Defence Science Journal, Vol. 59, No. 2, March 2009, pp. 183-188

Median Predictor-based Lossless Video Compression Algorithm for IR Image Sequences

Ram Saran, Hari Babu Srivastava, and Ashok Kumar

Instruments Research and Development Establishment, Dehradun-248 008

ABSTRACT

Lossless image compression has long been recognised as an important need for several applications such as medical imaging, storage of critical IR image sequences, and remote sensing. In this paper, a simple, fast and easy to realisable-on-hardware lossless video compression algorithm is proposed that is well-suited for IR imageries. Context-based median predictor is used for prediction of reference pixels. Three neighboring pixels are used as context for prediction. Inter-frame coding is performed by encoding the redundant pixels in an efficient way, using 1-bit code. Finally, the arithmetic coder is used as entropy coding. The proposed algorithm is able to operate in image compression and video compression mode. The proposed Median Predictor based Lossless Video Compression (MPLVC) algorithm is compared with Joint Pictures Experts Group-Lossless (JPEG-LS) and Fast and Efficient Lossless Image Compression System (FELICS) for compression performance. The results demonstrate that proposed algorithm is superior in encoding rate with added advantage in simplicity and ease in realization on hardware.

Keywords: JPEG-LS, Joint Pictures Experts Group-Lossless, FELICS, Fast and Efficient Lossless Image Compression System, MPLVC algorithm, Median Predictor based Lossless Video Compression, lossless video compression

1. INTRODUCTION

Lossless image compression is becoming increasingly important in many applications such as computerised axial tomography (CAT), magnetic resonance imaging (MRI), storage of IR video data for analysis of small targets, digital cinema, and many more. Number of lossless image compression algorithms[1-5] have already been reported in the literature such as LOCO-I (LOw COmplexity LOssless COmpression for Images), JPEG-LS, CALIC (Context Adaptive Lossless Image Compression), FELICS and SPIHT (Set Partitioning In Hierarchical Tree). Context-based prediction techniques are considered to be most efficient among these[1-5]. The standard algorithms in vogue are JPEG-LS, CALIC and FELICS. The algorithms presented[1-4] are based on the idea of predicting a pixel on the basis of adjacent pixels. Two basic components of the predictive compression algorithms are prediction of current pixel from the information of the previously encoded pixels and coding of prediction error using entropy coding techniques such as Arithmetic coder, Huffman coder, etc.

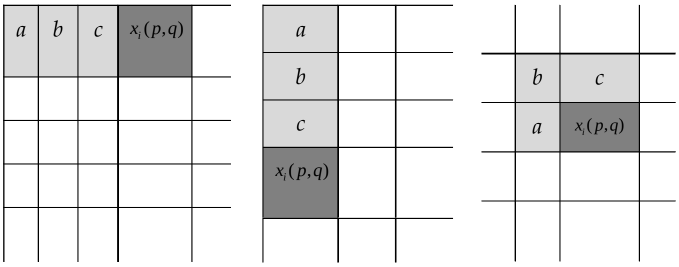
Lossy image and video compression schemes have achieved very high compression ratio and reached a quite matured stage, but lossless compression schemes are still struggling for high performance. While several schemes have been reported for lossless image compression, lossless video compression has not received much attention so far. Lossless video compression is now a hot topic for research community.

In this paper, a simple, fast, and easy to realisable-on-hardware, lossless video compression algorithm, that is well-suited for IR video storage applications, has been proposed. The image frames are encoded and decoded in raster scan order. Context-based median predictor is used for prediction of reference pixels. Three neighbouring pixels are used as context for prediction. Inter-frame coding is performed by encoding the redundant pixels in a very efficient way, using a single-bit code. Finally, the arithmetic coder is used for coding prediction error. The first image frame is encoded by removing the spatial redundancy only. In subsequent image frames, the pixel redundancy is handled using spatial as well as temporal information (temporal information is gathered from the context of previous image frame). This way, redundancy is reduced by a great deal. It can be seen that the algorithm works in image compression mode followed by video compression mode.

2. PROPOSED MPLVC ALGORITHM

In the proposed MPLVC algorithm, an image frame is encoded and decoded in raster scan order for real-time applications. Immediate previous three neighbouring pixels of current image frame were used as context for prediction as shown in Fig. 1. The first three pixels of first row and first column were encoded without compression in the absence of required neighbouring pixels. The context for prediction used for other pixels of first row and first column are shown in Fig.1(a) and Fig.1(b), respectively and the

Received 16 June 2008



(a) (b) (c)

Figure 1. (a) The pixels used for context of prediction for first row only, and (b) the pixels used for context of prediction for first column only. The pixels used for context of prediction for all other pixels.

context for prediction for all other pixels is taken as shown in Fig.1(c). Three neighbouring pixel intensities (*a*, *b*, and *c*) of the current image frame as context for prediction have been used. The *l*, *m*, and *h* are termed as minimum, median, and maximum pixel intensities. The reference pixel is denoted by $x_i(p,q)$, where symbol *i* represents index of image frame, (*p*, *q*) denotes the position of the pixel, and *x* denotes the gray-scale intensity. Here, symbols a, *b*, *c*, *l*, *m* and *h* are referred to their gray-scale pixel intensities. If the intensity of a reference pixel lies in between *l* and *h*, it is called in-range pixel; otherwise it is called out-of-range pixel. The encoding scheme uses one bit to indicate whether $x_i(p,q)$ is in-range or out-of-range. If it is in-range, one more bit is needed to indicate whether $x_i(p,q)$ is less than median, *m* or not, and next few bits are used to specify the exact value. For out-of-range pixel, one bit is used to encode below-range (logic '1') or above-range (logic '0') values for out-of-range values.

Steps of the Algorithm

Image frames of a video sequence are numbered as *i*=0, 1, 2, 3, and so on. Steps of the proposed algorithm for encoding video, operating in video compression mode are:

Step 1 Encode pixels $x_i(p,0)$, *p*=0, 1, 2 and $x_i(0,q)$, *q*=1, 2; without coding.

Step 2 For each, $x_i(p,q)$; if $x_i(p,q)$ is equal to $x_{i-1}(p,q)$, encode the pixel by using one bit code (code '0'is used) and go to step 7, otherwise output '1', as one bit code and go to step 3.

Step 3 For each, $x_i(p,q)$; calculate,
l=min (*a*, *b*, *c*),
m=median (*a*, *b*, *c*),
and, *h*=max (*a*, *b*, *c*)

Step 4 If $l \le x_i(p,q) \le h$, one more bit is used to indicate in-range pixel and code it '0' and go to step 5, otherwise code '1' to indicate out-of-range pixel, and go to step 6.

Step 5 If $x_i(p,q) \ge m$, use one more bit as '0' otherwise code '1'. Use arithmetic coder to encode to the difference of $x_i(p,q)$ and *m* and then go to step 7.

Step 6 If $x_i(p,q) > h$, code '0' to indicate above-range pixel, and then use arithmetic code to encode, otherwise code '1' to indicate below-range pixel, and then use arithmetic code to encode $l - x_i(p,q)$.

Step 7 Increment the counter to point to the next pixel of the image frame, until all pixels have been encoded.

To compress the very first image frame of video, the algorithm operates in image compression mode. As soon as second frame is available, it switches to video compression mode. The proposed algorithm works in image compression mode by removing step 2 of the stated algorithm. The decoding algorithm is just reverse process of encoding algorithm and it can be derived easily.

The proposed algorithm performs better for smooth changes in the neighbouring pixels and is obvious. In this case, the pixel being encoded will be in-range and the values of *l*, *m* and *h* will be close, leading to lower difference between $x_i(p,q)$ and *m* and results in better coding rate.

3. CHOICE OF ENTROPY CODING SCHEME: AN ARITHMETIC CODER

Entropy coder is used to encode the prediction errors using entropy coders such as Huffman coder[6], Arithmetic coder[7] and Golomb-Rice coder[8]. Prediction error is commonly modelled using a Laplacian distribution[9]. Huffman coder and Golomb-Rice coder are fast encoders with reduced complexity. But, efficient entropy coding can be achieved using an adaptive Arithmetic coder[10]. An adaptive arithmetic coderings[9] is used for entropy coding. Entropy coding by using an arithmetic coder[9] is performed by dividing the coding step in two parts: first, the determination of a probabilistic model for the source, and second, the entropy coding that uses the model. In a simple adaptive model, the data stream is encoded symbol by symbol, by updating the probabilistic model after each encoding. Re-normalisation is done to avoid overflow and to provide a way of discounting the past statistics. However, re-normalisation improves exploitation of non-stationarities of the data source; this method is not very effective for encoding the images. The performance of the arithmetic coder used[9] can be further improved by optimising the source code.

4. EXPERIMENTAL RESULTS

The proposed lossless video compression algorithm is tested on a number of video clips, and some of the results are listed here. Details of the video clips used for simulation is given in Table 1. Performance of the proposed MPLVC algorithm on different types of video clips is compared with JPEG-LS code[11] and FELICS executable code[12] as listed in Table 2. The results demonstrate that proposed algorithm is superior in encoding rate. Coding rate is calculated on 8-bit grey-scale values, because all image frames are 8-bit resolution. Validation of lossless mode of image/video compression is done using Linux cmp utility, which compares original image frame and decompressed image frame. It is found that all image frames are decompressed without loss of any pixel information.

Table 1. Description of the video clips

Video Clip No.	Description	Frame size	No. of frames
1.	Video clip containing birds flying in front of a hill (with positive polarity)	500x332	150
2.	Video clip containing birds flying in front of a hill (with negative polarity)	500x332	100
3.	Video clip containing airborne point target	352x288	30
4.	Video clip containing dense cloud	320x244	90
5.	Synthetic video clip containing fast moving cloud	928x264	158
6.	Video clip of moving ground target (vehicle)	352x288	120

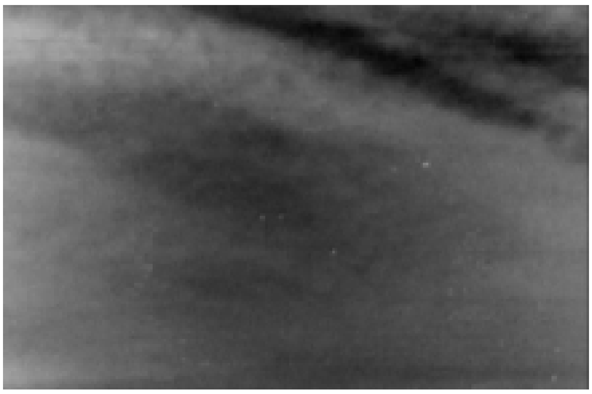
Table 2. Comparison of MPLVC algorithm with standard JPEG-LS and FELICS

Video clip	JPEG-LS		FELICS		Proposed MPLVC Algorithm	
	Compression ratio	Coding rate (bpp)	Compression ratio	Coding rate (bpp)	Compression ratio	Coding rate (bpp)
1.	2.97	2.67	2.92	2.74	3.65	2.19
2.	3.14	2.55	3.00	2.66	3.63	2.20
3.	2.73	2.93	2.31	3.47	8.00	1.00
4.	3.04	2.63	2.43	3.29	9.52	0.84
5.	6.15	1.30	5.01	1.59	10.39	0.77
6.	1.53	5.22	1.49	5.36	2.07	3.85

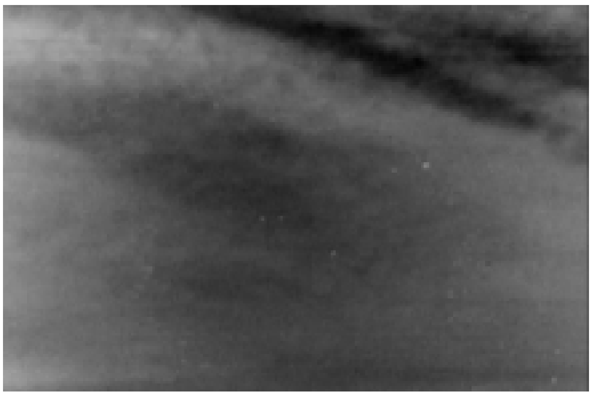
(a)

(b)

Figure 2. (a) Sample original image frame of video clip 1 and (b) Sample decompressed image frame.

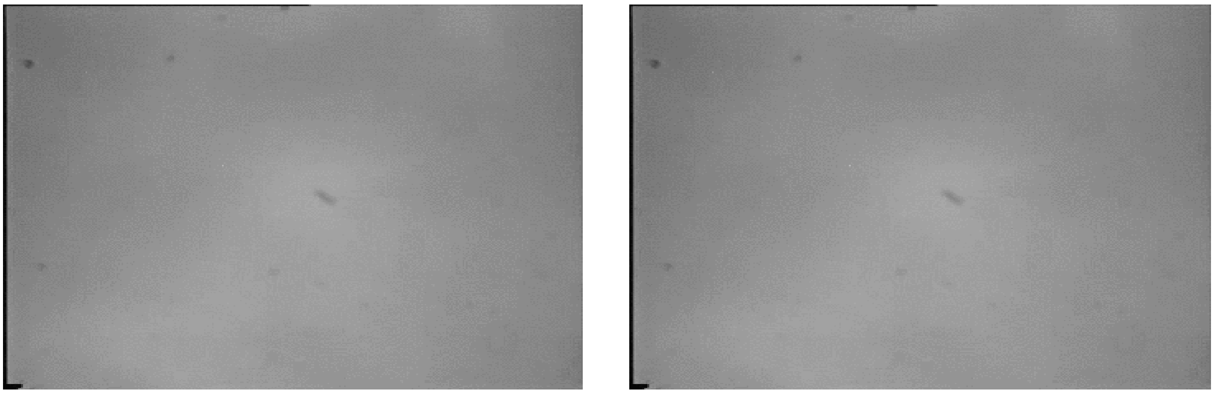


(a)



(b)

Figure 3. (a) Sample original image frame of video clip 2 and (b) Sample decompressed image frame.

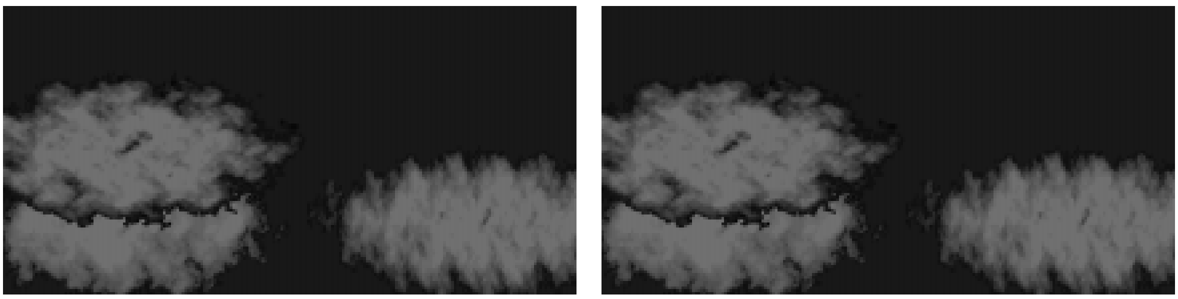


(a) (b)

Figure 4. (a) Sample original image frame of video clip 3 and (b) Sample decompressed image frame.

(a) (b)

Figure 5. (a) Sample original image frame of video clip 4 and (b) Sample decompressed image frame.



(a) (b)

Figure 6. (a) Sample original image frame of video clip 5 and (b) Sample decompressed image frame.

The compression ratio is expressed as original image size or video size divided by compressed size. The 8-bits gray scale images are used for experiments.

$$\text{Compression ratio} = \frac{\text{Original image (or video) size}}{\text{Compressed size}} \quad (1)$$

$$\text{Coding rate (bits per pixel)} = \frac{\text{Compressed size} \times 8}{\text{Original size}} \quad (2)$$

Compression ratio and coding rate as given in Eqns (1) and (2) respectively, and are used as figures of merit for comparing the performance of compression algorithms.

Sample original image frames and their respective decompressed image frames are shown in Figs. 2-7.

5. CONCLUSIONS

A lossless compression algorithm, is presented which operates in image compression mode as well as video compression mode. In video compression mode, temporal redundancy is removed by simple and efficient way, encoding

(a)

(b)

Figure 7. (a) Sample original image frame of video clip 6 and (b) Sample decompressed image frame.

the reference pixel by 1-bit code. Proposed lossless compression algorithm is simple, low complexity, minimal delay and easy to realisation on hardware. Results demonstrate that proposed MPLVC algorithm out performs standard JPEG-LS and FELICS.

ACKNOWLEDGEMENT

The authors thank Shri S.S. Sundaram, Director IRDE for permitting us to publish this work. The authors would like to acknowledge the encouragement given by Shri J.A.R. Krishna Moorty, Ex-Outstanding Scientist and Ex-Director, IRDE, Dehradun. Authors also wish to thank Prof U.B. Desai, Prof. S.N. Merchant of IIT Bombay, Mumbai, India and Dr A.K. Jaiswal, Associate Director of IRDE for reviewing the paper. We also acknowledge Dr S. S. Negi, Associate Director of IRDE for providing video clips of ground targets. Thanks are also due to Shri YB Limbu and Ms Neeta Kandpal for their suggestions and brain storming discussions on the subject.

REFERENCES

1. Sapiro,G.; Seroussi,G.& Weinberger, M. The LOCO-I lossless image compression algorithm: principles and standardisation into JPEG-LS. *IEEE Tans. Image Proc.*, 2000, **9**(8), 1309-324.
2. JPEG-LS; Lossless and near-lossless coding of continuous tone images, ISO/IEC JTC 1/SC 29/WG 1 FCD 14495, July 1997.
3. Wu, X. & Memon, N. Context-based, adaptive, lossless image codec. *IEEE Trans. Comm.*,1997, **45**.
4. Howard, P. G.& Vitter, J. S. Fast and efficient lossless image compression, http://www.cs.duke.edu/~jsv/Papers/HoV93.fitech.pdf.
5. Said, Amir and Pearlman,William A. A new fast and efficient image coder based on set partitioning in hierarchical trees. *IEEE Trans. Circ. Syst. Video Technol.*,1996, **6**, 243-50.
6. Vitter, J. S. Design and analysis of dynamic Huffman codes, *J. ACM*,1987, **34**, 825-45.
7. Witten, I. H. ; Neal, R.& Cleary, J. G. Arithmetic coding for data compression, *Communication ACM*, 1987, **30**, 520-40.
8. Rice, R.F. Some practical universal noiseless coding techniques. *In* Jet Propulsion Laboratory, JPL Publication 79-22, Pasadena, California, March 1979.
9. Howard, P. G. & Vitter, J.S. Error modelling for hierarchical lossless image compression, *In* Proceedings of Data Compression Conference, edited by J.A. Storer and M. Kohn, eds. Snowbird, Utah, March 24-26,1992, 269-78.
10. Wheeler, F. Adaptive arithmetic coding, Source code, http://ipl.rpi.edu/wheeler/ac
11. JPEG-LS: Code. http://spmg.ece.ubc.ca/
12. FELICS: Code. http://www.cs.mu.oz.au/mg/mg-1.2.1.tar.gz

Contributors

Mr Ram Saran obtained his BE in Computer Engineering from Moradabad Institute of Technology, Moradabad (MJP Rohilkhand University, Bareilly) in 2001. He worked as a faculty member at NIEC., Lucknow. He joined DRDO in 2003 and is presently, working with IRDE, Dehradun. His current interests include image processing of small targets, image filtering and lossless (reversible) image compression. He is a member of Computer Society of India.

Mr Hari Babu Srivastava
(*see page no.* 173)

Mr Ashok Kumar obtained his MSc in Computer Science from J.K. Institute of Applied Physics and Technology, University of Allahabad in 1989 and MTech in Computer Science & Engineering from IIT Madras in 2000. Since 1990, he has been with DRDO, working at IRDE, Dehradun, as Scientist. His main research interests include multidimensional signal and image processing with emphasis on detection and tracking of small targets in infrared image sequences.