AN INNOVATIVE LOSSLESS COMPRESSION METHOD FOR DISCRETE COLOR IMAGES

Presented by:
NAIR SREEJA VENUGOPALAN[EPALEIT068]
Guided by:
Asst.Prof. SAFEER BABU T

Department Of Information Technology

GEC Sreekrishnapuram

February 27, 2015

CONTENTS

CONTENTS

INTRODUCTION

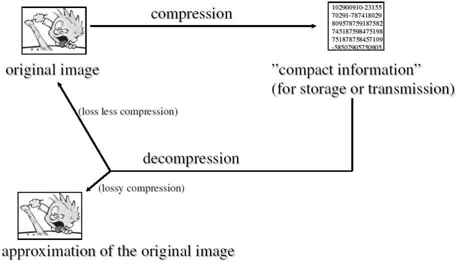
- Compression is a way to reduce the number of bits in a frame without degrading it.
- Image compression is the application of data compression on digital images.
- The objective is to reduce redundancy of the image data in order to store or transmit data in an efficient form.

Why compression is needed?

- To preserve the storage space.
- To reduce transmission cost/latency/bandwidth.
- To avoid redundancy.
- Compression ratio is the ratio of original data rate to the encoded data rate.

INTRODUCTION continued..

Compression are of two types:



Lossless

- Preserves all information.
- Exploits redundancy in data.
- Applied to general data.

Lossy

- May lose some information.
- Exploits redundancy and human perception.
- Applied to audio, image, video.

CONTENTS

Hybrid compression method based on block encoding[1]

- Hierarchical block coding is used.
- Predictive modeling has been employed to construct an error image.
- It is the difference between the predicted and original pixel values.
- Then, the error image is compressed using Huffman coding of bit patterns.

Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression [2]

- It is based on the hierarchical prediction and context-adaptive arithmetic coding.
- The input color image is first transformed into color space using color transform method.
- After the color transformation, the luminance channel Y is compressed by a conventional lossless image coder.
- Pixels in chrominance channels are predicted.
- Here arithmetic coding is applied to the error signal.

LITERATURE SURVEY

A New Efficient Algorithm for Lossless Binary Image Compression[3]

- It is based on Direct Redundancy Elimination and Improved Arithmetic Coding.
- The Adaptive Arithmetic Coder updates the probabilities immediately after each symbol is encoded.
- The Context Modeling is used to calculate the probability for each incoming symbol resides in.
- It improves the efficiency.

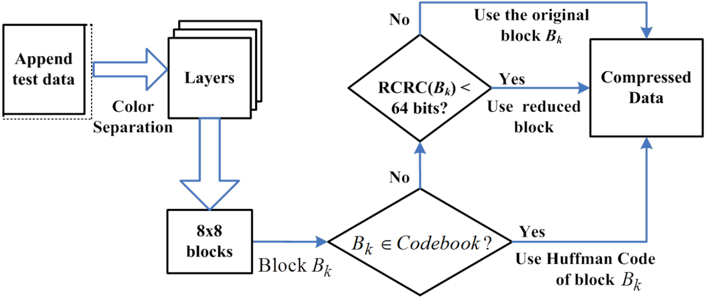
CONTENTS

There are three components in this method.

1. Preprocessing
2. A universal Huffman codebook
3. The Row-Column Reduction Coding

DIAGRAM OF THE COMPRESSION METHOD

General diagram of the proposed compression technique



PREPROCESSING

- The first phase is trimming the margins.
- Removes redundant background frame.
- It avoids biasing distribution of 0-valued or 1-valued 8x8 bocks.
- The second phase is modifying the dimensions of image divisible by 8.

THE CODEBOOK MODEL

- The data sample of 120 images are selected.
- Perform a frequency analysis on 8x8 blocks.
- The blocks that occur more than once are selected.
- For unique blocks,calculate entropy value.
- Build correct Huffman codes for the most frequent blocks.
- The resulting codebook is a fixed-to-variable dictionary containing 6952 entries.

How is the codebook employed?

- Search the codebook for each 8x8 block of a given source image.
- The X-by-Y image is partitioned into XY/64 8x8 blocks.
- If the block exists in the codebook, compress using the corresponding Huffman code.

An example of the codebook structure:

8x8 Block	Probability	Huffman Code
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	50.37%	0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	26.33%	10
1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	0.30%	11101101

The codebook entropy

This table shows the effect of block dimensions on entropy and the expected sample size

Block	*Alphabet Size*	*Entropy*	*% Max. Compression*	*E[T]*
2x2	16	1.36	66.00	55
4x4	65536	2.12	86.74	764647
8×8	2^{64}	4.09	93.60	10^{20}
12x12	2^{144}	7.89	94.52	10^{45}
16x16	2^{256}	9.72	96.20	10^{79}

- The average Huffman code length is calculated as:

$$L = \sum_{i=1}^{N} q_i \log_2 \frac{1}{q_i}$$

- The average Huffman code length is 4.094.
- Thus, the compression limit is (64 - 4.09)/64 = 93.60

The ROW-COLUMN REDUCTION CODING (RCRC)

- It operates on 8x8 blocks.

Uses two reference vectors

1. The Row Reference Vector (RRV).
2. The Column Reference Vector (CRV).

- It is used to remove the redundancy in the blocks.
- Checks whether the two consecutive row vectors are identical.
- If rows are identical, one is eliminated and the block is reduced by one row.
- If not, the next two consecutive row vectors are compared.
- The row reduction operation continues until the end of the block.

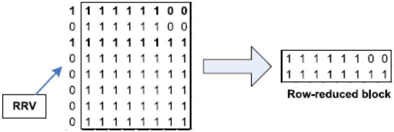
The ROW-COLUMN REDUCTION CODING continued..

- The column reduction operation is similar and elimination operations are stored in the CRV.
- The output of RCRC is a bit stream containing the RRV, CRV, and the reduced block
- It is Concatenated as S = RRV+CRV+RB
- Minimum length of S = 17 bits.

The ROW-COLUMN REDUCTION CODING continued..

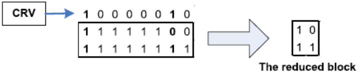
RCRC EXAMPLE..

Row elimination is performed on the block:



- Row 1 eliminates row 2.
- Row 3 eliminates all other rows.

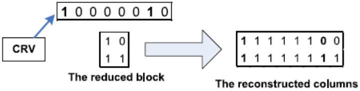
  Column elimination is performed on the row-reduced block:



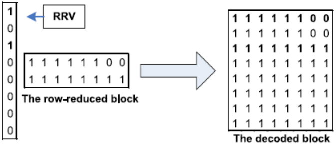
- The compressed bit stream for this example is: 10100000100000101011.

Reconstruction of the original block

- Column reconstruction based on CRV



- Row reconstruction based on RRV



CONTENTS

IMPLEMENTATION METHOD

The implementation coding scheme is shown in the table.

	Block encoding bits	**Description**
Case 1a	'11'	For the block with the shortest Huffman code in the codebook
Case 1b	'00' + 5 bits + Huffman Code	For other blocks found in the codebook
Case 2	'01' + RRV + CRV + RB	For blocks compressed by RCRC
Case 3	'10' + 64 bits	For uncompressed blocks

Table: The Coding Process

CONTENTS

Analytical Time Complexity

- The codebook contains fixed entries.
- RCRC is executed on fixed, 8x8 blocks.
- The variable input is the source image size.
- Time complexity is O(XY), where X and Y are the image dimensions.

Set of binary image samples used for testing:

EMPIRICAL RESULTS

This table shows empirical results for 15 solid binary images.

Image	*Dimensions*	E_p	E_p^*	*AC*	JBIG2
059	200x329	88.99	93.72	94.58	88.58
071	545x393	90.72	93.75	94.22	89.82
074	203x247	86.9	92.66	93.16	87.68
075	790x480	92.78	96.5	96.25	94.8
076	245x226	86.24	92.75	93.41	86.82
077	450x295	88.47	95.65	95.38	94.83
079	245x158	85.35	91.42	91.96	84.91
080	491x449	91.86	95.71	95.86	92.17
081	245x248	89.2	92.84	93.3	86.69
082	491x526	92.21	96.33	96.08	94.31
083	354x260	88.93	95.29	95.48	92.24
085	167x405	86.9	92.55	93.62	87.7
086	335x500	91.12	95.88	95.62	94.97
087	447x459	89.89	96.2	95.73	93.86
090	350x357	86.68	95.02	94.8	92.18
	Average	**90.36**	**95.26**	**95.27**	**92.43**

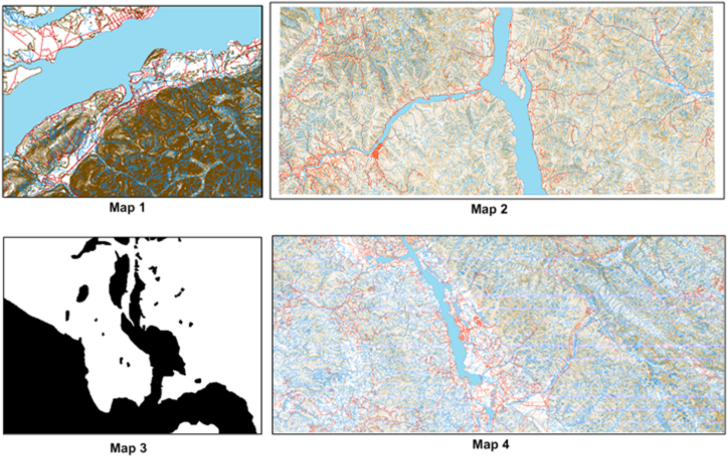
EMPIRICAL RESULTS

This table shows the percentage of blocks compressed by the proposed method.

Image	*Codebook*	*RCRC*	Incompressible
059	96.7	3.3	0
071	95.48	4.32	0.2
074	95.04	4.47	0.5
075	98.53	1.46	0.02
076	95.44	4.56	0
077	98.62	1.23	0.14
079	94.19	5.65	0.16
080	98.73	1.25	0.03
081	94.96	4.84	0.2
082	98.9	1.03	0.07
083	98.52	1.28	0.2
085	95.42	4.39	0.19
086	98.49	1.4	0.11
087	99.38	0.52	0.09
090	98.08	1.87	0.05

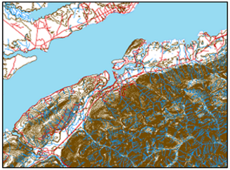
EMPIRICAL RESULTS

This figure shows the topographic maps used for testing.

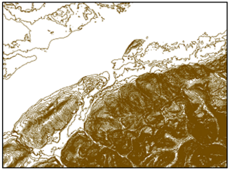


EMPIRICAL RESULTS

This figure shows the layer separation of a topographic map.

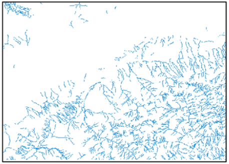


4 different colors



Layer 1: Contour Lines

Layer 2: Lakes



Layer 3: Rivers

Layer 4: Roads

EMPIRICAL RESULTS

This table shows the description of selected topographic map images.

Map	*Dimensions*	Size (KB)
1	2200	1700
2	5776	13056
3	5112	11600
	Total Size	**406708**

This table shows the compression results for map images using the proposed method.

Map	*Compressed Size (KB)*	*Compression Ratio (bpp)*	JBIG2 (bpp)
1	210.77	0.019	0.029
2	7489.27	0.034	0.052
3	6626. 27	0.038	0.055
Total	**14326.42**	**0.035**	**0.053**

CONTENTS

- An innovative method for lossless compression is introduced.
- It has Low-complexity.
- Provides high-efficiency.

This method has been successfully implemented on two major image categories.

1. Images that consist of a predetermined number of discrete colors such as digital maps,graphs.
2. Binary images.

The results of a large number of test images show that this method has greater compression ratio.

CONTENTS

REFERENCES

[1] P. Franti and O. Nevalainen, "Compression of binary images by composite methods based on block coding," J. Vis. Commun. Image Represent., vol. 6, no. 4, pp. 366-377, 1995.
[2] Seyun Kim and Nam Ik Cho, "Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression," IEEE Trans. Image Process., vol. 23, no. 1, pp. 445-449, Jan. 2014.
[3] L. Zhou and S. Zahir, "A New Efficient Algorithm for Lossless Binary Image Compression," Signal Process., Image Commun., vol. 6, no. 1, pp. 69-76, Mar. 1994.
[4] S. Zahir and A. Borici, "A fast lossless compression scheme for digital map images using color separation," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Mar. 2010, pp. 1318-1321.
[5] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," IEEE Trans. Commun., vol. 45, no. 4, pp. 437-444, Apr. 1997.

THANK YOU...:)