Image Compression Based on Reed-Muller Transform

Kamran Iravani, Marek Perkowski*

VLSI Technology, Inc., San Jose, CA, USA
* Portland State University, Dept. Electrical Engineering, Portland, OR, USA

Outline

- Important Points on Image Compression Techniques.
- Reed-Muller Transform.
- Fixed Polarity Reed-Muller Transform.
- FPRM Image Compression Method.
- Experimental Results.
- A Critique of the Reddy and Pai's Paper on Reed-Muller Image Compression.
- Conclusion.

Important Points on Image Compression Techniques

- Compression factor.
- Quality of the reconstructed image.
- Speed of the process.
- Hardware realization of the system.

- In lossless image compression the reconstructed image is exactly the same as the original one but the compression factor is not very high. In lossy Image compression the quality of the reconstructed image is poorer than the original one but the compression factor is high. Therefore there is a trade-off between compression factor and quality of the image.
- Because of simplicity of the hardware and speed of the process, Image compression using Reed-Muller transform, Proposed by Reddy and Pai, was assumed to be a good candidate as a fast compression technique.

Reed-Muller Transform

- Any switching function of n variables can be defined by 2^n coefficients in a sum of product (SOP) form as in Eq. 1:

$$f(x_0, x_1, \ldots, x_{n-1}) = d_0\overline{x_{n-1}}\,\overline{x_{n-2}}\ldots\overline{x_0} + d_0\overline{x_{n-1}}\,\overline{x_{n-2}}\ldots\overline{x_1}x_0 + \ldots + d_{2^n-1}x_{n-1}x_{n-2}\ldots x_0 \qquad (1)$$

  where $(d_0, d_1, \ldots, d_{2^n-1})$ are the coefficients of the products, which can be represented in vector form ***D*** called truth vector.

- The function can also be represented by the positive polarity Reed-Muller (PPRM) canonical form over Galois field (2) as in Eq. (2):

Reed-Muller Transform (cont.)

$$f(x_0, x_1, \ldots, x_{n-1}) = a_0 \oplus a_1 x_0 \oplus a_2 x_0 x_1 \oplus \ldots \oplus a_{2^n - 1} x_0 x_1 \ldots x_{n-1} \qquad (2)$$

Where $\oplus$ denotes modulo-2 addition and $(a_0, a_1, \ldots, a_{2^n - 1})$ are the coefficients of the expansion, which can be represented in vector form ***A***, called the function vector.

- The two representations are related to each other by a transform matrix ***T*** as in Eq. 3:

$$A = TD \qquad (3)$$

Where ***T*** can be written in a recursive form as in Eq. (4):

Reed-Muller Transform (cont.)

$$T = T_n = \begin{bmatrix} T_{n-1} & 0 \\ T_{n-1} & T_{n-1} \end{bmatrix} \quad \text{for n>=1} \qquad (4)$$

and $T_0 = 1$

- An important property of matrix ***T*** is that ***T*** is the inverse of itself, i.e. $\boldsymbol{T}=\boldsymbol{T}^{-1}$. Therefore Eq. (5) is also true:

$$D = TA \qquad (5)$$

Fixed Polarity Reed-Muller Form

- In the PPRM expansion form all the variables are in the positive form, while any variable can be substituted with its negation , and still retain the canonical form. In the case that each variable is restricted to retain the same polarity in all terms, i.e. either positive or negative but not both, the canonical form is called the Fixed Polarity Reed-Muller (FPRM) form.
- For a function with n variables the number of possible polarities is 2^n, so there are 2^n possible FPRM forms.
- By finding appropriate polarities the least number of terms in a specific function can be obtained.

FPRM Image Compression Method

- Every image consists of a set of pixels. In a grey-level picture, each pixel is usually represented by one byte. Therefore an image can be modeled by a matrix of pixel values, where each pixel value can vary in the range of 0 to 255, i.e. 256 levels of grey. In other words each pixel value consists of 8 bits, and each bit can be either 1 or 0.
- The picture matrix can be converted to 8 matrices from the most significant bit plane to the least significant bit plane, and each element of these matrices is either 1 or 0.

FPRM Image Compression Method (cont.)

- The algorithm to compress the image is as follows:

for(i=most significant bit plane to least significant bit plane)
{
for (each plane)
{
*(a) fetch a block of size (n*x*n)*
(b) find the best polarity of the variables
(c) compute the Reed-Muller transform of the block
(d) employ run-length coding on the resultant tranform domain bit plane
};
};

FPRM Image Compression Method (cont.)

- In this method each block of size nxn is considered as a Karnaugh map, so the number of 1s is each block corresponds to the number of true minterms. After finding its FPRM form with the minimum number of terms, the new map usually contains fewer number of 1s than the original map.

- For run-length coding three methods are being used: one dimentional Run Length Coding (RLC), Relative Adress Coding (RAC) and Relative Element Address Designate (READ).

FPRM Image Compression Method (cont.)

- RAC and READ give almost the same result and both of them are better than RLC.
- Since READ is more complicated than RAC but has almost the same effect on compression, RAC has been considered to be the best method of coding for this purpose.
- To determine the relative address distance codes in RAC, Huffman coding was used which gives the best compression factor.

Experimental Results

- Table 1 shows the compression factors and the average numbers of transition elements of the four MSB planes. These data have been calculated based on 10 different pictures of natural scenes.

Table 1:

Bit Plane	MSB	2nd MSB	3rd MSB	4th MSB
Compression Factor	1.9	1.2	1	1
Average number of transition elements for images of size 256x256	11000	19000	25500	29000

Comments on the Experimental Results

- Size of the blocks has been taken to be 16x16 in order to be comparable with Reddy and Pai's simulation results.
- The results show that the number of transition elements is high which causes the compression factor to be low.
- Even if loss of information is allowed in the four least significant bit planes, the compression factor is less than 2 which is low compared to commercially used compression techniques such as JPEG.

A Critique of the Reddy and Pai's Paper on Reed-Muller Image Compression

- The algorithm is as follows:

for(i=most significant bit plane to least significant bit plane)
{
for (each plane)
{
(a) fetch a block of size (nxn)
(b) compute the Reed-Muller transform of the block
(c) employ run-length coding on the resultant tranform domain bit plane
};
};

- Problems with step (b):

  * A good compression factor cannot be achieved by just computing the Reed-Muller transform of the block. Even for MSB planes it is less than 1.6.

  * If a Permutation is applied, the compression factor might get better in some cases. However, finding the best permuted input sequence can be done only exhaustively which cannot be realized by any hardware, because it requires too many cases for checking. Therefore it is not a practical method.

- Problems with step (c):

  * Table 2 shows the codes used by Reddy and Pai for relative distances:

Table 2:

relative distance	positive pistance	negative distance
1	00	10
2	01	11
3	0000	1010
4	0001	1011
5	0100	1110
6	0101	1111
7	000000	101010

* These codes cannot be used for relative distances in RAC because some of these codes are first part of some other codes. Therefore the codes cannot be detected.
* They have said that depending on the probability of the occurrence, the RAC distances (0, +1, -1) are given special code words. The obvious thing is that all the codes must be predefined. It would be impractical to first find the probability of the occurrence of the runlengths for every single matrix of an image, and then according to that find the codewords for them.
* Since the size of the blocks is 16x16, when the RAC method is used, there is a need to define the codes for 16 positive reletive distances, and not just for 8, although for negative relative distances 8 codes are needed.

Conclusion

- An image compression method based on Fixed polarity Reed-Muller transform was created and analyzed.
- It was shown that FPRM tansform cannot be a good candidate for compression of still images, although the method is very fast.
- It was demonstrated that the paper published by Reddy and Pai on Reed-Muller image compression contains several errors which make their results unacceptable.