Lossless Compression With Context And Average Encoding And Decoding And Error Modelling In Video Coding

Abstract:

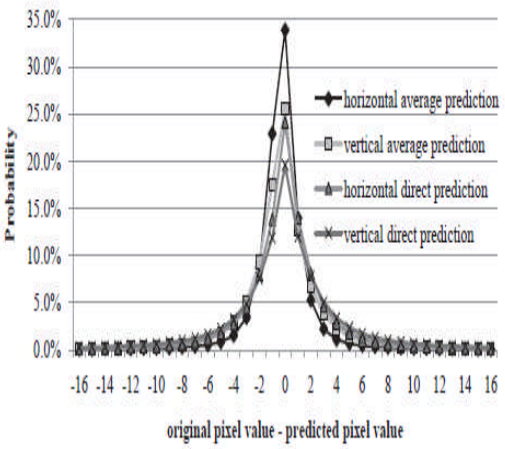
Image compression is now essential for applications such as transmission and storage in data bases. In this paper we review and discuss about the image compression, need of compression, its principles, and classes of compression and various algorithm of image compression. With the increase of image resolution in video application, the memory bandwidth is a critical problem in video coding. An embedded compression algorithm is a technique that can compress the frame data when stored in memory. It is possible to reduce memory requirements. In this paper, we propose a lossless embedded compression algorithm in addition to context-based error compensation and average encoding and decoding to reduce the memory bandwidth requirement. Experimental results have shown at least 50% memory bandwidth reduction on average and the data reduction ratio of the proposed algorithm is up to 5% higher than previously proposed lossless embedded compression algorithm.

Index terms:

Video coding, significant bit truncation, average prediction, entropy coding, lossy and lossless compression.

INTRODUCTION:

Many embedded compression (EC) algorithms have been proposed which reduce the amount of data between the off-chip memory and video coding system by up to 50%. However, algorithms proposed in are lossy compression algorithms where the quality loss is inevitable. Moreover, the Number of clock cycles in required to decompress a 16×16 macro block does not meet the requirement of real-time high- definition (HD) video coding. The probability distribution of difference of pixel values between the predicted pixel and the original pixel. On the other hand, there are some hardware-friendly lossless compression algorithms, but they require too many clock cycles to handle the HD video source. In this paper, we propose a lossless embedded compression algorithm based on a spatial prediction in a given block and the so-called truncated bit packing technique. Since there is a strong spatial correlation between neighboring pixels, the current pixel can be well predicted by using an average or a direct prediction from neighboring pixels. The resultant small errors in the prediction are compressed by the truncated bit packing technique which allows processing multiple symbols in a clock cycle.



- *MadhavanS is currently pursuing Bachelors degree program in electronics and communication engineering in Anna university,India,PH-+918056350477, E-mail:madragon16@gmail.com*
- *CManirathinam is currently pursuing Bachelors degree program in electronics and communication engineering in Anna university,India,PH-+919750342694*

2 PROCEDURE FOR PAPER SUBMISSION

2.1 Review Stage

In the review stage the base paper was analyzed and the

code was executed on the matlab software. The result was a compression ratio of approximately 5%. The deviation was analyzed thoroughly.

2.2 Final Stage

In the final stage we had decided the deviation and the adaptive prediction was introduced into the system instead of average prediction. This gave the compression ratio of 8%.

2.3 Figures

3 IMAGE COMPRESSION:

Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements. Compression is achieved by the removal of one or more of the three basic data redundancies:

1. Coding Redundancy
2. Interpixel Redundancy
3. Psycho visual Redundancy

Coding redundancy is present when less than optimal code words are used.

Interpixel redundancy results from correlations between the pixels of an image.

Psycho visual redundancy is due to data that is ignored by the human visual system.

Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. An inverse process called decompression (decoding) is applied to the compressed data to get there constructed image. The objective of compression is to reduce the number of bits as much as possible, while keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible. Image compression systems are composed of two distinct structural blocks: an encoder and a decoder.

Image f(x, y) is fed into the encoder, which creates a set of symbols form the input data and uses them to represent the image. If we let n1 and n2 denote the number of information carrying units(usually bits) in the original and encoded images respectively, the compression that is achieved can be quantified numerically via the compression ratio,CR = n1 /n2 .As shown in the figure, the encoder is responsible for reducing the coding,interpixel and psycho visual redundancies of input image. In first stage, the mapper transforms the input image into a format designed to reduce interpixel redundancies. The second stage, quantizer block reduces the accuracy of mapper's output in accordance with predefined criterion. In third and final stage, a symbol decoder creates a code for quantizer output and maps the output in accordance with the code. These blocks perform, in reverse order, the inverse operations of the encoder's symbol coder and mapper block. As quantization is irreversible, an inverse quantization is not included.

4 PRINCIPLES BEHIND COMPRESSION:

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS). In general, three types of redundancies can be identified.

A. Coding Redundancy:

A code is a system of symbols (letters, numbers, bits, and the like) used to represent a body of information or set of events. Each piece of information or events is assigned a sequence of code symbols, called a code word. The number of symbols in each code word is its length. The 8-bit codes that are used to represent the intensities in

the most 2-D intensity arrays contain more bits than are needed to represent the intensities.

B. Spatial Redundancy and Temporal Redundancy

Because the pixels of most 2-D intensity arrays are correlated spatially, information is unnecessarily replicated in the representations of the correlated pixels. In video sequence, temporally correlated pixels also duplicate information.

C. Irrelevant Information

Most 2-D intensity arrays contain information that is ignored by the human visual system and extraneous to the intended use of the image. It is redundant in the sense that it is not used. Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible.

NEED FOR COMPRESSION:

The qualitative transition from simple textto full-motion video data and the disk space transmission bandwidth, and transmission time needed to store and transmit such uncompressed data .Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required. The prefix kilo- denotes a factor of 1000 rather than 1024.

At the present state of technology, the only solution is to compress multimedia data before its storage and transmission, and decompress it at the receiver for play back. For example, with a compression ratio of 32:1, the space, bandwidth, and transmission time requirements can be reduced by a factor of32, with acceptable quality.

5 IMAGE COMPRESSION TECHNIQUES:

The image compression techniques are broadly classified into two categories depending whether or not an exact replica of the original image could be reconstructed using the compressed image .

These are:

1. Lossless technique
2. Lossy technique

Lossless compression technique:

In lossless compression techniques, the original image can be perfectly recovered form the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image).It is also known as entropy coding since it use statistics/decomposition techniques to eliminate/minimize redundancy. Lossless compression is used only for a few applications with stringent requirements such as medical imaging. Following techniques are included in lossless compression:

1. Run length encoding
2. Huffman encoding
3. LZW coding
4. Area coding

6 LOSSLESS COMPRESSION TECHNIQUES

1. Run Length Encoding

This is a very simple compression method used for sequential data. It is very useful in case of repetitive data. This technique replaces sequences of identical symbols (pixels), called runs by shorter symbols. The run length code for a gray scale image is represented by a sequence {Vi, Ri} where Vi is the intensity of pixel and Ri refers to the number of consecutive pixels with the intensity Vi as shown in the figure. If both Vi and Ri are represented by one byte, this span of 12 pixels is coded using eight bytes yielding a compression ratio of 1: 5

2. Huffman Encoding

This is a general technique for coding symbols based on their statistical occurrence frequencies (probabilities). The pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the (binary) code of any symbol is not the prefix of the code of any other symbol. Most image coding standards use lossy techniques in the earlier stages of compression and use Huffman coding as the final step.

3. LZW Coding

LZW (Lempel- Ziv – Welch) is a dictionary based coding. Dictionary based coding can be static or dynamic. In tactic dictionary coding, dictionary is fixed during the encoding and decoding processes. In dynamic dictionary coding, the dictionary is updated on fly. LZW is widely used in computer industry and is implemented as compress command on UNIX.

4. Area Coding

Area coding is an enhanced form of run length coding, reflecting the two dimensional character of images. This is a significant advance over the other lossless methods. For coding an image it does not make too much sense to interpret it as a sequential stream, as it is in fact an array of sequences

l Data Prediction /Transformation /Decomposition Entropy (Lossless)Coding QuantizationCompresseddatabuilding up a two dimensional object. The algorithms for area coding try to find rectangular regions with the same characteristics. These regions are coded in descriptive forms an element with two points and a certain structure. This type of coding can be highly effective but it bears the problem of a nonlinear method, which cannot be implemented in hardware.

7 ENTROPY:

Three Entropy coding techniques:

- Huffman coding
- Arithmetic coding
- Lempel-Ziv coding

Entropy (in our context) - smallest number of bits needed, on the average, to represent a symbol (the average on all the symbols code lengths). Entropy is a lower bound on the average number of bits needed to represent the symbols (the compression limit).

- Entropy coding methods:
- Aspire to achieve the entropy for a given alphabet, BPS_Entropy.
- A code achieving the entropy limit is optimal. Each symbol is assigned a variable-length code, depending on its frequency. The higher its frequency, the shorter the codeword.
- Number of bits for each codeword is an integral number
- A prefix code
- A variable-length code

Comparison

	Huffman	Arithmetic	Lempel-Ziv
Probabilities	Known in advance	Known in advance	Not known in advance
Alphabet	Known in advance	Known in advance	Not known in advance
Data loss	None	None	None
Symbols dependency	Not used	Not used	Used – better compression
Preprocessing	Tree building – O(n log n)	None	First pass on data (can be eliminated)
Entropy	If probabilities are negative powers of 2	Very close	Best results when alphabet not known
Codewords	One codeword for each symbol	One codeword for all data	Codewords for set of alphabet
Intuition	Intuitive	Not intuitive	Not intuitive

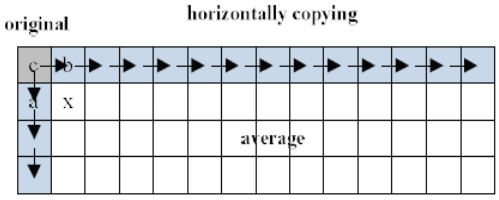
PROPOSED ALGORITHM

This paper is organized as follows: first, we describe proposed average prediction, followed by context-based error compensation. Finally, SBT-based entropy coding [2] that is adopted in the proposed algorithm is presented.

A. Average prediction

The average prediction scheme used here is shown in Fig.

1. The current pixel value x is differential-coded using the average value of the upper and left pixel values of the current one, b and a, respectively. Where the pixels placed on the left or top of the random access unit are predicted by copying horizontally or vertically and the others are predicted using average value of the upper and left pixels of the current pixel.



Formula used:

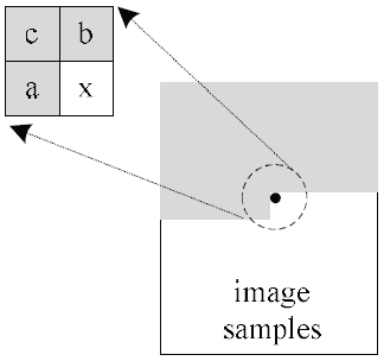
$$\hat{x}_{MED} \cong \begin{cases} \min(a,b), & \text{if } c \geq \max(a,b) \\ \max(a,b), & \text{else if } c \leq \min(a,b) \\ a+b-c, & \text{otherwise} \end{cases}.$$

$$\hat{x}_{AVG} \cong \frac{a+b+1}{2}$$

Context-based Prediction Error Compensation

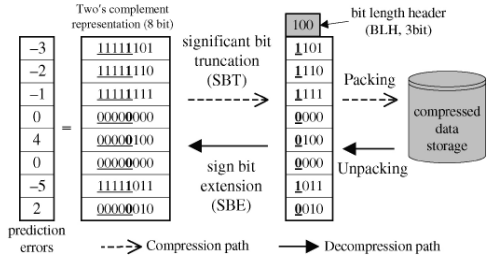
Fig. 2(a) presents the prediction error distribution. If it is the worst case of compression performance, the prediction error distribution is wider than the best case of that. The prediction error distribution is more concentrated to zero using context-based error compensation. For low complexity and storage efficiency, we quantize context conditions into 9 steps using T1, T2, and T3 threshold level. (T1 = 3, T2 = 7, T3 = 21). Thus, the quantization regions are represented as, {0}, {1, 2}, {3, 4, 5, 6}, {7, 8, …, 20}, ⊚{21, …, 255} and indexed [-4, 4]. A total of (2T + 1)3 = 729 contexts (T=4). By merging contexts of "opposite signs" the total number of contexts becomes ((2T + 1)3 +1)/ 2 = 365 context conditions [3]. We call this condition CTX999.

Meanwhile, in typical error compensation using the proposed context-based model, coding errors are accumulated according to the contexts that correspond to gradient values between neighborhood pixels within a frame. In EC algorithms, the context-based model utilization is largely constrained because each small coding unit is independently dealt with, which causes lack of statistical data for context accumulation.



Entropy Coding

The compensated prediction error using contexts is entropy-coded using Significant Bit Truncation (SBT) method proposed in [2]. It is worth noting that the average difference between the theoretical upper bound of the SBT method and entropy is proven as only 0.74 bit-per-pixel. As well as its simplicity, a superior coding performance is appropriate for EC algorithms.



Block prediction

The main purpose of block prediction is to remove the spatial redundancies. This section will consider prediction modes used for different block types. The 2x32 blocks are divided into smaller sub-blocks for sequential coding. 1x4 sub-block prediction is applied for flat and detail blocks while 2x4 subblock prediction is applied for random blocks. Pixelbased prediction and coding were implemented to achieve more accurate prediction for lossless compression. But it required a large number of bits to indicate the prediction mode for every pixel. The proposed algorithm uses block-based mode indication to save the mode indicating bits. Based on the experiments, 1x4 sub-block is chosen for flat and detail areas. This option permits small prediction error in all sub-blocks and requires a modest number of bits for indicating modes. For random areas, the prediction error is very high. In these areas, bigger sub-block of 2x4 pixels helps saving more bits needed for indicating the type of quantizer which is used later. The predicted pixels are subtracted from the original pixels to form the residual errors.

di = Xi-Xip

These residual errors are then quantized to diq in the encoding phase. For decoding, the quantized residual errors are added back to the predicted pixels to formulate the reconstructed pixels

XiR = XiP+di

For random blocks, high quantization error is still undistinguishable and coarse quantizer can be

used. This quantizer permits larger residual signal so the block size can be extended to 2x4 pixels.

D. Adaptive Encoding

After being sub-block predicted, the residual error will be quantized and encoded to form the bit-stream. For flat areas, prediction error is usually small. This error should be quantized with very fine quantization level to avoid large error, which is easily observed in these areas. For detail and random areas, prediction error is usually large. But these areas permit larger imperceptible quantization error than flat areas. That means larger quantization levels can be applied in these areas while the distortion is still visually lossless. The quantizer in this section is thus non-uniform with very fine quantization levels in the low value areas and with coarser quantization levels in the higher value of residual error. The quantized residual signal is obtained by approximating the residual error by its centroid value. Quantizers for flat, detail and random blocks have different quantization step sizes Δxi . This makes the non-uniform quantizers adapt to the block types. The maximum quantization interval is 63. But not always all quantization intervals are occupied. If the all residual error of the block is small, only some quantization intervals are needed. Using all intervals in these cases will waste the number of bits to encode the quantized value. Only a sufficient number of intervals should be implemented. This number of used intervals N is determined based on the maximum values max$d$ of all residual values $|di|$ in the sub-block. Each Mode is indicated by a Huffman code based on its occurrence probability. The quantized residual error of all pixels in the block is then encoded using a fixed-length coding scheme. If Mode = M, then M bits are needed to code each quantized residual signal. A similar scheme is used for encoding the residual error of 2x4 sub-block. Fig. 3.9 shows an example of the bit structure with Mode = 5 for 1x4 flat blocks. The encoded bits for residual errors are shown in Fig. 3.10. The residual error for each pixel in this example requires five bits to represent the quantized error and 4 bits to represent the sub block mode.

E.Adaptive Decoding

For the decoding, the reverse process will be implemented. The first 30 bits are read from the bit stream to determine the framesize. Then two more bits are extracted to find the block types. Based on this block types, the decoder will use the corresponding mode code for the sub-block as well as the quantizer for decoding. Next, the Huffman code for the subblock will be read to determine the mode value. If Mode = M, then the next M bits are extracted and use the fixed-length code to indicate the quantized residual error diq. This value is added back to the predicted pixel XiP to get the reconstruct pixel XiR. The next M bits are extracted for the next pixel decoding until all the pixels in the sub-block are reconstructed. The decoder continues decoding the next subblock until all sub-blocks in the 2x32 block are decoded, then it repeats the process until all blocks in the frame are reconstructed.

8 CONCLUSION

In this paper, we proposed a lossless embedded compression algorithm for video application. The proposed algorithm occurs in five steps: average prediction, context based error compensation, block prediction,average encoding and average decoding. The average prediction gives more low complexity compared to other prediction method. Through the context-based error compensation, more than 5% of data is compressed with no quality degradation and bit-rate increment. And, we can be implemented small memory size increase to store context conditions through temporal contexts or largely reduced quantized regions of context conditions. The compression performance gain of the proposed algorithm can enhance the video coding efficiency by enlarging the search range of motion estimation [4] or by reducing additional memory bandwidth for various video applications.

REFERENCES:

[1] Lossless Embedded Compression Algorithm With Context-Based Error Compensation For Video Application Hyerim Jeong, Jaehyun Kim, Kyohyuk Lee, Kiwon Yoo, And Jaemoon Kim, Member, IEEE.

[2] Visually Lossless Compression For Color Images With Low Memory Requirement Using Lossless Quantization Mary Jansi Rani. Y, Pon. L.T. Thai, John Peter. K.

[3] Comparison Of Compression Algorithms For High Definition And Super High Definition Video Signals Hrvoje Balaško Audio Video Consulting Ltd., Karlovačka 36b, 10020 Zagreb, Croatia.

[4] Multi-Mode Embedded Compression Codec Engine For Power-Aware Video Coding System Chih-Chi Cheng , Po-Chih Tseng, Chao-Tsung Huang, And Liang-Gee Chen DSP/IC Design Lab, Graduate Institute Of Electronics Engineering And Department Of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

[5] Gray-Level-Embedded Lossless Image Compression Mehmet Utku Celika, Gaurav Sharmab,*, A. Murat Tekalpa,C Adepartment Of Electrical And Computer Engineering, University Of Rochester, Rochester, NY 14627-0126, USA.

[6] A Dynamic Search Range Algorithm For Stabilized Reduction Of Memory Traffic In Video Encoder Jongpil Jung, Jaemoon Kim, *Student Member, IEEE,* And Chong-Min Kyung, *Fellow, IEEE.*

[7] The LOCO-I Lossless Image Compression Algorithm: Principles And Standardization Into JPEG-LS Marcelo J. Weinberger And Gadiel Seroussi Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA Guillermo Sapiro∗ Department Of Electrical And Computer Engineering University Of Minnesota, Minneapolis, MN 55455, USA.

[8] A Lossless Embedded Compression Algorithm For High Definition Video Coding *Jaemoon Kim, Jungsoo Kim And Chong-Min Kyung*