A New Efficient Algorithm for Lossless Binary Image Compression

Lele Zhou
Image Processing, Graphics, and Multimedia Lab
Computer Science Department
The University of Northern British Columbia
e-mail: zhoul@unbc.ca

Saif Zahir
Image Processing, Graphics, and Multimedia Lab
Computer Science Department
The University of Northern British Columbia
e-mail: zahirs@unbc.ca

Abstract

Binary image compression is desirable for a wide range of applications, such as digital libraries, map archives, fingerprint databases, facsimile, etc. In this paper, we present a new highly efficient algorithm for lossless binary image compression. The proposed algorithm introduces a new method, Direct Redundancy Elimination, to efficiently exploit the two-dimensional redundancy of an image, as well as a novel Dynamic Context Model to improve the efficiency of arithmetic coding. Simulation results show that the proposed algorithm has comparable compression ratio to JBIG standard. In many cases, the proposed algorithm outperforms the JBIG standard.

Keywords— *binary image; lossless compression; context modeling; arithmetic coding*

1 Introduction

Binary image compression is desirable for a wide range of applications. During the last decades, a number of algorithms have been developed, such as Huffman Coding [1][2], Run Length Coding [3][4], Arithmetic Coding [5][6], and geometric based coding [7], etc. In this paper, we present a new highly efficient algorithm for lossless binary image compression. The proposed algorithm consists of two modules: (1) Direct Redundancy Elimination (DRE); (2) Improved Arithmetic Coding (IAC); as shown 1.

2 Direct Redundancy Elimination (DRE)

Direct Redundancy Elimination efficiently exploits the two-dimensional redundancy of an image. It removes redundant rows and columns of pixels. Binary vectors are generated to indicate the exact locations of removed redundant rows and columns of pixels. These binary vectors are referred to as reference vectors. DRE comprises three parts: (i) Margin Elimination; (ii) Macro Redundancy Elimination; (iii) Micro Redundancy Elimination.

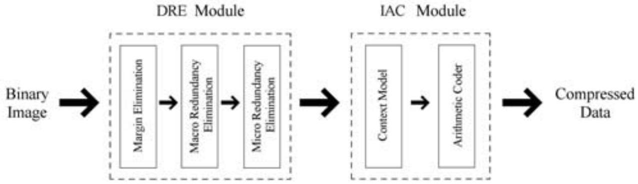

Figure 1: the Proposed Algorithm

2.1 Margin Elimination

Binary images are likely to have blank spaces bordering the objects of the image. The blank spaces can be either black or white and they are defined as margins in this paper. Margin Elimination is the process of cropping out the margins, which results in a new image. The coordinates to specify where the new image lies must be saved for the reconstruction process.

2.2 Macro Redundancy Elimination

Macro Redundancy Elimination exploits the redundant spaces between consecutive rows or columns of pixels. Macro Redundant Rows and Columns are considered as two different redundant spaces. They are defined as follows.

Let I be a binary image, $I = (R_1, R_2, \cdots, R_n)$, the size of I is $n \times m$. $R_i \in \{R_1, R_2, \cdots, R_n\}$ is a row of pixels. A Macro Redundant Row is defined as R_{marri}, and

$$marri = \{i \mid R_i = R_{i-1}\}, \quad 1 < i \leq n \tag{1}$$

in words, if row R_i is identical to its above neighbouring row R_{i-1}, we say R_i is a Macro Redundant Row.

Likewise, Let $I = (C_1, C_2, \cdots, C_m)$, and $C_j \in \{C_1, C_2, \cdots, C_m\}$ is a column of pixels. A Macro Redundant Column is defined as C_{marci}, and

$$marci = \{j \mid C_j = C_{j-1}\}, \quad 1 < j \leq m \tag{2}$$

in words, Column C_j is a Macro Redundant Column if C_j is identical to its left neighbouring column C_{j-1}.

A Macro Redundant row or column should be removed upon its identification. This process can be reversed by duplicating the preceding rows or columns. In order to do that, reference vectors are generated to indicate the exact location of Macro Redundant Rows or Columns. The following schemes are presented to construct the reference vectors V_{row} and V_{column}.

$$V_{row}(i) = \begin{cases} 0 & \text{if } i \in marri \\ 1 & \text{otherwise} \end{cases} \qquad 1 \leq i \leq n \tag{3}$$

$$V_{column}(j) = \begin{cases} 0 & \text{if } j \in marci \\ 1 & \text{otherwise} \end{cases} \qquad 1 \leq j \leq n \tag{4}$$

$V_{macro} = (V_{row}, V_{column})$ will be further processed via IAC module in Section 3.

2.3 Micro Redundancy Elimination

Micro Redundancy Elimination removes the redundant rows and columns within a block of an image. This part of the algorithm is important because the size of an image is tremendously reduced, yet the algorithm produces a “stair” phenomenon which is exploited by the proposed Dynamic Context Model as explained Section 3. Micro Redundancy consists of two types of redundant spaces: (i) Micro Redundant Rows; (ii) Micro Redundant Columns.

2.3.1 Micro Redundant Rows

Since Micro Redundant Rows are defined within a block of an image, an image needs to be partitioned into a number of regions which are referred to as blocks in this paper.

Consider the image $I_{motorcycle}$ with the size of $n \times m$ in Figure 2. It is columnwisely partitioned into a number of blocks. The size of each block is $n \times k$ if m is divisible by k, where $0 < k \leq m$. If m is not divisible by k, we have a remainder block which is the rightmost block of the image. Although the size of the remainder block is different from the rest of the blocks, it is treated the same in the subsequent steps. Thus, $I = (B_1, B_2, \cdots, B_u)$ and $B_x \in \{B_1, B_2, \cdots, B_u\}$.

Block B_x in Figure 3 is a composition of a number of rows, $B_x = (R_1, R_2, \cdots, R_n)$. $R_i \in \{R_1, R_2, \cdots, R_n\}$, is a row in block B_x. A Micro Redundant Row is a row identical to its above neighbouring row in the block, which is defined as R_{mirri}, and

$$mirri = \{i \mid R_i = R_{i-1}\}. \tag{5}$$

For example, in Figure 3, R_6 and R_7 are Micro Redundant Rows, because $R_6 = R_5$ and $R_7 = R_6$. Micro Redundant Rows need to be removed.

For decompression purposes, reference vectors are generated to indicate the exact locations of Micro Redundant Rows. Hence, there is a reference vector V_x for every corresponding block B_x.

$$V_x(i) = \begin{cases} 0 & \text{if } i \in mirri \\ 1 & \text{otherwise} \end{cases} \qquad 1 \leq i \leq n \tag{6}$$

$V_{micro\ row} = (V_1, V_2, \cdots, V_u)$ will be further compressed via IAC module.

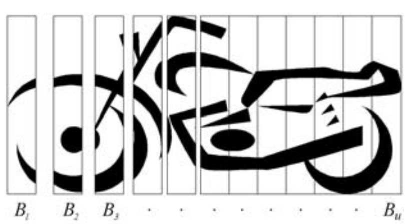

Figure 2: Columnwise Partitioning on Image $I_{motorcycle}$

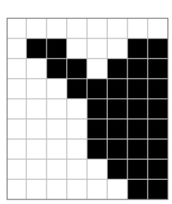

Figure 3: Block B_i $size = n \times k$

2.3.2 Micro Redundant Columns

The resulting bitmap from the previous step is further reduced by eliminating the Micro Redundant Columns. Micro Redundant Columns are those columns which are exactly identical or partially identical to their left neighbouring columns.

From the previous step, an image is columnwisely partitioned into a number of blocks: $I = (B_1, B_2, \cdots, B_u)$. For every block $B_x \in \{B_1, B_2, \cdots, B_u\}$, B_x is further partitioned rowwisely into a number of sub-blocks, which is $B_x = (b_1, b_2, \cdots, b_v)$ and $b_y \in \{b_1, b_2, \cdots, b_v\}$. For example, consider B_3 in Figure 4, it is partitioned into three sub-blocks, which are $B_3 = (b_1, b_2, b_3)$.

A sub-block b_y can be considered as a composition of a number of columns, which is $b_y = (C_1, C_2, \cdots, C_k)$ and $C_j \in \{C_1, C_2, \cdots, C_k)\}$, where k is the number of columns within the sub-block b_y. Micro Redundant Columns have two groups: $full$ and $partial$.

A full Micro Redundant Column is a column that is exactly identical to its left neighbouring column. It is defined as C_{full},

$$full = \{j \mid C_j = C_{j-1}\}, \quad 1 < j \leq k. \tag{7}$$

Consider a sub-block b_2 in Figure 5. Column C_6 is identified as a Micro Redundant Column, because $C_6 = C_5$. As C_1 is the first column in sub-block b_2, it cannot be a Micro Redundant Column according to the definition defined in Equation 7, $1 < j \leq k$. However C_1 may be identical to its left neighbouring column which is located in the left neighbouring sub-block. Therefore, another definition is needed to manage the first column in every sub-block.

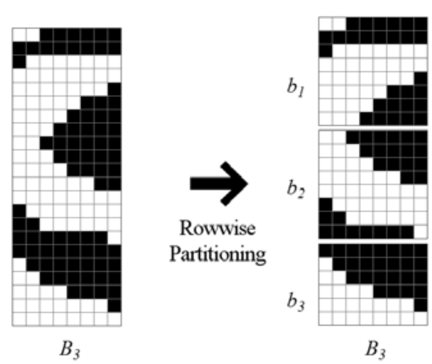

Figure 4: Rowwise Partitioning

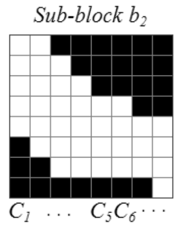

Figure 5: Full Micro Redundant Columns

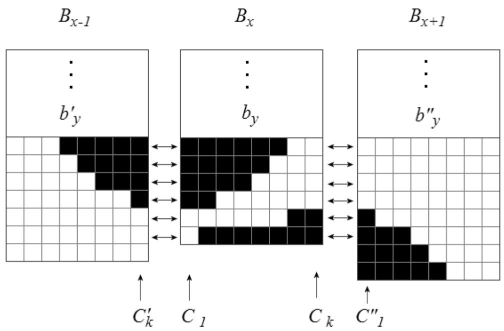

Figure 6: Partial Micro Redundant Columns

Consider three neighbouring sub-blocks b'_y, b_y, b''_y belong to three different blocks B_{x-1}, B_x and B_{x+1} in Figure 6. The length of C_1, denoted as $l(C_1)$, is less than or equal to the length of C'_k, $l(C_1) \leq l(C'_k)$, and each entry of C_1 is identical to each entry of C'_k in its corresponding position, denoted as $C_1 = C'_k(1 : l(C_1))$. Then C_1 is considered as a partial Micro Redundant Column and will be removed. This is because, C_1 can be fully recovered by duplicating the entries of its neighbouring column C'_k until it reaches its length. Notice that $l(C''_1) \nleq l(C_k))$, as a result, C''_1 is not considered a partial Micro Redundant Column. This is because C''_1 cannot be fully recovered if it is removed as a partial Micro Redundant Column.

Therefore, the partial Micro Redundant Column is defined as $C_{partial}$,

$$partial = \begin{cases} 1 & \text{if} \quad l(C_1) \leq l(C'_k) \wedge C_1 = C'_k(1 : l(C_1)) \\ \phi & \text{otherwise} \end{cases} \tag{8}$$

where, C_1 is the first column of a sub-block b_y in a block B_x, C'_k is the last column of a sub-block b'_y in a block B_{x-1}.

Hence, Micro Redundant Column can be defined as C_{mirci}

$$C_{mirci} = C_{full} \cup C_{partial}. \tag{9}$$

A reference vector V_y is generated to indicate the locations of Micro Redundant Columns for every corresponding sub-block b_y.

$$V_y(i) = \begin{cases} 0 & \text{if } C_j \in C_{mirci} \\ 1 & \text{otherwise} \end{cases} \qquad 1 \leq j \leq k \tag{10}$$

$V_{micro\ column}$ is the concatenation of all reference vectors V_y of all sub-blocks. The process of removing Micro Redundant Columns results in a reduced sub-block, denoted as Reduced Block (RB).

3 Improved Arithmetic Coding (IAC)

Improved Arithmetic Coding consists of two parts: Context Modeling and Adaptive Arithmetic Coder. In this section, we introduce a new Dynamic Context Model to improve the efficiency of arithmetic coding.

3.1 Context Modeling

The Context Modeling presented in this paper is based on the principle: the probability for each incoming symbol is calculated based on the context that the symbol resides in. In fact, the context consists of nothing more than the symbols that have been encountered. In this section, we use the proposed Dynamic Context Model to calculate the probability on the Reduced Blocks RB. On the other hand, high order Markov Models [8] are used to calculate the probability on the reference vectors: V_{macro}, $V_{micro\ row}$ and $V_{micro\ column}$.

The order of a model refers to the number of preceding symbols which make up the context. The order 1 model can be characterized by $P(X_i \mid X_{i-1})$, which is the probability distribution of X_i provided that X_{i-1} is the immediate preceding symbol. Thus order 2 can be derived from order 1, which is $P(X_i \mid X_{i-1}X_{i-2})$. The high order Markov Model refers to order 3 or above, which can be described as $P(X_i \mid X_{i-1}X_{i-2} \cdots X_{i-k})$, where k is the number of the order. Table I shows the average compression results on the reference vectors of 500 test images coded by the Adaptive Arithmetic Coder using different orders of models. The table shows that order 4, order 5 and order 3 models behave the best on reference vectors V_{macro}, $V_{micro\ row}$ and $V_{micro\ column}$ respectively. Hence, they are employed to achieve high compression ratio.

TABLE I

Compression Results Using Different Orders of Models

reference vector	V_{macro} (bits)	$V_{micro\ row}$ (bits)	$V_{micro\ column}$ (bits)
original	1133	29835	4866
order 1	347	13560	3769
order 2	274	12405	3742
order 3	262	11940	3740
order 4	260	11802	3924
order 5	261	11688	3937

A Reduced Block RB is likely to have a "stair" phenomenon as a result of Micro Redundancy Elimination. The "stair" phenomenon has four possible orientations and different sizes, one of which is shown in Figure 7. Based on these properties, a novel Dynamic Context Model is proposed. The new model calculates the probability of the

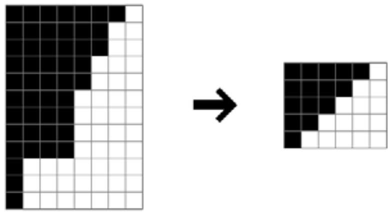

Figure 7: "stair" Phenomenon

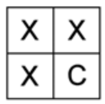

Figure 8: 3-Bits Template

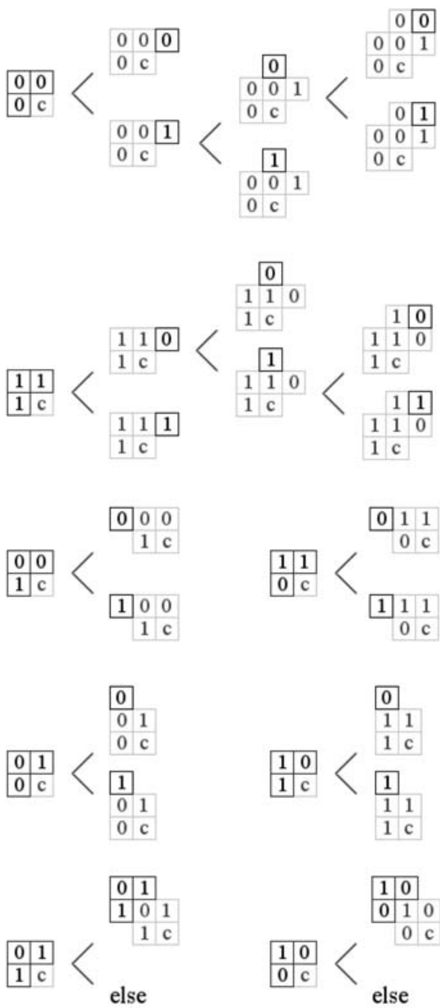

Figure 9: Dynamic Context Modeling Templates

current symbol according to the dynamically changing templates. Unlike JBIG [9] which applies a fixed 10-bits template, our method models the RB dynamically based on the possible values of the bits correlated to a 3-bits root template shown in Figure 8. The proposed model provides favorable statistics for the arithmetic coding to achieve a better compression ratio.

The Dynamic Context Model is presented in a binary tree structure in Figure 9. The probability of the current symbol c is calculated through the frequencies of c in its context divided by the cumulative freqencies of the context in where c resides. Each external node of a tree is a possible case of probability distribution. Each root is a possible derivation from the 3-bits root template. Compared to JBIG which has 1024 possible contexts and 10 bits within each context, the proposed template only has a total number of 20 possible contexts and each context varies from 4 to 6 bits. This leads to a faster modeling speed and reduction of compression time.

3.2 Adaptive Arithmetic Coder

A generic arithmetic coder is a two pass algorithm. It runs through the data sequence to obtain the probability distributions on the first pass and encodes each symbol on the second pass based on the probabilities previously obtained. This type of arithmetic coder has two drawbacks: first, two pass encoding delays the compression time; second, all probability distributions need to be stored for decoding, which results in a lower compression ratio. To avoid these drawbacks, the Adaptive Arithmetic Coder is employed. The Adaptive Arithmetic Coder is an one pass algorithm which updates the probabilities immediately after each symbol is encoded. The decoder follows the same fashion to get the exactly identical probabilities. Therefore, all probabilities are obtained independently during the process of encoding and decoding, which avoids the overhead of storing probabilities.

4 Conclusions

In this paper, a new highly efficient algorithm for lossless binary image compression is presented. This algorithm consists of two modules:(1) Direct Redundancy Elimination, which efficiently exploit the two-dimensional redundancy of an image; (2) Improved Arithmetic Coding, which presents a novel Dynamic Context Model to improve the efficiency of arithmetic coding. The proposed algorithm has been tested on numerous images of varying sizes and complexity. The simulation results shows that the proposed algorithm is significantly comparable to JBIG. Table II shows that the proposed algorithm has an overall 8% better compression ratio than JBIG on a sample of 5 test images which are shown in Figure 10.

TABLE II
COMPRESSION RESULTS ON 5 TESTED IMAGES

Files Name	Original		JBIG		Proposed Algorithm	
	Dimensions	Size (bits)	Size (bits)	Ratio	Size (bits)	Ratio
cheetah	720 × 576	414720	286750	30.85%	263810	36.39%
wallpaper	720 × 576	414720	292510	29.47%	276230	33.40%
animals	720 × 576	414720	117680	71.62%	112990	72.76%
fruit	509 × 496	252464	39864	84.21%	26725	89.41%
tower	359 × 225	80775	21112	73.86%	19055	76.41%
mother	296 × 200	59200	16800	71.62%	14915	74.81%
TOTAL		**1636599**	**774716**		**713725**	

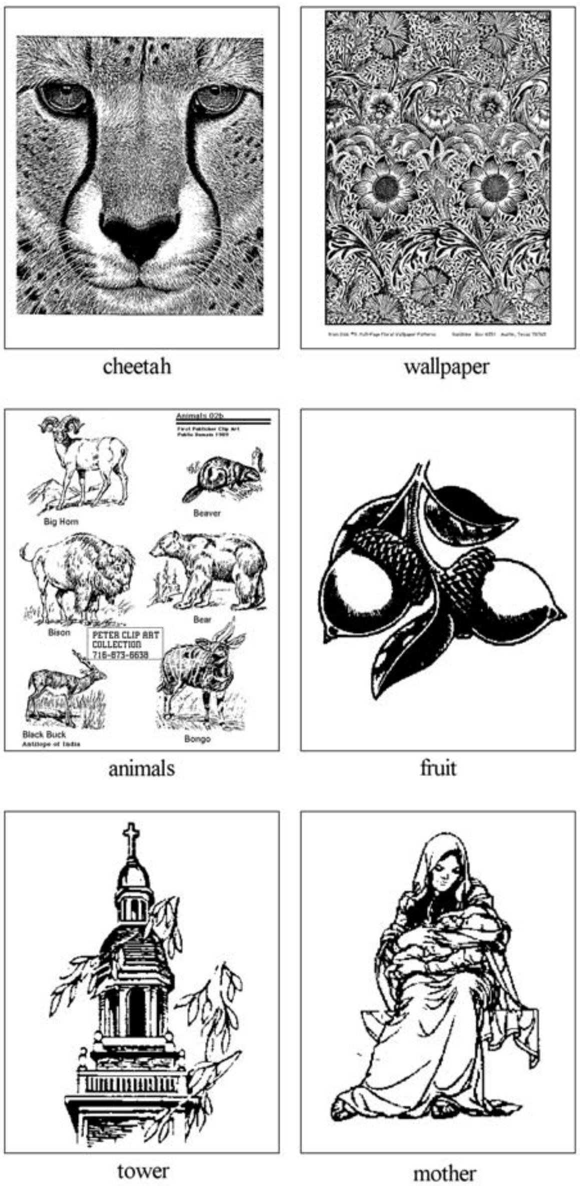

Figure 10: Tested Images

References

[1] D. A. Huffman, *"A Method for the Construction of Minimum-Redundancy Codes," Proceeding of IRE*, vol. 40, pp. 1098-1101, 1952.

[2] W. Lu and M. P. Gough, *"A Fast-Adaptive Huffman Coding Algorithm," IEEE Transactions on Communications*, vol. 41, no. 4, pp. 535-538, 1993.

[3] H. Tanaka and A. Leon-Garcia, *"Efficient Run-Length Encodings," IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 880-890, 1982.

[4] S. Golomb, *"Run-Length Encodings," IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 880-890, 1966.

[5] G. G. Langdon and J. Rissanen, *"Compression of Black-White images with Arithmetic Coding," IEEE Transactions on Communications*, vol. 29, no 6, pp. 858-867, 1981.

[6] E. Bodden, M. Clasen, and J. Kneis, *"Arithmetic Coding Revealed," Proseminar Datenkompression*, 2002

[7] S. Zahir and M. Naqvi, *"A New Rectangular Partitioning Based Lossless Binary Image Compression Scheme," IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 267-271, Saskatoon, May, 2005.

[8] A.A. Markov, *"Extension of the limit theorems of probability theory to a sum of variables connected in a chain", reprinted in Appendix B of: R. Howard. Dynamic Probabilistic Systems*, vol. 1: Markov Chains. John Wiley and Sons, 1971.

[9] ISO/IEC JTC1/SC29/WG1 N1359, *"Coding of Still Pictures, Final Committee Draft,"* JBIG Committee, July 16, 1999.

[10] D. Salomon, *"Data Compression The Complete Reference 3rd Edition,"* Springer-Verlag New York, 2004.

[11] P. Kopylov and P. Franti, *"Compression of Map Images by Multilayer Context Tree Modeling," IEEE Trans on Image Processing*, vol. 14, no 1, 2005