ISO/IEC JTC 1/SC 29/WG1

N1014

Date: 1998-10-

20

ISO / IEC JTC 1 / SC 29 / WG 1
(ITU-T SG8)

Coding of Still Pictures

JBIG	**JPEG**
Joint Bi-level Image Experts Group	**Joint Photographic Experts Group**

TITLE: **Report on Core Experiment CodEff26:**
JBIG-Like Coding of Bi-Level Image Data in JPEG-2000

SOURCE: **Faouzi Kossentini (faouzi@ece.ubc.ca) and Dave Tompkins**
Department of ECE, University of BC, Vancouver, BC, Canada V6T 1Z4
Soeren Forchhammer (sf@hpl.tele.dtu.dk), Bo Martins and Ole Jensen
Department of Telecommunication, TDU, Lyngby, Denmark, DK-2800
Ian Caven, Image Power Inc., Vancouver, BC, Canada V6E 4B1
Paul Howard, AT&T Labs, NJ, USA

PROJECT: **JPEG-2000**

STATUS: **Core Experiment Results**

REQUESTED ACTION: **Discussion**

DISTRIBUTION: **14th Meeting of ISO/IEC JTC1/SC 29/WG1**

Contact:
ISO/IEC JTC 1/SC 29/WG 1 Convener - Dr. Daniel T. Lee
Hewlett-Packard Company, 11000 Wolfe Road, MS42U0, Cupertion, California 95014, USA

Tel: +1 408 447 4160, Fax: +1 408 447 2842, E-mail: Daniel_Lee@hp.com

1. Update

This document updates results previously reported in WG1N863. Specifically, the upcoming JBIG-2 Verification Model (VM) was used to generate the compression rates in Table 1. These results more accurately reflect the expected bit rates, and include the overhead required for the proper JBIG-2 bitstream formatting. Results may be better in the final implementation, but these results are guaranteed to be attainable with the current VM. The JBIG-2 VM will be available in December 1998.

2. Summary

This document presents experimental results that compare the compression performance several JBIG-like coders to the VM0 baseline coder. The images CCIT1 and CCIT2, taken from the JPEG-2000 test set, and the images f03_200 and f04_200, taken from the JBIG-2 test set, are used in our experiments. Lossless coding results for JBIG-1, MQ-based JBIG-1, JBIG-2 generic coding, JBIG-2 symbol coding (SPM) and VM2.1 coding, and lossy coding results for JBIG-1 (with resolution reduction), lossy CSM, JBIG-2 generic coding, JBIG-2 symbol coding and VM2.1 coding at several bit rates, are presented. The lossless results obtained with the JBIG-like coders are, in all cases, better than those of the VM2.1. Moreover, the JBIG-like coders outperform substantially the VM2.1 at most bit rates.

3. Background

Most of the bi-level image data consists of text, halftone and line art. Basically, we can segment a bi-level image into different class of image data. We propose to use JBIG-like coders with different modes to encode each class of the image data. Some data, like line art data, will be encoded by a cleanup coder, essentially a basic bitmap coder like JBIG-1. Textual data is coded by pattern matching and substitution (CSM and SPM), with an additional refinement step constituting the soft pattern matching (SPM) method. The basic idea of CSM/SPM is that, instead of coding all the pixels of each occurrence of each character, we code the bitmap of one representative instance of the character and put it into a ``dictionary''. Then for each character on the page, we code both a pointer to the corresponding representative bitmap in the dictionary and the position of the character on the page. In addition to that, a refinement stage can be included to recreate the original character on the page with enough accuracy so that substitution errors are unlikely. Halftone data is coded with template-based JBIG-like coder that employs adaptive pixels, hoping to capture the periodicity structure of the halftones. For more information about the above coding components and others, please refer to the WG1 document N692 , the JBIG standard and the references [1-4].

4. Experimental Results

Table 1 presents the lossless results obtained using JBIG-1, modified JBIG-1 (using the MQ coder), JBIG-2 generic coding (template coding), JBIG-2 symbol coding (SPM) and the VM2.1. The JBIG-2 generic coder employs a 16-pixel template. Lossless SPM is based on pattern matching and refinement stages. For CCIT2, the SPM coder segmented the image and coded some regions with a 16-pixel template. Notice that the JBIG-like coders outperform the VM2.1 by 20%-75% in compression performance.

Encoding Method	CCIT1 (bpp)	CCIT2 (bpp)	ITU F03_200 (bpp)	ITU F04_200 (bpp)
JBIG-1	0.045	0.130	0.0395	0.097
JBIG-1 (using the MQ coder)	0.040	0.101	0.038	0.091
Template coding (JBIG-2 Generic)	0.040	0.097	0.038	0.091
SPM (JBIG-2 Symbol coding)	0.038	0.095	0.034	0.074
VM2.1	0.050	0.164	0.0420	0.100

Table 1. Lossless Coding Results

Table 2 presents the lossy results obtained with JBIG-1 (with resolution reduction), Lossy CSM (JBIG-2 Symbol coding), lossy SPM (JBIG-2 Symbol coding), template coding (JBIG-2 Generic coding) and the VM2.1 at several bit rates (bpp). The symbol %dist indicates the percentage of flipped pixels, and the symbol qual indicates quality, which is expressed according to the following scale:

- v.good: Almost distinguishable from the original image
- good: Objects and text appear to have slightly jagged edges as compared to the original image
- OK: Indicates that e.g. text is readable but appear rather jagged as compared to the original image
- poor: means that e.g. text is very difficult or almost impossible to read.
- v.poor: text is unreadable and other objects can be barely identified.

For JBIG-1, the resolution reduction is performed such that the resulting image width and height are reduced with powers of 2 (no adaptive template pixel was invoked). Lossy CSM performs hard pattern matching, while lossy SPM performs pre-processing and soft pattern matching. The 16-pixel template coder employs 4 adaptive pixels, and it also introduces rate-controlled pixel flipping to achieve a specific bit rate (at the expense of additional encoder complexity). The VM2.1 coder was run with only the necessary command line parameters (with –u1). Note that rate control is not possible in many of the selected coders, and that the resulting rate closest to the target rate is selected. A meaningful comparison with the VM2.1 is, however, still possible.

It is clear that the JBIG-like coders outperform the VM2.1 coder by 1.5:1 to 3:1 in bit rate for the same distortion, expressed in terms of percentage of pixel errors and/or subjective quality. Moreover, the JBIG-2 coders perform similarly, and they mostly outperform the resolution-reduction-based JBIG-1 coders.

Encoding Method	CCIT1 (bpp/%dist/qual)	CCIT2 (bpp/%dist/qual)	ITU F03_200 (bpp/%dist/qual)	ITU F04_200 (bpp/%dist/qual)
JBIG-1 (res.red=2)	0.0021/2.41/good	0.048/9.29/good	0.019/1.58/OK	0.044/3.26/OK
JBIG-1 (res.red=4)	0.0098/5.69/OK	0.022/15.87/poor	0.007/3.88/v.poor	0.019/7.77/v.poor
JBIG-1 (res.red=8)	0.0044/10.68/poor	0.008/18.91/v.poor	0.003/7.09/v.poor	.0061/12.5/v.poor
Lossy CSM (JBIG-2 Symbol)	0.043/0.10/v.good	0.101/0.02/v.good	0.024/1.17/v.good	0.032/1.45/v.good
Template coding (JBIG-2 Generic)	0.03125/0.3/v.good	0.06085/1.1/v.good	.03125/.14/v.good	.085/0.15/v.good
Lossy SPM (JBIG-2 Symbol)	0.03125/x.x/v.good	0.085/x.x/v.good	0.0173/x.x/v.good	0.0303/x.x/v.good
VM2.1	N/A	0.125/4.55/v.good	N/A	N/A
VM2.1	N/A	0.111/4.93/good	N/A	N/A
VM2.1	N/A	0.085/5.85/OK	N/A	.085/2.8/good
VM2.1	N/A	0.0625/6.58/poor	N/A	.0625/3.65/poor
VM2.1	.03125/2.41/OK	.03125/10.1/v.poor	.03125/1.55/good	.03125/5.9/v.poor

Table 2. Lossy Coding Results

5. Complexity Analysis

This section provides a basic analysis for one of the more complicated coding schemes, the Soft Pattern Matching (SPM) algorithm. Due to time constraints, a detailed analysis of the all the considered algorithms is not possible. Such will be provided after the up-coming WG1 meeting.

Even within the confines of a SPM, there are several different approaches which can effect the complexity of the algorithm. For example, a symbol that will appear in the final image can be encoded directly, as an aggregate of several smaller symbols, as a refinement of a different symbol, or be simply encoded in a residue. Each strategy will have similar computation complexities, but will have different memory (buffer) requirements. In general, there are four components to the SPM bit stream: 1) the Symbol Table, 2) the Symbol Instance Table, 3) the Refinement Information (image information not represented as a symbol) and 4) the Overhead. What becomes evident in the breakdown is how dependent the size of the bit stream is on the number and type of symbols. In fact, this will make the analysis of the SPM complexity very difficult. Instead of having an algorithm of order O(n) where n represents the size of the image, it will be O(n + m) where m is the number of symbols in the image.

The Symbol Table will require a sizable buffer to store all of the information required, whereas the Instance Table and the Refinement Information can be read directly to the output bit stream, with only some small buffers required. The Arithmetic coder used by all of the components can also have some significant buffer requirements, but that is dependent on the parameters of the bit stream, and can be adjusted to reduce buffer requirements. To significantly reduce buffer requirements, Huffman encoding can be used for most of the components. Shown below is the signal flow graph:

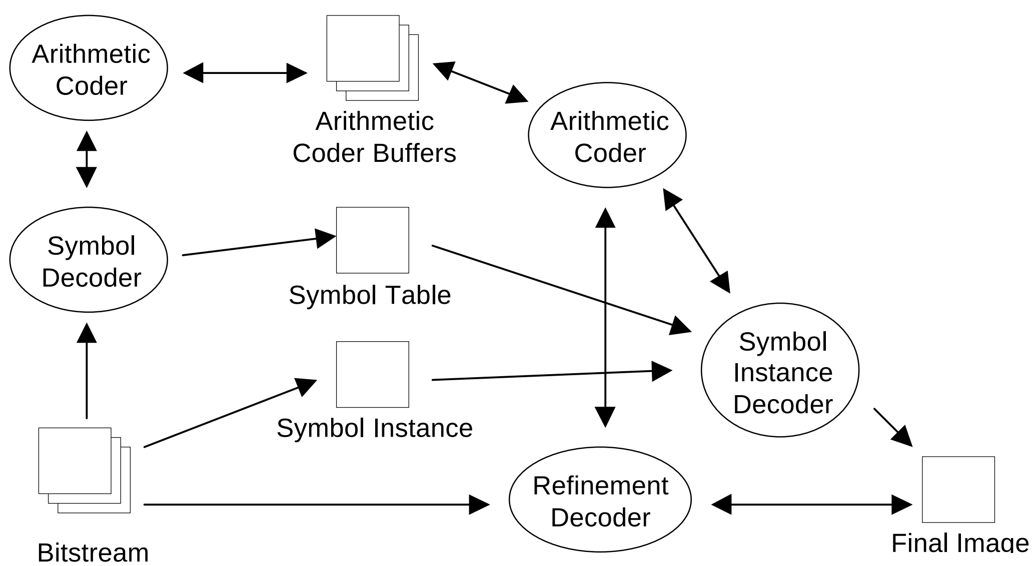

Figure 5-1: Algorithmic Modeling of JBIG-2 SPM Decoder (Overview)

To simplify the diagram for the SPM decoder, only the major buffers, control flows and function points are illustrated. There are numerous smaller buffers, function points and overheads involved with decoding a JBIG-2 SPM bit stream.

Finally, it is important to note that the memory requirements can be independent of image size, and relate to the number of symbols. Encoder complexity can differ greatly with the sophistication of the encoder, but it should be noted that the encoder complexity will also be affected by the number of symbols in the image.

6. References

[1] W. Pratt, P. Capitant, W. Chen, E. Hamilton and R. Walls, "Combined Symbol Matching Facsimile Data Compression System," Proceedings of the IEEE, vol. 68, pp. 786-796, July 1980.

[2] I. Witten, A. Moffat and T. Bell, "Managing Gigabytes: Compressing and Indexing Documents and Images", New York: Van Nostrand Reinhold, 1994.

[3] P. Howard, "Text Image Compression using Soft Pattern Matching", Computer Journal, 40:2-3, 1997.

[4] B. Martins and S. Forchhammer, "Lossless, near-lossless and refinement coding of bi-level images," IEEE transactions on Image Processing, accepted for publication (1998).