Lossless Compression Method for Medical Image Sequences Using Super-Spatial Structure Prediction and Inter-frame Coding

Mudassar Raza, Ahmed Adnan, Muhammad Sharif*, Syed Waqas Haider

Department of Computer Sciences,
COMSATS Institute of Information Technology Wah Cantt., 47040, Pakistan
* muhammadsharifmalik@yahoo.com

ABSTRACT
Space research organizations, hospitals and military air surveillance activities, among others, produce a huge amount of data in the form of images hence a large storage space is required to record this information. In hospitals, data produced during medical examination is in the form of a sequence of images and are very much correlated; because these images have great importance, some kind of lossless image compression technique is needed. Moreover, these images are often required to be transmitted over the network. Since the availability of storage and bandwidth is limited, a compression technique is required to reduce the number of bits to store these images and take less time to transmit them over the network. For this purpose, there are many state-of the-art lossless image compression algorithms like CALIC, LOCO-I, JPEG-LS, JPEG20000; Nevertheless, these compression algorithms take only a single file to compress and cannot exploit the correlation among the sequence frames of MRI or CE images. To exploit the correlation, a new algorithm is proposed in this paper. The primary goals of the proposed compression method are to minimize the memory resource during storage of compressed data as well as minimize the bandwidth requirement during transmission of compressed data. For achieving these goals, the proposed compression method combines the single image compression technique called super spatial structure prediction with inter-frame coding to acquire grater compression ratio. An efficient compression method requires elimination of redundancy of data during compression; therefore, for elimination of redundancy of data, initially, the super spatial structure prediction algorithm is applied with the fast block matching approach and later Huffman coding is applied for reducing the number of bits required for transmitting and storing single pixel value. Also, to speed up the block-matching process during motion estimation, the proposed method compares those blocks that have identical sum and leave the others; therefore, the time taken by the block-matching process is reduced by minimizing the unnecessary overhead during the block-matching process. Thus, in the proposed fast lossless compression method for medical image sequences, the two-stage redundant data elimination process ultimately reduces the memory resource required for storing and transmission. The method is tested on the sequences of MRI and CE images and produces an improved compression rate.

Keywords: Compression, JPEG-LS, blocks, CALIC, MRI, CE, interframe coding, image sequence.

1. Introduction

Since computers play a vital role in solving the complex issues of our daily life, it has applications even in medical diagnostics. This diagnostics includes magnetic resonance imaging (MRI), ultrasound, computed tomography (CT), capsule endoscopy (CE), etc. These diagnostic machines produce their results in the form of sequences of digital images. These sequences require enormous disk space for storage and take more time for transmission over the network.

The solution to this problem can be image compression. Since these images are of high importance and any loss of information in them during the compression process could have greater consequences, we need to use the lossless image compression algorithm. There are many state-of-the-art algorithms like CALIC, LOCO-I, JPEG-LS, JPEG20000 to perform image compression in a way that no information gets lost after decompression of the images.

JPEG-LS has exceptional coding and computational efficiency and has many lossless image compression algorithms including JPEG 2000[1] [2]. But the super-spatial structure prediction algorithm proposed in [3] has outperformed the JPEG-LS algorithm. This algorithm divides the image into two regions, i.e., structure regions (SRs) and non-structure regions (NSRs). The structure regions are compressed using the super-spatial prediction technique, while the non-structure regions are compressed using CALIC.

The idea of super-spatial structure prediction is taken from video coding. A single image consists of different objects and each object consits of many structures. These structures are repeated in many places in a single image. These structure elements include edges, pattern and textures.

Despite the fact that the super-spatial structure prediction algorithm has outperformed the state-of-the-art algorithms, the drawback is that it deals only with single images and does not perform correlation among the frames in a sequence. Since there is too much correlation among medical sequences, we can achieve a higher compression ratio using interframe coding.

The idea of a compression sequence was first adopted in [5] for lossless image compression and used in [6], [7] and [8] for lossless video compression. But the compression ratio was significantly low, i.e., 2.5, which was even less than the compression ratio achieved by some state-of-the-art algorithms for single image compression.

In [4], the idea of interframe coding was again adopted; nevertheless, this time they used JPEG-LS, the state-of-the-art algorithm for single image compression with interframe coding, through this, a higher compression ratio was achieved. However, this ratio can still be further improved by using super-spatial structure prediction technique used for single image compression, hence replacing the JPEG-LS used for the image sequence.

Since correlation among medical image sequences is very high, it can be exploited by using single-image coding with interframe coding. In this paper, a new technique is proposed for image compression using interframe coding with super-spatial structure prediction.

2. Sequence Compression With Super-Spatial Structure Prediction

Since most of the compression algorithms use a single file compression technique, they do not exploit the interframe correlation of images. The aim of the proposed algorithm is to improve the compression rate by using interframe correlation. Figure 1 illustrates the complete encoding technique for the proposed algorithm.

In this algorithm, the first image is always compressed using the super-spatial structure prediction algorithm *SSP()*. The output file of the super-spatial structure prediction algorithm is further compressed using Huffman coding scheme *HMNC()*. Once the first image is compressed, the second image becomes the current image and the first image becomes the reference image for the second image.

Now motion estimation and motion compensation is done for these two images. Both methods are established methods for video compression. After motion estimation and motion compensation, the difference of images is processed for compression. This difference is also compressed using super-spatial structure prediction algorithm. After that, Huffman coding is applied to remove those remaining redundancies from the image. Once the compression of the second frame is done, it becomes the reference frame for the third frame and this process goes on until the last frame.

2.1 Usability of Motion Estimation and Motion compensation

Motion estimation means the estimation of the displacement (or velocity) of the predicted structural image from one frame to another in a time sequence of 2-D images. The structure is the group of pixels having direction which can be observed on the 2-D axis and magnitude in terms of pixel intensity value. Expressing the usability of motion estimation in medical image sequences compression, consider Figure 2 where S(x,y) is a group of pixels which formed a predicted structure, ME( ), which can be defined as in Equation 1.

$$ME() = (S^{\wedge}{}_{X,Y})^{T} \qquad (1)$$

where $S^{\wedge}{}_{X,Y}$ is the derivative of structure $S_{X,Y}$ with respect to time T, and T is equal to the initial time change in time, i.e., Δt. In the motion estimation phase *ME(),* just like in video coding, the current frame is compared with the reference frame, i.e., previous frame for any change. If no change occurs, then the encoder marks it as duplicate, otherwise it records changes among the frame [4]. Motion compensation [9] MC( ) uses motion estimation outputs which are displacements of the predicted structure at different *time*; hence, we create a reference frame after every fourth frame because much of the information is the same in the consecutive frames. On decoding time, we create frames from the reference frame. This increases the compression ratio and enables us to save the frame in less space [4]. Although during execution of compression the processor's resource use is directly proportional to the size of the predicted structure, i.e., number of pixels occupied by the predicted structure among sequence of frames. After motion estimation and compensation, the image difference *ID()* is computed by calculating each pixel's intensity level of reference block and given block, which can also come in negative values. Therefore, to handle negative values which cannot be accommodated in the 8-bit format, the 9 bits are used to encode the single pixel intensity value.

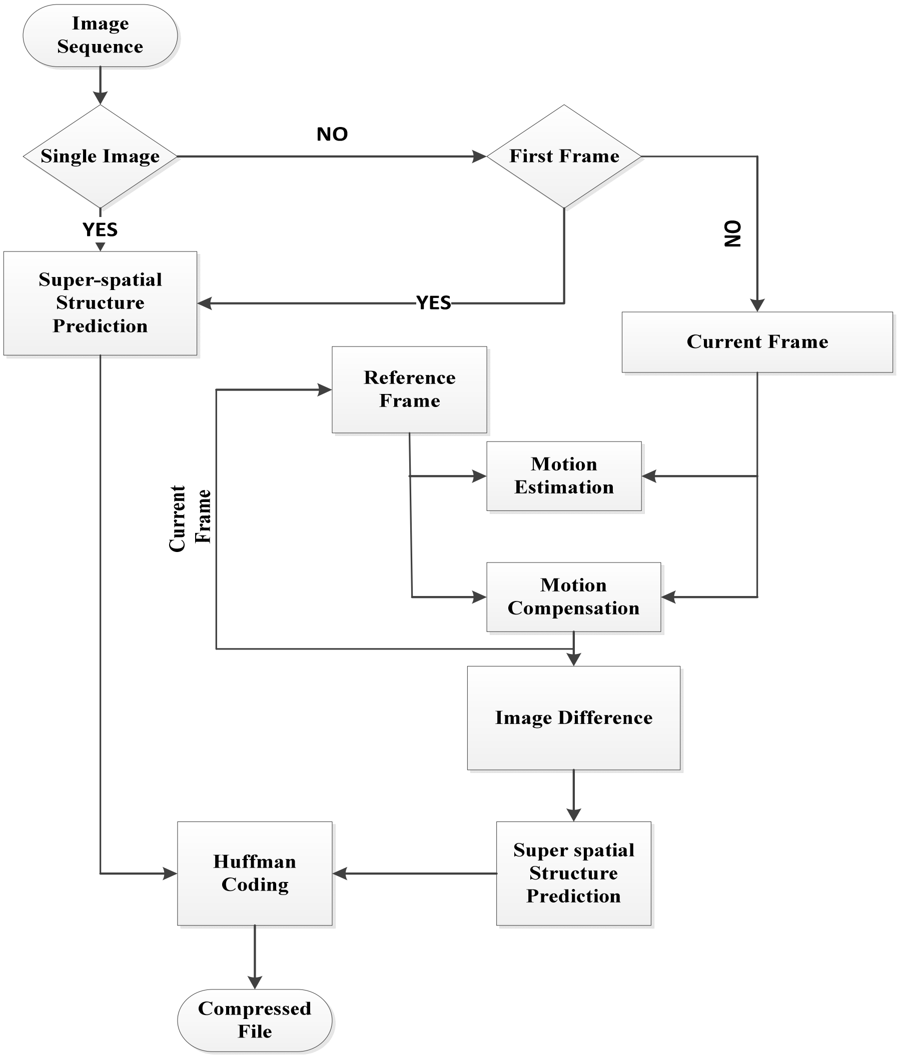

Figure 1. Flow diagram compression phase.

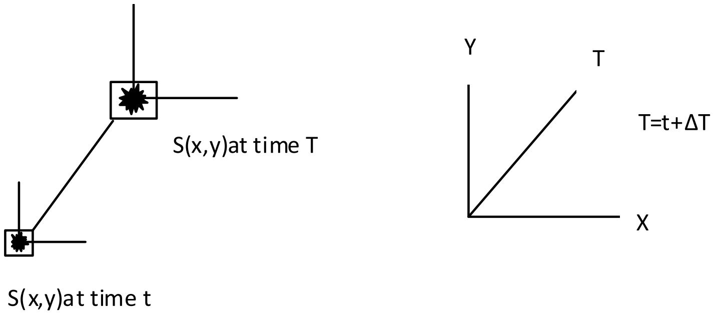

Figure 2. Predicted structure motion estimation.

The purpose of applying Huffman coding *HMNC()* is to further eliminate the redundancies. This algorithm is applied to data produced after super-spatial structure prediction. The data consists of numeric values which represent color values for each pixel in an image; hence, by using Huffman coding the redundant numeric values are further removed. The whole compression phase is expressed as

$$compression = \left| \begin{array}{ll} SSP() + HMNC() & I(X_1, Y_1) \\ & if \\ ME() + MC() + ID() + SSP() + HMNC() & \neq I(X_1, Y_1) \end{array} \right| \tag{2}$$

From expression 2, it can be observed that when the input frame is the first frame, i.e., $I(x_1, y_1)$, then straightforward apply SSP() and later HMNC(). Similarly, if the input is not the first frame, i.e., $\neq I(x_1, y_1)$, then apply ME( ), MC( ), ID( ), SSP( ) and HMNC( ).

2.2 Block Matching

Since the block-matching process during the functions ME( ) and MC( ) (which are taken from [4] and represented in Equation 1) takes too much time, we need a better block-matching technique to speed up the compression process. In order to speed up the block-matching process, we only compare those blocks that have the same sum and leave the others; therefore, the time taken for the block matching process is reduced. For this purpose, we used a binary tree. The block-matching process is expressed mathematically as in Equation 3:

$$IND() = \left| \begin{array}{l} if\ BS(X_C, Y_C) \neq BS(X_{C-1}, Y_{C-1}) \\ or \\ if\ BS(X_C, Y_C) \neq \sum_{i-1}^{n} BS(X_i, Y_i) \end{array} \right| \tag{3}$$

From expression 3, the function *IND() Insert Node into Binary tree* is defined as “the nodes of the binary tree hold sum of the blocks and their index number in the array”. In this technique, the first block is placed in the first index of block array and its sum and index is placed in the root node of the binary tree, i.e., $BS(X_{C-1}, Y_{C-1})$, which represents the block sum of the reference block.

Once the first block is placed in the block array and in the binary tree, the sum of the second block is calculated, i.e., $BS(X_C, Y_C)$, which is actually the current block with respect to previous block ($BS(X_{C-1}, Y_{C-1})$). Now the current block sum is compared with the sum of the first block. If the sum of these blocks is identical and if the second block is the same and already exists in the block array, then no new block is inserted and no node is created in the binary tree. Figure 3 illustrates the idea of the block matching process.

If the sum of the blocks does match (see Equation 3) but the blocks are not identical, then a new block is inserted into the block array and a new node is also created in the binary tree. If the sum of the current block *BS()* does not match the whole binary tree, a new block is inserted into the block array and a new node is also created in the binary tree. This process continues until the last block. Since in the block-matching process we only compare those blocks that have the same sum and leave the others, the time taken for the block-matching process is reduced by minimizing the unnecessary overhead.

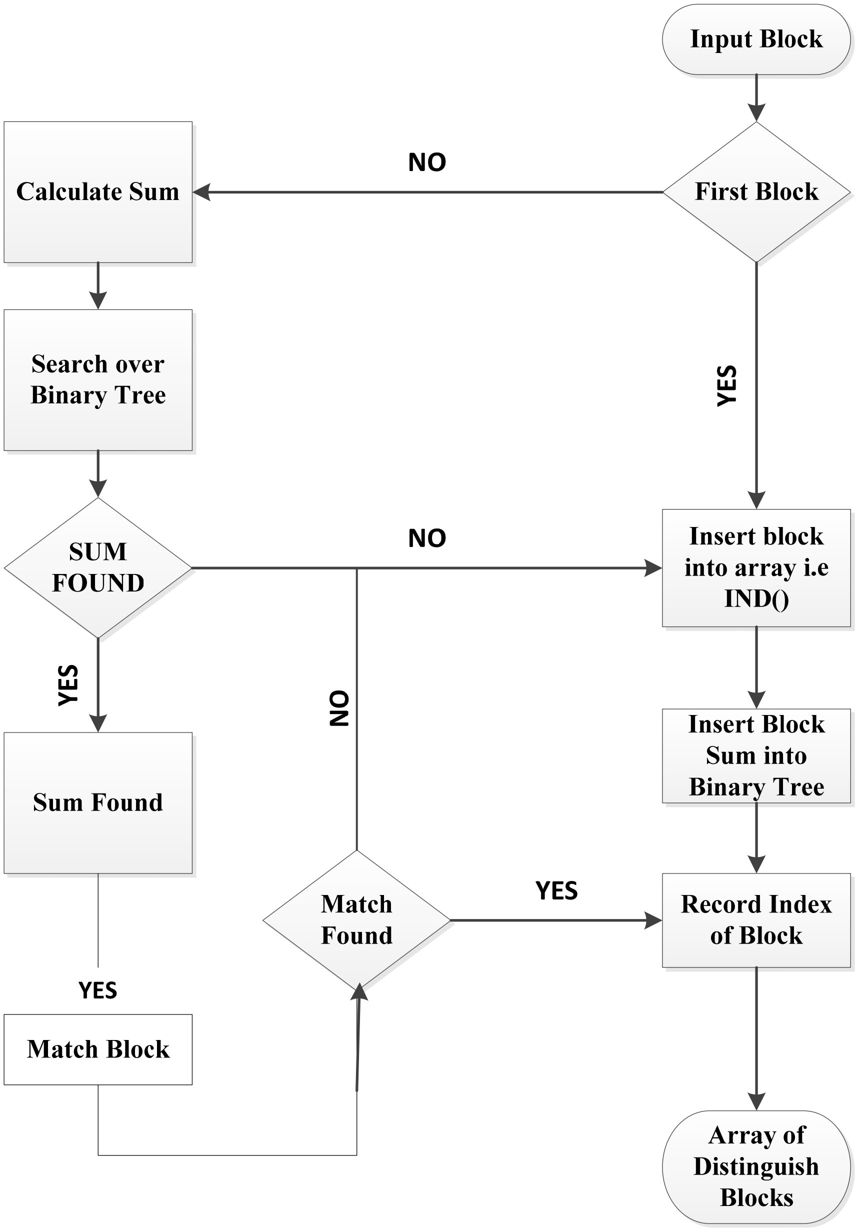

Figure 3. Fast block-matching process.

3. Example Of Compression Phase

Given below are the initial frames of MRI sequence which are taken from the MR-TIP database (see Figure 4) for the evaluation of the proposed algorithm. In this algorithm, the first frame is always compressed using the super-spatial structure prediction algorithm. Once it is compressed, the output is further compressed using Huffman coding for the text.

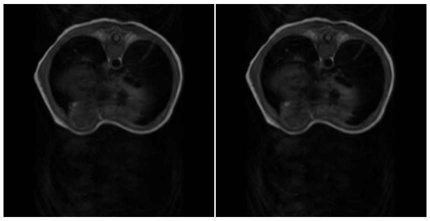

Figure 4. Few images of MRI sequence.

To compress the first frame using the super-spatial structure prediction algorithm, the frame is first converted into fixed-size blocks. In this experiment, the picture is divided into 4 x 4-block size (see Figure 5).

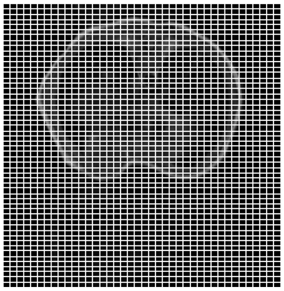

Figure 5. Frame conversion into blocks.

Once the frame is converted into fixed size blocks, the redundant blocks are eliminated from the frame. After the removal of redundant blocks from the frame, the rest of the blocks of frame are further compressed using Huffman coding scheme. After compressing the first frame, it is taken as reference frame for the next frame. Motion estimation and motion compensation (MEMC) are performed in the reference frame and the current frame. The image resulting after ME() and MC () is shown in Figure 6.

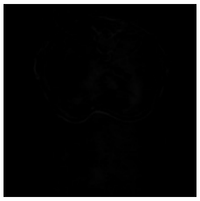

Figure 6. Image resulting after ME(), MC().

The output of the above process is further compressed using the super-spatial structure followed by Huffman coding scheme. The above process continues until the last frame.

4. Decompression Phase

The decompression phase is more simple than the compression phase and requires less number of steps as expressed in Equation 4; the decompression process flow is shown in Figure 7.

$$Decompression = \begin{vmatrix} HMND() + SSP() & & I(X_1, Y_1) \\ & if & \\ AID + BM & & \neq I(X_1, Y_1) \end{vmatrix} \tag{4}$$

In this phase we do not need to calculate motion estimation or perform motion compensation. We create our frame with the help of the reference frame and the frame produced by the image difference. In the decompression phase, the compressed file is first decompressed using Huffman coding *HMND()*. The output of this is then further processed for super-spatial structure prediction decompression SSP(). There, the first frame is decompressed completely and becomes the reference frame for the next frame.

The *AID(). adding image difference*, is defined as “the first and second frame to be now decompressed.” This difference frame is now added into the first frame to create the second frame and becomes the reference frame for the next frame. This process continues until all the frames are decompressed.

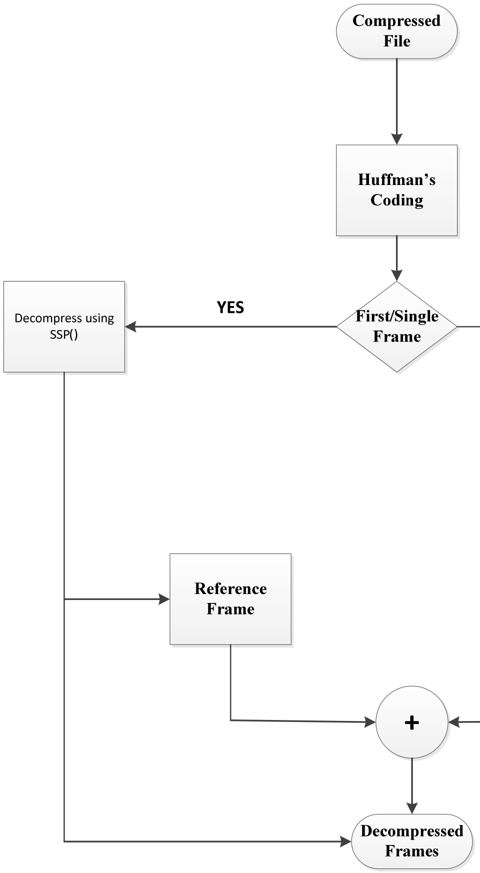

Figure 7. Decompression process.

5. Example Of Decompression Phase

The decompression phase is more simple than the compression phase. First of all, the first frame is decompressed using Huffman decoding scheme followed by the super- spatial structure decoder.

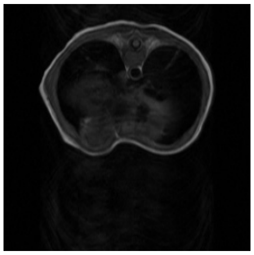

Figure 8. First decompressed frame

Once the first frame is reproduced, the difference image of the rest of the frames is decompressed. The first decompressed frame is shown in Figure 8; it becomes the reference frame for the next frame. To produce the next frame, the reference frame is added into the reference. When the second frame is produced, it now becomes the reference frame for the next frame. This process continues until all the frames are decoded.

6. Experimental Results

To evaluate the performance of the proposed algorithm, we tested the algorithm on a single image as well as on a sequence of images. The sequences we used for this purpose consisted of grey-scale MRI images and color CE images taken also from the MR-TIP database; some of them are shown in Figures 9 and 10. The images in these sequences were of dimensions of 256 × 256 with 8 bits for each color component.

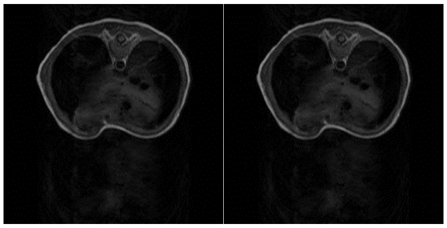

Figure 9. Grey-scale MRI images

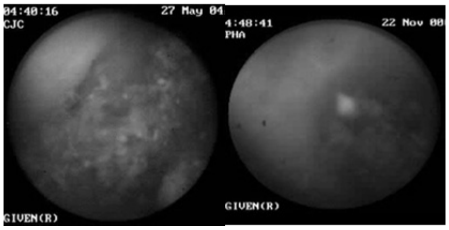

Figure 10. CE color images.

Image	Compression Ratio (CR)		Bits Per Pixel (BPP)	
	JPEG-LS	Proposed	JPEG-LS	Proposed
Tall building	2.22	2.78	3.6	2.87
big tree	2.14	3.18	3.73	2.51
Bridge	1.93	2.52	4.15	3.18
Cathedral	2.24	3.2	3.57	2.5
Deer	1.7	2.99	4.7	2.68
Fireworks	5.45	8.17	1.47	0.98
flower_foveon	3.93	9.88	2.04	0.81
Hdr	3.68	7.32	2.17	1.09
leaves_iso_200	2.09	2.12	3.83	3.77
leaves_iso_1600	1.78	1.93	4.5	4.15
nightshot_iso_100	3.76	6.96	2.13	1.15
nightshot_iso_1600	2.01	3.12	3.97	2.56
spider web	4.51	8.04	1.77	0.99
zone plate	1.08	1.34	7.43	5.97

Table 1. Compression Ratio (CR), bits per pixel (BPP) of single Image.

The algorithm was first tested on individual images. The result of this algorithm was also graphically compared with JPEG-LS as shown in Figure 11. The compression ratio (CR) and the bits per pixel are shown in Table 1. The results showed that the proposed algorithm produced improved results.

Since the purpose of this algorithm was to exploit the correlation among the frames to achieve a higher compression ratio, we tested this algorithm on the same MRI images used in the experiment in [4]. The results of other algorithms tested on these images are also taken from [4]. Results shown in Tables 2 and 3 prove that the proposed algorithm has shielded better results than in the previous work.

Method	CR
JPEG2000	2.596
JPEG-LS	2.727
JPEG-LS+MV+VAR	4.841
Proposed	**6.257**
Gain over JPEG2000	141%
Gain over JPEG-LS	129.4%
Gain over JPEG-LS+VAR+MV	29.3%

Table 2. Compression ratio for MRI Sequence.

Method	BPP
JPEG2000	3.08
JPEG-LS	2.93
JPEG-LS+MV+VAR	1.65
Proposed	**1.28**

Table 3. Bits per pixel (BPP) for MRI Sequence.

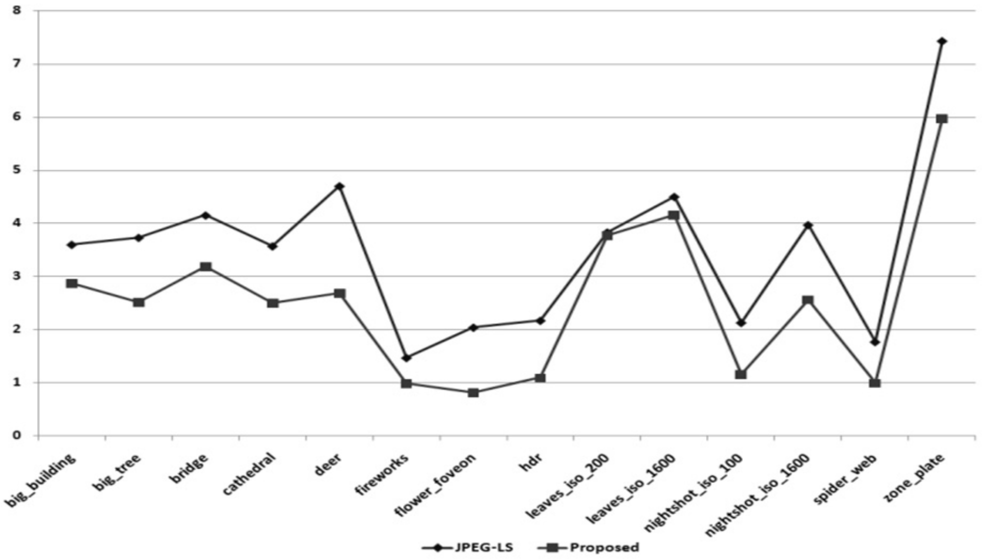

Figure 11. JPEG LS vs. proposed comparative graph.

The proposed algorithm was also tested on a grey-scale MRI image sequence and four color capsule endoscopy (CE) image sequences. Each CE sequence contained 100 color images (8-bit color component). The images chosen for the experiments were the same as the ones used in [4] and the results of earlier algorithms were also taken from [4]. In Tables 4 and 5, the experimental results are shown and prove that the proposed algorithm produced improved results.

Method	1	2	4	6
JPEG2000	3.034	2.902	2.882	2.309
JPEG-LS	3.48	3.342	3.326	2.259
JPEG-LS+VAR+MV	3.74	3.782	4.377	2.775
Proposed	**4.471**	**4.378**	**4.479**	**4.226**
Gain over JPEG2000	47.4%	50.9%	55.4%	83%
Gain over JPEG-LS	28.5%	31%	34.7%	87.1%
Gain over JPEG-LS+VAR+MV	19.5%	15.8%	2.3%	52.3%

Table 4. Compression ratio for CE sequences.

Method	1	2	4	6
JPEG2000	2.637	2.757	2.776	3.465
JPEG-LS	2.299	2.394	2.405	3.541
JPEG-LS+VAR+MV	2.139	2.115	1.828	2.883
Proposed	**1.789**	**1.827**	**1.786**	**1.893**

Table 5. Bits per pixel (BPP) for CE sequences.

7. Comparison

The proposed algorithm was tested for its evaluation on single images as well as on sequences of medical images. MRI sequences were compressed using the proposed algorithm and produced 29.3% improved results for JPEG-LS+MV+VAR and an average of 22.47% improved results for CE image sequence.

8. Quality

To ensure the quality of decompressed images, the utility PSNR stated as the peak signal-to-noise ratio in decibels between two images was used. This ratio was used as a quality measurement between the original and a decompressed image. Better image quality was obtained if the PSNR was higher. To compute the PSNR, it was defined as

$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right) \quad (5)$$

where R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type then R is 1. In our case value, R is 511 because we used 9 bit encoding. To compute R we have:

$$R= 2^{B}-1$$

where B is the number of bits used. In Equation 6, the MSE mean-squared error between the original and decompressed image is defined as

$$MSE=\frac{\sum_{M,N}[I_1(m,n)-I_2(m,n)]^2}{M*N} \qquad (6)$$

where M and N are the number of rows and columns in the input images, respectively. I_1 is the original image and I_2 is the decompressed image. MSE is calculated by means of the original image minus the decompressed image, taking square of the resulting image and dividing by the size of image M*N.

By computing, we get 66.09% PSNR which is comparatively high as it is also the lossless image compression algorithm.

9. Method Resource Utilization Analysis

In this section the proposed compression method for medical image sequences is analyzed in terms of memory and processing time required during compression operation according to specific workload. During the compression phase the method is based upon two individual processes; the super spatial structure prediction algorithm and huffman coding. Suppose the input image is of size 240*140 which means 33600 pixels. For analysis if we take five bits per pixel including operational bits required to process one pixel then it will give 164.062 kilo bits. Adding the control bits reuired for two stages (the super spatial structure prediction and huffman coding), the net memory requirement would be approximately 216.34 kilo bits. Generally the memory reuirement during compression could be defined in expression 7 as:

Memory Requirement = N bits per pixel + operational bits per image

(7)

Similary assuming the workload of 216.34 kilo bits and a system of processing power 1 Gigabits persecond, the system with such processing speed will process the assumed workload in 75 micro seconds. Therefore, it can be observed that the processing time will be directly proportional to the number of bits per pixel and oprational bits as given in expression 8, if the analysis is limited to single frame. But the processing time analysis of number of images or sequences definietly will be more and it would depend on the speed of the processing system.

Processing Time $\propto$ Number of bits per image

(8)

The oprational bits are constant but the image size could vary the number of bits and hence the processing time will increase according to the increasing size of the image.

10. Conclusion

The method presented in this paper makes use of lossless video compression and image compression techniques on medical image sequences to achieve a higher compression rate. The primary goals of the proposed compression method are to minimize the memory resource during storage of compressed data as well as minimize the bandwidth requirement during transmission of compressed data. For achieving these goals, the proposed compression method combines the single-image compression technique called super-spatial structure prediction with interframe coding. The technique that we used for lossless image compression (super-spatial structure prediction) is one of the state-of-the-art techniques and outperformed the JPEG-LS. Also, the two-stage redundant data elimination processes the initial super-spatial structure prediction with fast block-matching ability and later Huffman coding is used which ultimately reduces the memory resource requirement. Therefore, from the results shown in the above comparison tables, we conclude that the proposed method produced much better results than the other lossless image compression algorithms for medical image sequence compression. Since this paper has also exploited the interframe correlation in the form of analyzing motion estimation and motion compensation, the proposed method is also compared with the method proposed in [4]. The results produced by the proposed algorithm were also better the earlier interframe coding algorithm for medical images. In the interframe coding algorithm, the main time taking process was the block-matching process; therefore, there was a need to adopt a technique that could reduce this time. For this purpose we only compared those blocks that had identical sum and left the others; hence, the time taken for the block-matching process is reduced by minimizing unnecessary overhead during the block-matching process. Also, the use of a binary tree is an efficient way for storing block sum and block index which considerably speeded up the block-matching process and, later, the decompression process. In the future, the complexity, efficiency and resource optimization of the presented compression method can be explored.

References

[1] Marcelo J Weinberger, Gadiel. Seroussi, and Guillermo Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEGLS” IEEE Transaction Image Processing, volume 9, no. 8, pp. 1309–1324, August 2000.

[2] Gerald Schaefer, Roman Starosolski and Shao Ying Zhu, “An evaluation of lossless compression algorithms for medical infrared images,” IEEE Engineering Medical Biological Conference, pp. 1673–1676, September 2005.

[3] Xiwen OwenZhao, Zhihai HenryHe, "Lossless Image Compression Using Super-Spatial Structure Prediction", IEEE Signal Processing, Volume. 17, No. 4, April 2010

[4] Shaou-Gang Miaou*, Member, IEEE*, Fu-Sheng Ke, and Shu-Ching Chen “A Lossless Compression Method for Medical Image Sequences Using JPEG-LS and Interframe Coding”, IEEE transactions on information technology in biomedicine, volume. 13, no. 5, September 2009

[5] Y. D. Wang, “The implementation of undistorted dynamic compression technique for biomedical image,”

[6] Master’s thesis, Department of Electrical Engineering at National Cheng Kung University, Taiwan, 2005.

[7] Dania Brunello, Giancarlo Calvagno, Gian Antonio Mian, and Roberto Rinaldo, “Lossless compression of video using temporal information,” IEEE Transaction on Image processing, Volume 12, No. 2, pp 132-139, February 2003

[8] N.D.Memon and Khalid Sayood, “Lossless compression of video sequences,” IEEE Transaction Commun., volume. 44, no. 10, pp. 1340–1345, October. 1996.

[9] M. F. Zhang, J. Hu, and L. M. Zhang, “Lossless video compression using combination of temporal and spatial prediction,” IEEE International Conference Neural Network, Signal Processing, volume. 2, pp. 1193–1196, December 2003.

[10] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Trans. Commun., Vol. COM-29, No. 12, pp. 1799-1808, Dec. 1981.ff