line ruling is given, the contone image should be sub-sampled accordingly. Therefore, the quality and the compression ratio can be further improved.

The image analyzer in the PostScript hard edge solution is not necessary if the image box information is given. However, in the use of the image analyzer, the functional ability, mis-recognition tolerance and performance of the analyzer will restrict the application of the PostScript hard edge solution.

Conclusions

1. The PostScript solution can compress the 2400 dpi high resolution raster images to the size of at least 1/64 of the original.

2. The test results show that the PostScript solution compressed image reserves the same hard edges as its original on text image and no distinguishable image distortion on contone image.

3. The performance could be linear to that of pure RunLength encoding.

4. The PostScript solution can easily fit into various workflow, particularly for PostScript and PDF.

References

Adobe Systems, Inc.
"PostScript Language Reference Manual", 2nd ed., Jan. 1991.

Adobe Systems, Inc.
"Portable Document Format Reference Manual, v1.2, Nov. 27, 1996.

Lucien Xu
"Internal Report On Preliminary Feasibility Study of Hard Edge Solutions for Digital Proof 2.0", Nov. 17, 1997.

Lucien Xu
"Internal Report On Prototype Test of Hard Edge Solution", Jan. 21, 1998.