Contribution to ISO/IEC JTC1 N2393
Lossy Encoding of Bi-Level Images

Cartesian Products, Inc.

July 9, 1995

1 Name of the Proposed Algorithm

The name of the algorithm is Cartesian Perceptual Compression (CPC).

2 Name and Addresses of Contributors

The contributing organization is Cartesian Products, Inc., a high-technology research and development company based in Cambridge, Massachusetts. The company was founded in September 1991 to provide advanced software technology to improve worldwide communication and information access.

The inventors of the method are Peter B. Mark of 30 Wesley Street, Newton, Massachusetts and Stuart M. Shieber of 243 Concord Avenue #2, Cambridge, Massachusetts.

3 Contact Person

For further information, contact:

Stuart M. Shieber
243 Concord Avenue #2
Cambridge, MA 02138
email: shieber@das.harvard.edu
phone: 617-495-2344
fax: 617-496-1066

4 Statement of Intellectual Property Rights

The CPC method is the subject of a United States patent number 5,303,313. International patents have been applied for.

5 The CPC Algorithm

The CPC method is a symbol-matching compression method for bi-level document images that incorporates several innovative features that give it excellent image quality and compression ratio while being practical in computational resources required. Successive pages of a document are segmented into symbols consisting of contiguous regions of black pixels. These symbols are then grouped according to perceptual similarity by matching the symbols against an (initially empty) catalog of template symbols that is augmented with new (perceptually distinct) symbols as they arise. The bit image of each template symbol is composed from those of the matching source symbols by an averaging process. No distinction is made between regions of the image comprising text and regions comprising graphics; all are treated as composed of symbols subject to matching.

The key to the success of any symbol-matching compression scheme is the ability to find a definition of perceptual similarity that trades off the countervailing goals of (i) conservativeness, so that perceptually distinct characters are not deemed similar, causing substitution errors of one for the other; and (ii) aggressiveness, so that a large number of matches are found allowing for good compression results; while maintaining (iii) efficiency, so that matching can be tested quickly, as it must be done many times while processing a document. The CPC method uses a combination of indexing and rough filters to eliminate large numbers of matches, and a diverse set of novel efficiently computable fine-grained match tests to eliminate false matches. The CPC method has been designed always to prefer conservativeness over aggressiveness; no perceptible image degradation has been deemed admissible.

The matching process reduces the image to three major streams of information, providing information about the bit images of the templates, and the position and template reference for each symbol in the document. These streams are then further coded using an arithmetic coder combined with a cascade of separate varying-order models for each stream.

The method operates on image documents which are comprised of individual image pages. There is no limit to the number of pages contained within a document.

The method has been designed to operate within a fixed memory bound, regardless of document length. Typically, we assume that a megabyte of memory is available for run-time data structures, although configurations with as little as 512k have been tested, and exhibit only a small loss in compression. The method is symmetrical in that the decompressor is required to have at least as much memory as the compressor.

5.1 Other Features of the CPC Method

The following additional benefits not mentioned above accrue to the CPC compression method:

- *Adaptiveness:* Because the method is adaptive, compression performance improves on longer documents, often dramatically.

- *Sequential access:* The method allows for the decompression of pages in a document prior to the transmission of the entire document. For long documents, this allows the initial pages of the document to be decompressed while subsequent pages are still in transmission.

- *Random access:* In a random access environment, a small (on the order of a byte per page) auxiliary index can be maintained. Using this index, any page in the document can be rendered by decoding only the description of that page, and possibly a small number of surrounding pages. This can be a significant performance benefit for long documents, bounding the access time to access any particular page.

- *Image enhancement:* The composition of multiple source characters into a single template character can improve the appearance of the reconstructed image relative to the original. The composition process effectively recovers high-frequency information which was lost due to quantization error.

- *Facilitation of optical-character recognition:* By reducing the number of items that need to be recognized — from the number of characters to the far smaller number of templates — and improving their appearance, the compression method may make an ideal front end for OCR systems.

- *Facilitation of image editing:* As the image in compressed format is already decomposed into characters in approximate reading order, visual editing of the image at a character level (as, for instance, in the Xerox image editor) may be facilitated.

6 Conformance to Functional Requirements

This section describes the conformance of the CPC method to the requirements listed in Appendix A of the Solicitation of Contributions.

The CPC method is a compression algorithm for bi-level images as defined in the functional requirements. Any bi-level image, including but not limited to scanned and computer-generated images containing text, line-drawing, and half-tone images can be processed by this method.

Implementability has been demonstrated by the existence of a complete software implementation. This implementation, written in highly portable C, has been ported to several architectures, including Motorola 680x0, Intel x86, MIPS, SPARC, and others, and works interoperably on all architectures. Our current implementation of the algorithm has been voluminously described in a full-length manuscript incorporating the code.

Coding efficiency over a broad range of documents shows that the method exhibits typical compression ratios many times better than the CCITT standard methods. (For pure text documents, compression ratios as much as twenty times better than Group 4 have been observed.) To date, no images have been found for which the method exhibits worse compression than these methods. Comparisons with JBIG have been more limited. However, compression ratio on primarily text documents is, in our experience, again far superior to JBIG, and on half-toned images ranges from comparable to substantially better.

The algorithm is symmetrical, in the sense that memory requirements for compression and decompression are identical, complexity is similar, and speed is roughly comparable. (Decompression speed tends to be about a factor of two faster than compression speed for text documents; decompression speed tends to equal compression speed for half-toned images.)

7 Performance of the Algorithm

We report here the performance of the algorithm in terms of compression ratio, speed, memory requirements, and image quality. The CPC method has been designed as a lossy encoding method that is, in the terminology of the Solicitation of Proposals "perceptibly lossless". Consequently, we report only performance as per item 2A in the Solicitation of Proposals. (There are no algorithmic parameters for manipulating image quality.) Table 1 provides performance data on the Cartesian Perceptual Compression method (version 3.2.6), along with comparisons to standard CCITT methods.

Compression ratio is reported in bytes. The CPC-compressed format includes no header bits; all sizes reported are for entropy-coded data. Nonetheless, no further information is necessary to reconstruct the source page images.

Compression and decompression times are reported in seconds. Timings were taken from a software implementation running on a Motorola 25 megahertz 68040, under the NextStep operating system (version 2.1). No special hardware or other processing facilities were used. (Compression times include the time to decompress the source images from their TIFF compression format.)

Memory requirements for the method are fixed. The tests reported here were conducted using the default 1 megabyte memory configuration. Equal amounts of memory are required at both the compressor and decompressor.

To assess image quality, electronic versions of the reconstructed images are being made available electronically. We believe that the reconstructed images show no perceptible degradation of the source image.

Document Name	DPI	Number of Chars	Templates	Compression Time	Decomp	Size	Ratio	Compared to G3	G3-2D	G4
				CCITT documents						
f01	200	1098	191	1.76	.6	4547	111.11	7.32	5.94	3.65
f02	200	223	122	1.87	2.1	7869	64.2	4.22	3.04	1.33
f03	200	1459	311	3.67	1.8	9270	54.5	6.28	4.74	2.76
f04	200	4328	326	6.05	1.4	10274	49.18	9.34	7.96	6.23
f05	200	2047	286	3.61	1.7	8317	60.75	7.31	5.67	3.52
f05	300	2160	326	5.25	2.9	12077	94.11	8.91	6.52	3.65
f05	400	2297	367	6.78	4.5	16713	120.89	10.02	7.08	3.64
f05	600	3626	563	12.09	9.3	33998	133.72	8.95	6.17	3.02
f06	200	513	252	3.47	3.2	10647	47.45	4.59	3.23	1.48
f07	200	3509	810	7.05	2.2	21366	23.65	4.82	4.06	3.11
f07	300	3817	714	8.47	3.0	26046	43.64	6.85	5.42	3.74
f07	400	4034	758	10.62	4.4	33526	60.27	8.16	6.16	3.91
f07	600	3559	585	8.33	5.7	38391	51.23	5.7	4.14	2.43
f08	200	431	374	3.41	3.4	13619	37.1	4.52	3.07	1.32
				Half-tone chart						
f10	200	3360	1708	10.89	5.5	49487	10.2	3.05	3.00	2.87
f10	300	4384	1782	20.77	12.5	111476	10.05	3.21	3.40	3.53
				Sailboat						
f11	400	1791	29	39.45	29.2	103476	7.6	9.11	9.63	10.49
f12	400	1954	165	31.94	26.3	149829	5.25	4.92	4.59	4.25
f13	400	1561	65	43.45	31.2	99984	7.87	11.46	16.44	15.23
f14	400	3466	18	39.61	28.0	83902	9.37	11.57	13.73	15.84
				Magazine text — half-tone/composite						
f16	800	11500	2942	36.41	30.6	170025	11.22	2.4	1.82	1.21
f17	400	200	54	46.83	41.1	185224	9.02	4.5	5.02	5.38

Table 1: Compression performance of Cartesian Perceptual Compression on the ISO test images. Times are CPU times in seconds on a Motorola 25 megahertz 68040. Sizes are in bytes. Comparisons give the compression ratio of CPC-compressed images compared to the same image compressed using CCITT standard compression methods.