International Journal of Trend in Research and Development, Volume 2(4), ISSN 2394-9333
www.ijtrd.com

Video Compression System for Online Usage Using DCT

[1]S.B. Midhun Kumar, [2]Mr.A.Jayakumar M.E
[1]UG Student, [2]Associate Professor
Department Electronics and Communication Engineering
IFET College of Engineering
Villupuram, Tamilnadu.

Abstract: *This paper provides an efficient video compression using a simple transform called Discrete Cosine Transform (DCT) which gives the compression ratio more than that of the all the existing algorithms. At the same time by maintaining the video quality good while compressing the video. Also this algorithm will execute fast and consumes less time. Simulation results show that a reduction of video size gives 50% more compression.*

I. INTRODUCTION

Before going to video compression we have to know about how compression can be done by using Discrete Cosine Transform. Normally compression is of two types lossy and lossless. For compression of text files we use the lossless compression because while compressing the text files we have to be clear that the data in the text files should not be lost on the other side, For video or image compression we will prefer the lossy compression why because in this video or image compression the background will not change for the some part so we are going to concentrate more on background means the low frequency components of the image are discarded thereby we can achieve more compression. There are different types of compression standards some of them are MPEG-2, MPEG-4, H.261, H.264 etc., these standards use different types algorithms, of those algorithms this DCT algorithm is the most prominent algorithm for compression. The process of DCT algorithm is explained here clearly.

The overview of DCT process is, first the input image is broken into 8x8 blocks of pixels. Then DCT is applied on each block. The DCT works by separating the image into parts of different frequencies. Now each block is compressed using the quantization technique. During this step a part of compression actually occurs, the less important frequencies are discarded. The image is now stored in the reduced amount of space. When desired the image is reconstructed through decompression, a process that uses the Inverse Discrete Cosine Transform is used.

Before we begin to discuss about the DCT, it should be noted that the pixel values of black and white image range from 0 to 255 in steps of 1, where pure black is represented by 0 and pure white is represented by 255 let's start with a block of image pixel values. This particular block was chosen from very upper left hand corner of the image.

Because the DCT is designed to work on the values ranging from -128 to 127, the original block is leveled off by subtracting 128 from each entry. After matrix multiplication now we are ready for the compression by quantization [4]. The highly useful feature in this process is that, varying levels of image compression and quality are obtainable through selection of specific quantization matrices. This enables the user to decide on quality levels ranging from 1 to 100. Where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression. Subjective experiments involving the human visual systems have resulted in the standard quantization matrix with a quality level of 50 the matrix renders both high compression and excellent decompressed image quality. The scaled quantization matrix is rounded off to have positive integer values ranging from 1 to 255.

The quantized matrix is now ready for the final step of compression. Before storage all coefficients of are converted by an encoder to a stream of binary data (0110101010…). After quantization it is quite common that for most of coefficients equal zero. JPEG takes the advantage of this by encoding quantized coefficients in zig-zag sequence. The advantage lies in the consolidation of relatively large number of zeros, which compress very well. Reconstruction of our image begins by decoding the bit stream representation.

II. EXISTING METHOD

The existing algorithm for compression is that the Block Matching Algorithm (BMA). It is a way of locating matching blocks in a sequence of digital video frames for the purposed of video compression.

Motion estimation using a block-matching algorithm (BMA) is widely used in many motion-compensated video coding systems, such as those recommended by the H.261 and MPEG standards [6], [7], to remove interframe redundancy and thus achieve high data compression. In a typical BMA, the current frame of a video sequence is divided into nonoverlapping square blocks of pixels, say, of size N X N. For each reference block in the current frame, BMA searches for the best matched block within a search window of size (2W + N) X (2W + N) in the previous frame, where W stands for the maximum allowed displacement. Then the relative position between the reference and its best matched block is represented as the motion vector of the reference block. Region-Of-Interest (ROI) coding of video using computational models of visual attention [5], has been recognized as a promising approach to achieve high-performance video compression. The idea behind most of these methods is to encode an area around the predicted attention grabbing (salient) regions with higher quality compared to other less visually important regions.

The extension for the BMA algorithm is Fast Full Search Block Matching Algorithm (FFBMA) [27] and 3 step hierarchical search block matching algorithm [20]. The basic idea behind this fast full-search BMA relies on constructing three fast matching criteria and, during the period of block matching, employing these three fast matching criteria to discard the candidate blocks in the search window which are not matched to the reference block in the current frame, before using the time-consuming matching error. These fast matching criteria are derived from the integral projections since the integral projections are simple and relevant features to a block of pixels. Roughly speaking, the integral projections can be regarded as the intensity sums of spatial pixels along any fixed direction in a block of pixels. For any given block in frame three kinds of integral projections are defined as vertical projections, horizontal projections and massive projection.

Video quality will be degraded. Performing the image recovery and noise suppression is less and frame recombination is difficult.

III. PROPOSED METHOD

Our method for video or image compression is discrete cosine transform. DCT is an image compression algorithm that samples an image at regular intervals, analyzes the frequency components present in the sample, and discards those frequencies which do not affect the image as the human eye perceives it. The frequencies which are discarded are low frequencies hence it does not show much difference.

Discrete Cosine Transform is common for spatial redundancies. The proposed method aims at reducing salient coding artifacts in non-ROI parts of the frame in order to keep user's attention on ROI [2]. Further, the method allows saliency to increase in high quality parts of the frame, and allows saliency to reduce in non-ROI parts. Experimental results indicate that the proposed method is able to improve visual quality of encoded video relative to conventional rate distortion optimized video coding, as well as two state-of-the art perceptual video coding methods.

For the process of the compression we will use 2D version of DCT For analysis of two-dimensional (2D) signals such as images, we need a 2D version of the DCT. For an n x m matrix s, the 2D DCT is computed in a simple way. Since the 2D DCT can be computed by applying 1D transforms separately to the rows and columns, we say that the 2D DCT is separable in the two dimensions.

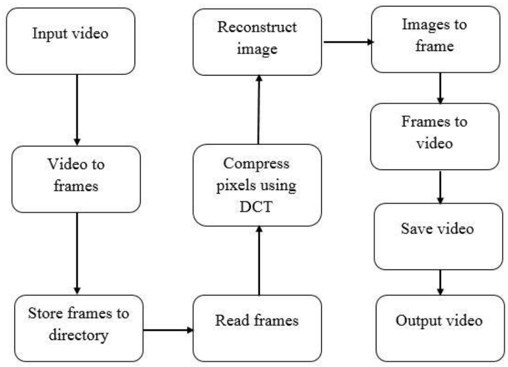

Fig.3.1Video compression system using DCT algorithm

Figure 1 explains about the video compression using the DCT technique. The process of this video compression using the DCT is as follows. First we will convert the input video into the frames by using the some MATLAB coding. These frames will be stored in the directory then these frames will be read and these pixels of the frames will be adjusted and compressed by using DCT then the images will be reconstructed. These images will be converted into frames and then frames to video. At last the obtained output video will be saved.

We proposed a saliency-preserving framework for region-of-interest (ROI) video coding, whose main goal was to reduce attention-grabbing coding artifacts in non-ROI parts of the frame in order to keep viewer's attention on ROI parts where the video quality was higher. The method proposed in was based on the algorithm called Discrete Cosine Transform. In this transform we are going to find a quantization parameter (QP) matrix for each and every video frame. So that the difference between the saliency map of the coded frame and the saliency map of original raw frame was minimized under a given target bit rate.

STEPS:

1. Video to frames and noise elimination
2. Frame resolution adjustment
3. Image compression module
4. Output frame assembling as video

1. Video to frames and noise elimination:

At first we convert the video to frames by taking .avi video as input and any number of frames per seconds like (20 FPS).FPS means for one second video 20 frames will be the output. Normally 24 frames will gives the perfect video quality. While compression we can take any number of frames per second but if we increase the number of frames then the frame recombination and alignment will be difficult.

We propose to write the code in such a way that we can get output frames in any format like .jpg, .png, .tiff etc…. The image frames will be saved serially like 1.jpg, 2.jpg, 3.jpg, 4.jpg.

2. Frame Resolution Adjustment:

The resolution of frames will be adjusted to maintain the quality or pixel distortions but without increasing the size of the image. This can be achieved by taking care of the Region of Interest (ROI) parts alone in the image and concentrating less on the non ROI parts. Thereby we can be able to achieve the good video quality without increasing the size of the video. Hence the output video will be having good quality.

3. Image Compression Module:

At first we will convert the video to frames this can be done by using the software called MATLAB. These obtained frames will be saved as images then the image compression is done using the Discrete Cosine Transform. Because this technique is most prominent, less time consuming and gives more compression ratio. The same compression will be repeated for all the frames obtained from the video.

4. Output Frame Assembling as Video:

This is another crucial process in the compression process. The compressed frames will be reassembled in the same order how we converted the video into the frames. The output video will be tested for the size.

IV. SIMULATION RESULTS

Simulation results shows that the compression achieved by the DCT algorithm is high with a good quality of video. For conversion of video to frames and to video we will use the Graphical User Interface which gives the user friendly environment.

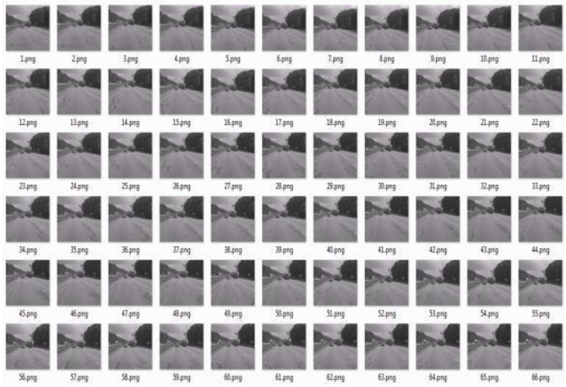

Fig 4.1 frames of the input video after conversion

After the conversion of input video into the frames the frames are arranged in the order that is 1png, 2png,
3png……. As shown in the fig 4.1

The output video after the compression using the DCT algorithm is shown in the fig 4.2 which shows that the quality of the video after the compression

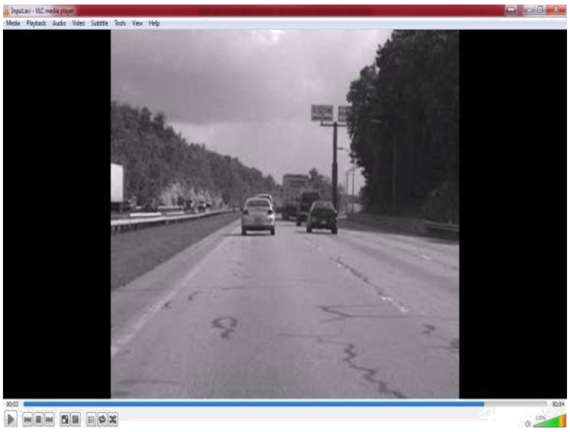

Fig 4.2: output video after the compression

This video is the compressed video which is having the less size than the original file and the quality of the video will be almost equal to that of the original input video.

V. CONCLUSION

In this paper we have presented a compression technique called Discrete Cosine Transform. This DCT quantizes the image and also provides security in which the other algorithms will not provide. This method attempts to reduce the effect of non ROI artifacts in the video and increase the effect of ROI artifacts. This will be done by concentrating more on the ROI parts of the image. The experimental results show that the proposed method gives the more compression ratio and also improves quality of the video when compared to all the remaining algorithms. The framework could be further enhanced by optimizing the saliency-related Lagrange parameter, possibly on a block-by-block basis.

REFERENCES:

[1] Eric. C. Larson., Damon M.Chandler., "Most Apparent Distortion: Fullreference Image Quality Assessment And The Role Of Strategy" journal of Electronic Imaging 19(1), 011006 (Jan-Mar 2010)

[2] Y. Liu, Z. Li, and Y. C. Soh,, "Region-Of-Interest Based Resource Allocation For Conversational Video Communications Of H.264/Avc" IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 1, pp. 134–139, Jan. 2008

[3] A. Leontaris, P. C. Cosman, and A. R. Reibman, "Quality Evaluation Of Motion-Compensated Edge Artifacts In Compressed Video," IEEE TRANS. Image process., vol. 16, no. 11, pp. 943–956, apr. 2007.

[4] M. A. Robertson and R. L. Stevenson., "Dct Quantization Noise In Compressed Images" IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 1, pp. 27–38, Jan. 2005.

[5] L. Itti., "Automatic Foveation For Video Compression Using A Neurobiological Model Of Visual Attention" IEEE Trans Image Process., vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[6] CCITT SG XV, "Recommendation H.261—Video codec for audiovisual services at p*64 kbits/s," Tech. Rep. COM XV-R37-E, Aug. 1990.

[7] MPEG, "ISO CD 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 M bits/s," Nov. 1991.

[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf. New Orleans, LA, Nov. 1981, pp. G5.3.1–G5.3.5.

[9] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COM-29, pp. 1799–1808, Dec. 1981.

[10] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," IEEE Trans. Commun., vol. COM-33, pp. 888–896, Aug. 1985.

[11] S. Kapagantula and K. R. Rao, "Motion predictive interframe coding," IEEE Trans. Commun., vol. COM-33, pp. 1011–1015, Sept. 1985.

[12] A. Puri, H. M. Hang, and D. L. Schilling, "An efficient block-matching algorithm for motion-compensated coding," in Proc. Int. Conf. Acoust., Speech, Signal Processing, 1987, pp. 25.4.1–25.4.4.

[13] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Commun., vol. 38, pp. 950–953, July 1990.

[14] C. H. Hsieh, P. C. Lu, J. S. Shyn, and E. H. Lu, "Motion estimation algorithm using interblock correlation," Electron. Lett., vol. 26, pp. 276–277, Mar. 1990.

[15] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, pp. 148–157, Apr. 1993.

[16] L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie,"Dynamic searchwindow adjustment and interlaced search for block-matching algorithm," IEEE Trans. Circuits Syst. Video Technol., vol. 3, pp. 85–87, Feb. 1993.

[17] M. J. Chen, L. G. Chen, and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," IEEE Trans. Circuits Syst. VideoTechnol., vol. 4, pp. 504–509, Oct. 1994.

[18] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," Proc. IEEE, vol. 83, pp. 858–876, June 1995.

[19] K. M. Yang, M. T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," IEEE Trans. Circuits Syst., vol. 36,

pp. 1317–1325, Oct. 1989.

[20] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Parallel architectures for 3step hierachical search block-matching algorithm," IEEE Trans. Circuits Syst. Video Technol.. vol. 4, pp. 407–416, Aug. 1994.

[21] J. S. Kim and R. H. Park, "A fast feature-based block matching algorithm using integral projections," IEEE J. Select. Areas Commun., vol. 10, pp. 968–971, June 1992.

[22] Y. H. Fok and O. C. Au, "An improved fast feature-based block motion estimation," in Proc. IEEE 1994 Int. Conf. Image Processing, 1994, pp. 741–745.

[23] H. B. Park and C. Wang, "Image compression by vector quantization of projection data," IEICE Trans. Inform. Syst., vol. E75-D, pp. 148–155, Jan. 1992.

[24] K. H. Jung and C. Wang, "Projective image representation and its application to image compression," IEICE Trans. Inform. Syst., vol. E79-D, pp. 136–142, Feb. 1996.

[25] C. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," IEEE Trans. Commun., vol. COM-33, pp. 1132–1133, Oct. 1985.

[26] S. H. Huang and S. H. Chen, "Fast encoding algorithm for VQ-based image coding," Electron. Lett., vol. 26, pp. 1618–1619, Sept. 1990.

[27] C. H. Lee and L. H. Chen, "A fast search algorithm for vector quantization using mean pyramids of codewords," IEEE Trans. Commun., vol. 43, pp. 1697–1702, Feb./Mar./Apr. 1995.