An Analysis of Raster Formats for Printing

August 9, 2011

Michael R Sweet
Apple Inc.

Abstract

This white paper provides an analysis of the common CUPS, JPEG, and PNG raster formats that can be used for printing along with supplementary compression algorithms to determine which combinations offer the best utilization of resources and bandwidth.

Introduction

Printers have traditionally relied upon multiple vendor- or device-specific markup languages to produce output. As printers have become more capable and more kinds of clients want to produce output on printers, it is important to identify the useful generic document formats that are feasible to support in both printers and clients.

This white paper explores several existing image formats along with supplemental compression algorithms that can be applied to those formats. Space and time performance has been measured with a series of test documents with benchmarks performed on an Apple MacBook Air with a 2.13 GHz Intel Core 2 Duo processor and 4GB of RAM.

Testing

The test documents for used for this analysis consist of sample email messages, maps, web pages, brochures, and a JPEG digital camera image. Raster data was produced using CUPS filters included with Mac OS X (cgimagetopdf, cgpdftoraster, rastertopwg) as well as several custom filters for the standard image file formats (rastertojpeg, rastertopng). Supplemental compression and timing were performed using standard command-line tools (awk, compress, gzip, time).

The results are a simple average of three test runs with raster data at 300 pixels per inch (PPI). There was very little variation between runs. CUPS Raster was tested both for version 2 (RLE + PackBits) and version 3 (uncompressed). JPEG output used the default quality provided by the IJG (libjpeg 8c) implementation. PNG output used the default compression settings provided by the libpng (1.5.4) implementation. Supplemental compression (bzip2, compress, gzip, and xz) used the default settings offered by each program.

Summary of Results

Table 1 lists the various compression algorithms used for the common file formats with the RAM and CPU requirements and average compression ratio for color printing. Server CPU usage is low in all cases, and in general combining CUPS Raster V2 with a supplemental compression algorithm produced similar compression ratios with lower Client CPU usage.

Format (supplemental algorithm)	Max Client RAM	Max Server RAM	Client CPU	Average Ratio
CUPS Raster V3 (uncompressed)	1 * line	1 * line	low	1:1
CUPS Raster V3 (BZIP2)	7600kiB	3700kiB	high	14.9:1
CUPS Raster V3 (LZ77/gzip)	64kiB	64kiB	medium	10.7:1
CUPS Raster V3 (LZMA/xz)	1000000kiB	1000000kiB	high	15.7:1
CUPS Raster V3 (LZW/compress)	512kiB	512kiB	low	9.1:1
CUPS Raster V2	3 * line	2 * line	low	5.5:1
CUPS Raster V2 (BZIP2)	7600kiB + 3 * line	3700kiB + 2 * line	medium	13.7:1
CUPS Raster V2 (LZ77/gzip)	64kiB + 3 * line	64kiB + 2 * line	low	10.4:1
CUPS Raster V2 (LZMA/xz)	1000000kiB + 3 * line	1000000kiB + 2 * line	high	14.2:1
CUPS Raster V2 (LZW/compress)	512kiB + 3 * line	512kiB + 2 * line	low	8.5:1
JPEG	16 * line	128 * line	medium	18:1
PNG	64kiB + image	64kiB + image	high	29.2:1

Table 1: Summary of resource requirements and performance of different compression algorithms

CUPS Raster V3 (uncompressed)

CUPS Raster version 3 is an uncompressed raster format and has low CPU and memory usage. The lack of compression makes it unsuitable for use over an external interface.

CUPS Raster V3 (BZIP2)

BZIP2 is a large dictionary compression algorithm loosely based on LZ77. For 24-bit RGB data at 300 PPI, performance on the sample documents ranged from 1.6:1 to 374.2:1 with an average performance of 11:1.

CPU usage is typically high on the client and low on the server. Memory usage can be an issue when using larger dictionary sizes.

CUPS Raster V3 (LZ77/gzip)

LZ77 is a medium dictionary compression algorithm with performance ranging from 1.4:1 to 128.6:1 and an average performance of 7.7:1.

CPU usage is typically medium on the client and low on the server. Memory usage is modest at 64kiB.

CUPS Raster V3 (LZMA/xz)

LZMA is an extremely large dictionary algorithm with performance from 1.6:1 to 249.2:1 and an average performance of 11.1:1.

CPU usage is typically high on the client and low on the server. Memory usage can be prohibitive.

CUPS Raster V3 (LZW/compress)

LZW is a dynamic string algorithm with performance ranging from 1.2:1 to 145.2:1 and an average performance of 6.4:1.

CPU usage is typically low on both the client and server. Memory usage is modest.

CUPS Raster V2

CUPS Raster version 2 uses a basic run-length encoding algorithm that encodes sequences of repeating and non-repeating pixels. In CUPS Raster files, the encoding is done for whole pixel values instead of the traditional byte-oriented implementation and can represent up to 128 repeated or non-repeated pixels. In addition, a repeat count is used for each line as a whole so that (typically) blank lines and vertical elements can be represented efficiently.

For 24-bit RGB data, the worst-case performance of the CUPS Raster V2 implementation is 1:1.002 (that is, an increase in size of 0.2%) while the best-case performance is 96:1. Average performance for the sample documents was 4.5:1 at 300 PPI.

CPU usage on both the client and server is very low and can easily be accelerated in hardware. Memory usage is also very low.

CUPS Raster V2 (BZIP2)

Combining the CUPS Raster V2 compression with BZIP2 yielded performance equivalent to CUPS Raster V3 with BZIP2 in 1/3 the time. However CPU and memory usage remain as issues.

CUPS Raster V2 (LZ77/gzip)

Combining the CUPS Raster V2 compression with LZ77 yielded performance ranging from 1.3:1 to 237.4:1 with an average performance of 7.5:1. CPU usage was reduced sufficiently to rate as low for the combination but still double that of CUPS Raster V2 alone.

CUPS Raster V2 (LZMA/xz)

Combining the CUPS Raster V2 compression with LZMA yielded performance ranging from 1.6:1 to 401.6:1 with an average performance of 10.3:1. CPU usage was reduced by half but still remained high.

CUPS Raster V2 (LZW/compress)

Combining the CUPS Raster V2 compression with LZW yielded performance and CPU usage equivalent to CUPS Raster V3 with LZW.

JPEG

JPEG is a discrete cosine transform algorithm with Huffman encoding. Unlike all of the other algorithms discussed in this document, JPEG is lossy and shows visible artifacts at all compression levels for high-frequency data such as text. For 24-bit RGB data at 300 PPI, performance on the sample documents ranged from 21.4:1 to 54.2:1 with an average performance of 38.9:1.

CPU usage is typically medium on the client and low on the server and can easily be accelerated in hardware. Memory usage is low to moderate as long as the JPEG progressive scan modes are not used.

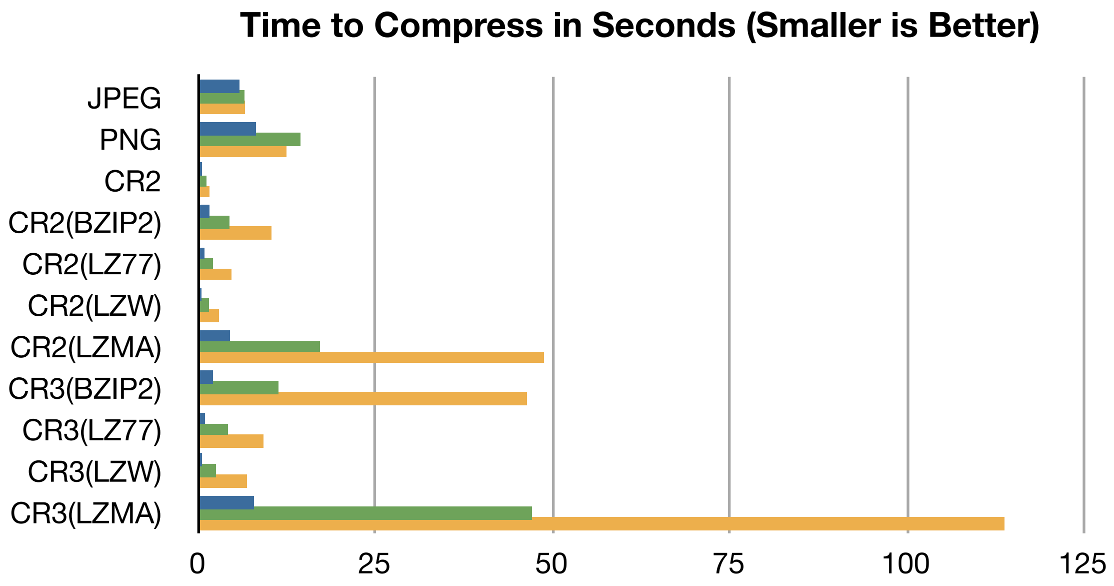
PNG

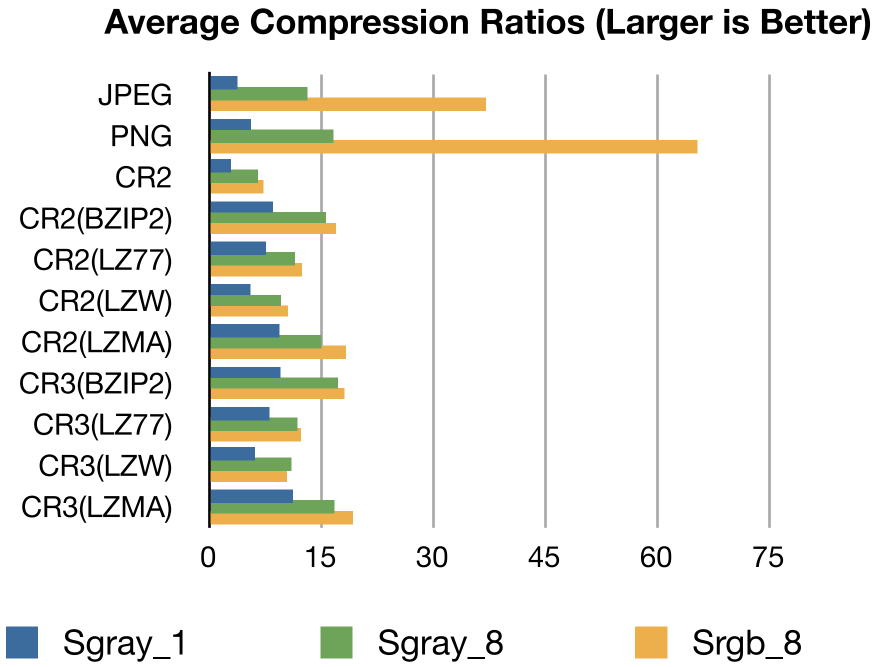
PNG is a LZ77-based algorithm that performs filtering and/or reordering of the image data prior to compression. For 24-bit RGB data at 300 PPI, performance on the sample documents ranged from 2.3:1 to 190.6:1 with an average performance of 12.9:1.

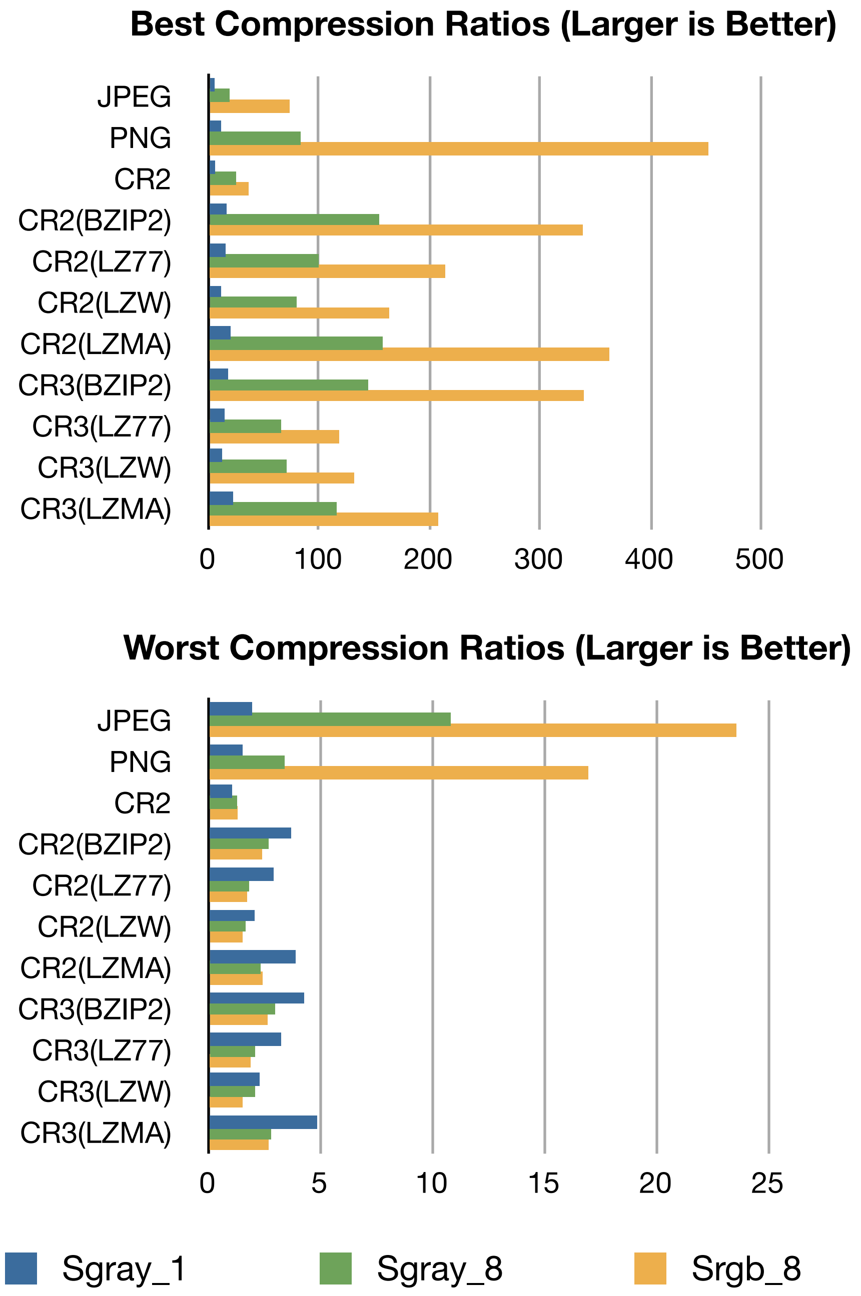
CPU usage is typically high on the client and medium on the server. Memory requirements can be an issue for large images.

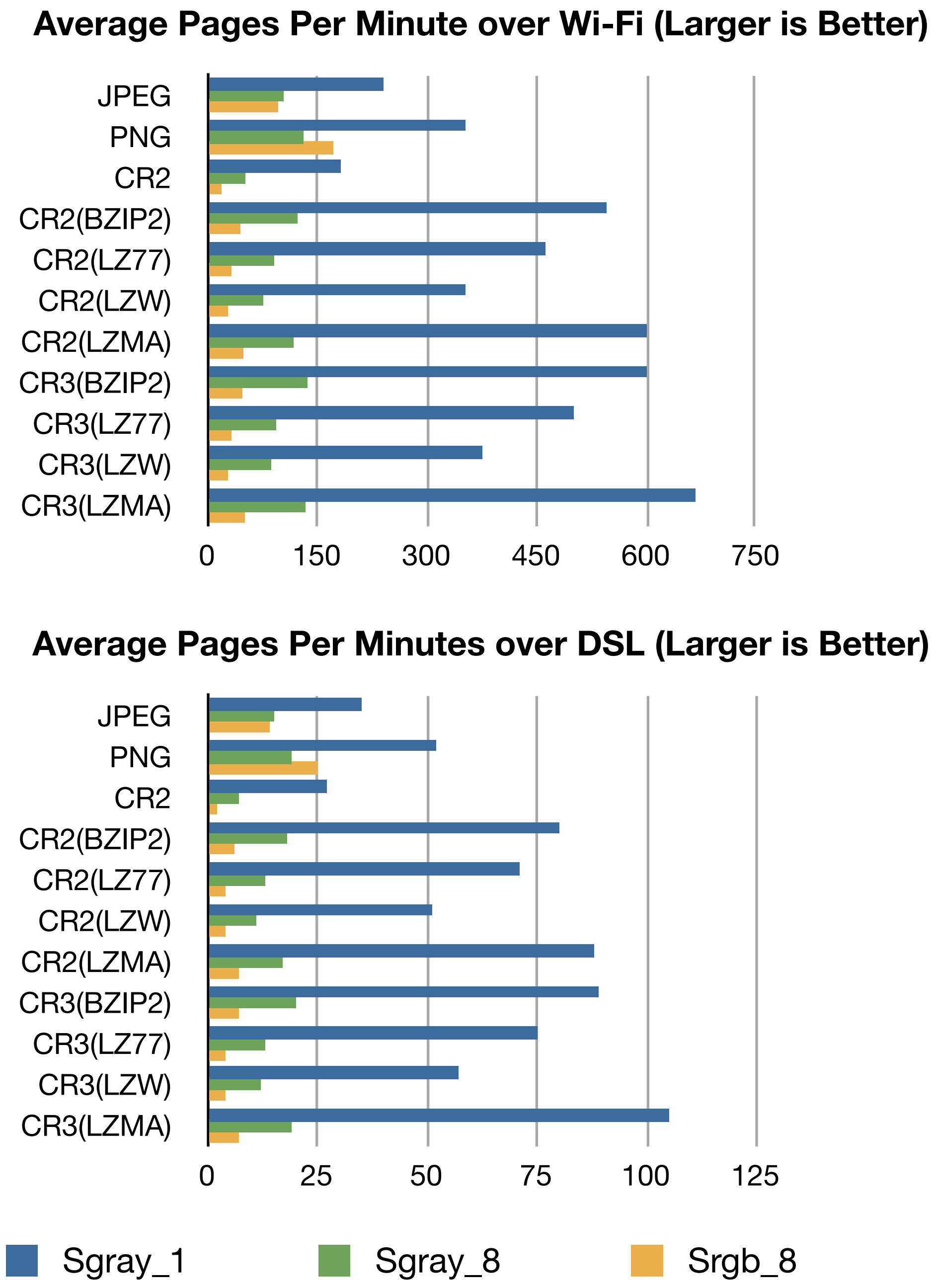
Details

The following charts show the compression performance for all of the text documents as well as the base-case and worst-case results seen for each of the formats along with the sustainable pages per minute for Wi-Fi and DSL networks. The general trend was for the line art and plain text email documents to compress the most and the JPEG image to compress the least, with mixed documents close to the average performance.






Best Compression Ratios (Larger is Better)
JPEG
PNG
CR2
CR2(BZIP2)
CR2(LZ77)
CR2(LZW)
CR2(LZMA)
CR3(BZIP2)
CR3(LZ77)
CR3(LZW)
CR3(LZMA)
0
100
200
300
400
500
Worst Compression Ratios (Larger is Better)
JPEG
PNG
CR2
CR2(BZIP2)
CR2(LZ77)
CR2(LZW)
CR2(LZMA)
CR3(BZIP2)
CR3(LZ77)
CR3(LZW)
CR3(LZMA)
0
5
10
15
20
25
Sgray_1
Sgray_8
Srgb_8



Average Pages Per Minute over Wi-Fi (Larger is Better)

Average Pages Per Minutes over DSL (Larger is Better)

Conclusions

Given the goal of supporting low-cost, resource-limited clients and printers, the BZIP2 and LZMA algorithms are probably not appropriate due to their memory requirements and CPU usage on the client side.

JPEG is an obvious choice when printing photos since it is both the standard format for such files and already supported in existing low-cost printers. However, the compression artifacts and CPU overhead make it a poor choice for compressing normal documents.

The best combination seems to be CUPS Raster Version 2 with LZ77 which has low CPU and memory requirements with excellent overall performance. For extremely limited clients and servers CUPS Raster Version 2 provides extremely low CPU and memory requirements with good overall performance.

For the sample documents rendered at 300 PPI, CUPS Raster Version 2 can easily provide 18 pages per minute over Wi-Fi in color and and about 2 pages per minute over DSL. Adding LZ77 basically doubles the potential print speed to 32 pages per minute over Wi-Fi and 4 pages per minute over DSL.