Australian Journal of Basic and Applied Sciences, 9(20) June 2015, Pages: 418-422

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

LMSE Based Lossy Compression and Recovery of Correlated Video Data

[1]N. Sumathi, [2]K.M. Mehata, [3]N. Nithiyanandam

[1]*Department of Computer Science and Engineering, B.S. Abdur Rahman University, Box. 6000048,Chennai, India*
[2]*Professor & Dean, School of Computer, Information & Mathematical Sciences, B.S. Abdur Rahman University, Box. 6000048 Chennai, India*
[3]*Professor, Department of Electronics and Communication Engineering, B.S. Abdur Rahman University, Box. 6000048,Chennai, India*

ARTICLE INFO

Article history:
Received 16 April 2015
Accepted 12 June 2015
Available online 1 July 2015

Keywords:
Lossy compression; Correlated data compression; LMSE based video compression.

ABSTRACT

In general, consecutive image frames of a video are correlated in their pixel intensities. Video images from any security camera, collected on a 24x7 routine, generate a big data set. A technique for lossy compression of such big data set is developed, using a factorization method based on least mean square error (LMSE) criterion. Image data from 5 consecutive frames of a video are compressed. Then the compressed video data set is decompressed to reconstruct and recover an approximation of the 5 video frames. The mean square error (MSE) of each reconstructed frame is estimated along with the frame correlation coefficient. From the analysis, it is inferred that higher correlation among images results in reduced error in reconstructed images. Although the compression is lossy, the visual quality of the reconstructed images is good.

To Cite This Article: N. Sumathi, K.M. Mehata, N. Nithiyanandam, LMSE Based Lossy Compression and Recovery of Correlated Video Data. ***Aust. J. Basic & Appl. Sci.,*** *9(20): 418-422, 2015*

INTRODUCTION

Section 1: Introduction

Multimedia data include text, audio and video in digital form of bits (1's and 0's). Data compression is essential for optimum use of storage space in computers. Text data compression should be lossless, since lossy compression will result in intolerably erroneous data on reconstruction using decompression. On the other hand, audio and video data can have tolerable amounts of errors in the decompressed data. In particular, the quantum of data bits in video is large requiring more bits to be compressed to achieve any specified 'compression factor' (CF). A number of techniques for lossy image compression and lossless image compression have been developed. This paper describes a lossy compression technique suitable for compression of correlated video data, usually found in the consecutive image frames of any video.

Section 2 describes the mathematical aspects of lossy compression and decompression techniques using an example of 5 sets of correlated data of decimal numbers.

Section 3 deals with the application of this compression and decompression technique to correlated image data of five consecutive frames of a video.

Section 4 deals with the performance analysis of the correlated image data compression technique.

The concluding Section 5 lists the benefits of the compression technique and indicates the scope for further work in correlated video data compression.

Section 2: Lossy compression and decompression of correlated data sets

Computers handle all data in binary form using bits of 0's and 1's. However for human understanding, digits (decimal numbers) are more convenient to handle. Image is made of pixels (picture elements) represented by varying light intensities. The values of light intensity form the image data. Since light intensity is always positive, these intensity values vary from a minimum of zero for a black pixel, to a maximum positive value for a white pixel. In computers, intensity of each image pixel is represented by 8 bits. This corresponds to any quantized level from 00000000 (decimal 0) to 11111111 (decimal 255).

To explain the mathematical aspect of correlated data compression, five correlated sets of each 13 decimal numbers, in the range of 1 to 256, are generated as listed in Table 1. The sum of the 13 numbers of a set is shown in the last column.

Corresponding Author: N. Sumathi, Department of Computer Science and Engineering, B.S. Abdur Rahman University, Box. 600048, Vandalur, Chennai, India.
E-mail: mail_nsumathi@yahoo.com

Table 1: The 5 sets of correlated decimal numbers.

Set 1	9	13	26	36	44	54	66	76	86	96	109	111	127	*Sum* 853
Set 2	3	19	21	39	46	51	63	79	81	99	104	113	124	*Sum* 842
Set 3	1	14	24	33	49	56	69	73	83	93	101	116	122	*Sum* 834
Set 4	4	16	23	31	43	59	61	74	84	91	106	119	123	*Sum* 834
Set 5	6	11	29	34	41	55	64	71	89	94	103	114	120	*Sum* 831

The set 1 with the largest sum of 853 is taken as the reference set, which will be used to generate the 'factors' (F1 to F5) for the five sets. The procedure to calculate F1 to F5 is based on the definition (Perumalsamy and Natarajan, 2013) that 'F' of a set, is the ratio of the sum of the products of the 13 numbers of that set with the corresponding position numbers of the reference set to the sum of the squares of the 13 numbers of the reference set1.

For example, F2 of set2 is calculated with reference to set1 as

$$\frac{(3)(9)+(19)(13)+(21)(26)+(39)(36)+(46)(44)+(51)(54)+(63)(66)+(79)(76)+(81)(86)+(99)(96)+(104)(109)+(113)(111)+(124)(127)}{(9)(9)+(13)(13)+(260(26)+(36)(36)+(44)(44)+(54)(54)+(66)(66)+(76)(76)+(86)(86)+(96)(96)+(109)(109)+(111)(111)+(127)(127)}$$

This F2 works out to be 0.9883. Similarly F3 = 0.9811, F4 = 0.9849, F5 = 0.9760 and, by definition, F1 for set 1 is obviously 1.

The original 5 sets have a total of 65 (5 x13) numbers. The compressed data has 18 values (13 numbers of reference set 1 and 5 factors, from F1 to F5). Now we reconstruct an approximation of these 5 sets by estimating the values of the 13 numbers of a set by multiplying the F value of that set with the corresponding position number in the reference set 1. Obviously F1 being 1, all 13 numbers of set 1 will be exactly reconstructed. The reconstructed numbers for all other sets will be a close approximation to their original values. The error, which is the difference between the reconstructed number and the original number, is used to estimate the Mean Square Error (MSE) of the set. By taking the square root of MSE, we get the Root Mean Square Error (RMSE) of the set.

MSE = [Sum of the squared errors for the 13 numbers of a set] / 13.

The Peak Signal to Noise Ratio (PSNR) of the set is then calculated as:

PSNR in dB (decibel) = 20 log10 (Maximum of the 13 numbers of the original set / RMSE of the set)

Since all the 65 (5x13) numbers of the original sets are reconstructed using only 18 values, the Compression Factor (CF) of this data compression process is 3.61 (65/18). The reconstructed data sets with their factors (F) are tabulated in Table 2.

Table 2: The 5 sets of reconstructed data.

Reconstructed Set 1	9	13	26	36	44	54	66	76	86	96	109	111	127	F1=1
Reconstructed Set 2	9	13	26	36	43	53	65	75	85	95	108	110	123	F2=0.9883
Reconstructed Set 3	9	13	25	35	43	53	65	74	84	94	107	109	124	F3=0.9811
Reconstructed Set 4	9	13	26	35	43	53	65	75	85	95	107	109	125	F4=0.9849
Reconstructed Set 5	9	10	25	35	43	53	64	74	84	94	104	108	124	F5=0.976

Error (E) = Number in the reconstructed set – Corresponding number in the original set. The Table 3 lists the RMSE and PSNR values for each set. It is observed that the Set 1 has no error. This is obvious because it is the reference set and hence its factor F1 = 1. Reference set is 100% correlated with itself.

Table 3: The RMSE and PSNR values for each set.

(Error)2 Set 1	0	0	0	0	0	0	0	0	0	0	0	0	0	RMSE=0	PSNR ∞dB
(Error)2 Set 2	36	36	25	9	9	4	4	16	16	16	16	9	1	RMSE=3.89	PSNR 30.069dB
(Error)2 Set 3	64	1	1	4	36	9	16	1	1	1	36	49	4	RMSE=4.14	PSNR 29.1798dB
(Error)2 Set 4	25	9	9	16	0	36	16	1	1	16	1	100	4	RMSE=3.21	PSNR 31.668dB
(Error)2 Set 5	9	4	16	1	4	4	0	9	25	0	9	36	16	RMSE=3.2	PSNR 31.4806dB

Correlation between two sets of numbers is defined as follows. Let Set 1 = {x1,x2 ....xN} with a mean of X and standard deviation of σx, Set 2 = {y1, y2 ....yN} with a mean of Y and standard deviation of σy.

The correlation coefficient 'CC' = $$\frac{(x1-X)(y1-Y)+(x2-X)(y2-Y)+\ldots\ldots\ldots+(xN-X)(yN-Y)}{N(\sigma x)(\sigma y)}$$

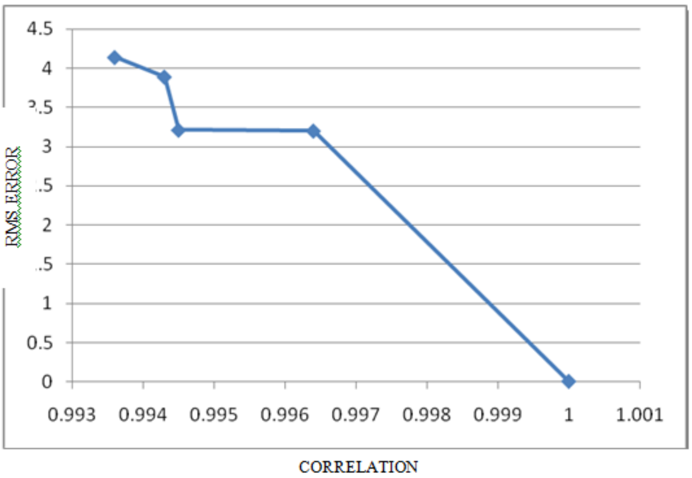
where standard deviation (σx) of Set1 is = $$\sqrt{\frac{1}{N}(x1-X)^2+(x2-X)^2+\ldots\ldots\ldots+(xN-X)^2}$$

and standard deviation (σy) of Set 2 is $= \sqrt{\frac{1}{N}(y1-Y)^2 + (y2-Y)^2 + \ldots\ldots\ldots + (yN-Y)^2}$

The correlation coefficients of the 5 sets of numbers in Table 1, with the reference Set1 are calculated and listed with the RMSE of the sets, in Table 4. Table 4 is plotted as Graph 1. It is observed that the error decreases as the correlation increases.

Table 4: The Correlation Coefficient and RMSE of 5 sets.

Set	Correlation Coefficient	RMSE
1	1	0
2	0.9943	3.89
3	0.9936	4.14
4	0.9945	3.21
5	0.9964	3.2



Graph 1: Plot of RMSE versus Correlation Coefficient.

Section 3: Compression of video data

The video images are captured in frame format @ 30 frames per second, to create flicker-free visual effects. Capture time for a single frame is 33.33 millisecond. The image contents in consecutive frames are highly correlated, since very little change is expected in frames in such a short time span of 33.33 millisecond. A set of 5 consecutive frames are taken from a video and subjected to lossy compression, explained in Section 2.

The video image considered is of format 160x120 pixels per frame denoting width x height. The corresponding image intensity data is available as a matrix of 120 rows and 160 columns. The first row (160 pixels) data from each of the 5 consecutive frames form 5 sets of each 160 numbers. They are compressed and then reconstructed, as explained in Section 2. This compression and reconstruction is carried out for all the 120 rows of the 5 matrices. Thus the 5 image frames are first compressed and then reconstructed. The original image frame and the reconstructed image frame using the compressed data are shown in Figure 1.

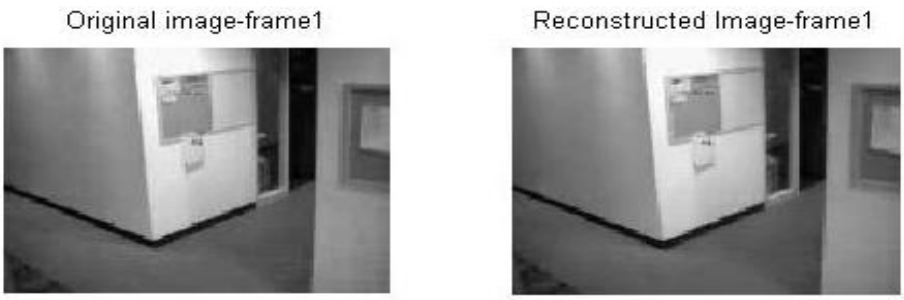


Fig. 1:

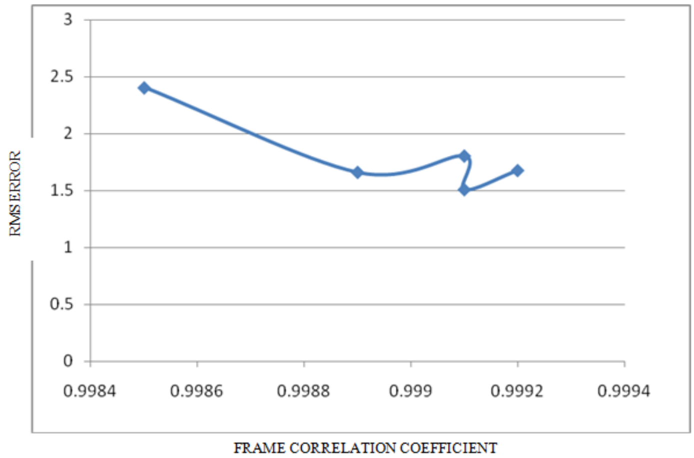
Section 4: Performance analysis of video data compression

In this section, we present the parameters associated with data compression. The 120x160 data matrix of a frame has 160 numbers in each row. Using the matrices of 5 consecutive frames, row by row image compression is done as detailed in Section 3 .This generates a set of 5 'factors' (F1 to F5) for each set of 5 rows.

The 160 numbers of the reference row and the 5 factors are used to reconstruct 800 (5x160) numbers of the set of 5 rows. Hence the Compression Factor (CF) is 4.848 (800/165). This CF and the values of RMSE, PSNR, Processing Time and Correlation Coefficient are shown in Table 5. The image pixel intensity is an 8 bit data from an 8 bit quantizer with a maximum of 256 steps. For a 1 volt peak video signal, the step size Δ is 3.90625 millivolt (1/256). Hence the RMS Error is calculated in Δ units. It is observed that the error decreases as the correlation increases. From Table 5, the Frame Correlation Coefficient and RMSE are plotted in Graph2.

Table 5: RMSE, PSNR, Processing Time, CF & Correlation Coefficient.

Image Frame Number	RMS Error In Δ units	PSNR Value in dB	Image Processing Time in Seconds	Compression Factor	Frame Correlation Coefficient
1	2.4041	88.5744	0.223099	4.848	0.9985
2	1.6597	91.7928	0.206051	4.848	0.9989
3	1.8037	91.0702	0.231801	4.848	0.9991
4	1.5073	92.6295	0.215481	4.848	0.9991
5	1.6759	91.7083	0.212001	4.848	0.9992



Graph 2: Plot of RMSE versus Frame Correlation Coefficient.

Section 5: Conclusion and scope for further work

From the image frame in Fig.1, shown in Section 3, the visual quality of the compressed image frame is as good as the original image frame. From Table 5 and Graph 2, it is evident that the RMS Error decreases as the Frame Correlation increases. The image Compression Factor of nearly 5 (4.848) indicates that 80% of the storage space required for the uncompressed images is saved by compression. Correspondingly, the time for compressed image data transfer will also be only 20% of time required for uncompressed images. The processing time for image compression is under 1 second, which is good enough for most of the real time applications. Currently video cameras for security are extensively used, with a need for storage of large video data collected on a 24x7 basis. In such applications, image compression is highly useful.

REFERENCES

Perumalsamy, N. and N. Natarajan, 2013. "A non-iterative method for factorization of positive matrix in discrete wavelet transform based image compression", American Journal of Applied Sciences, 10: 684-688.

Taubman, D., 2000. "High performance scalable image compression with EBCOT", IEEE transactions on Image Processing, 9(7): 1158-1170.

Delp, E.J., P. Salama, E. Asbun, N. Saenz and K. Shen, 1999. "Rate scalable image and video compression techniques",42nd Midwest IEEE Symposium on Circuits and Systems, 2: 635-638.

Said, A. and W.A. Pearlman, 1996. "A new, fast and efficient image codec based on set partitioning in hierarchical trees", Transactions on Circuits and Systems for Video Technology, 6(3): 243-250.

Dony, R.D. and S. Haykin, 1995. "Neural network approaches to image compression", Proceedings of the IEEE, 83(2), 288-303.

Koc, B., Z. Arnavut and H. Kocak, 2013.

"Lossless Compression of Dithered Images," IEEE Photonics Journal, 5(3): 508-517.

Yerva, S., S. Nair and K. Kutty, 2011. "Lossless image compression based on data folding",IEEE International Conference on Recent Trends in Information Technology (ICRTIT- 2011), pp: 999-1004.

Zhao, X.O. and Z.H. He, 2010. "Lossless image compression using super-spatial structure prediction", IEEE Signal Processing Letters, 17(4): 383-386.

Pan, H., W.C. Siu and N.F. Law, 2007. "Lossless image compression using binary wavelet transform", IET Image Processing, 1(4): 353-362.

Boulgouris, N.V., D. Tzovaras and M.G. Strintzis, 2001. "Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding", IEEE Transactions on Image Processing, 10(1): 1-14.

Mandyam, G., N. Ahmed and N. Magotra, 1997. "Lossless image compression using the discrete cosine transform", Journal of Visual Communication and Image Representation, 8(1): 21-26.

Weinberger, M.J., G. Seroussi and G. Sapiro, 1996. "LOCO-I: A low complexity, context-based, lossless image compression algorithm",. IEEE conference on Data Compression (IEEE - DCC'96), pp: 140-149.

Delp, E. and O. Mitchell, 1979. "Image compression using block truncation coding", IEEE Transactions on Communications, 27(9): 1335-1342.

Wallace, G.K., 1992. "The JPEG still picture compression standard", IEEE Transactions on Consumer Electronics, 38(1): xviii-xxxiv.