360-degree Video Compression Using Different Block Matching Algorithms

Saad Alkhalifah and Alfonso Diaz

Abstract— **This document analyses different block search algorithms used in video compression. Samples from two different 360-degree videos with a resolution of 360p were encoded and decoded using developed MATLAB code. The computation varied each time under various conditions such as different block matching cost functions and search ranges. The main block search algorithms visited in this investigation are the Exhaustive Search (ES) Algorithm which is the most straight forward but also most computationally intensive approach due to its the lack of efficiency. However, the ES produces the highest Peak-Signal-to-Noise-Ratio (PSNR) overall with respect to other methods. The Diamond Search (DS) Algorithm is also visited in this paper and although it does not generate as much overall PSNR as ES, their values are very similar, the main advantage of DS however resides on its more efficient and less computational expensive process.**

I. INTRODUCTION

The beginnings of practical video compression started with the International Telecommunication Union (ITU) H.261 standard set in 1990. Its scheme was significantly different from its predecessor H.120 (standardized in 1984) since H261 utilized a hybrid video coding scheme for the very first time. The basics of hybrid video coding can be summarized as follows: Estimation and compensation from previous frames is calculated and the information generated used to predict frames. In addition to this, the hybrid video coding also transforms and quantify the data after which the data is encoded using a loss-less compression method such as Huffman or Arithmetic Coding.[1]

New standards such as H.264/MPEG-4 AVC were developed on the with similar characteristics, the most important of which was the motion estimation algorithms.[1] The first widely adopted motion estimation algorithm was ES due to its high PSNR, however this search method was later replaced by more cost efficient algorithms such as Three Step Search (TSS), New Three Step Search (NTSS), Simple and Efficient (SES), four Step Search (4SS), Diamond Search (DS), and Adaptive Rood Pattern Search (ARPS).

In our investigation we examined and compared the results using the Exhaustive Search Algorithm and the Diamond Search Algorithm to compress two 360-degree videos resulting in operational rate-distortion curves and PSNR per frame for every case.

II. BLOCK MATCHING ALGORITHMS

A. Exhaustive Search

This algorithm uses one of the cost functions, mentioned in this paper, to find the best matching block in the reference frame within the widow search. Each matching block will have the highest PSNR. The drawback of this method is when the search window size increases, the number of computations increases which makes this algorithm the most computationally extensive method among all.

B. Diamond Search

The Diamond search uses a center biased searching scheme and employs provisions for a half way stop, reducing the computational cost of the algorithm. Initially a fixed step size of 2 blocks is set forming a search point pattern shaped like a diamond. Thus we initially look at 9 locations, including the center of the search window. The Diamond Search uses two types of search patterns: Large Diamond Search Pattern and Small Diamond Search Pattern. All steps in the procedure expect the last one use the Large Diamond Search Pattern. During the first step the weights are calculated using the cost function, if the least weight is found to be at the center of the search window, the procedure moves on to the last step. If this is not the case and the least cost is found at one of the surrounding elements, this element becomes the center of the diamond search area and the cost for all 9 elements are calculated and the process iterates again. It is important to note that there is no limit in the number of steps. When the last step is reached, the Small Diamond Search Pattern is used and the location with the smallest weight is considered to be the best match.[2][3]

III. COST FUNCTIONS

The cost functions are used to find the 'best matching' block inside the desired search range. For each cost function computation we use two blocks from different frames which outputs a single value that measures the similarity between the two chosen blocks. From a selection of blocks, the block for which pair results in the lowest value is used as the best match.

A. SAD (Sum of Absolute Differences)

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \tag{1}$$

B. MAD (Mean of Absolute Differences)

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \tag{2}$$

IV. DISCRETE COSINE TRANSFORM (DCT)

The DCT is a block transform; this means that it is used for input data which is divided into blocks of equal size. In a block transform each block is encoded separately but very frequently information from adjacent blocks is also used. The key advantage of using block transforms is the reduced computation delay and memory requirements. The DCT gets its name from the fact that the rows of the image transform matrix are obtained as a function of cosines.

V. ENTROPY CODING

The type of entropy coding used in this project is Arithmetic coding which is loss-less data compression. The technique governs the concept of assigning short bits to frequent characters and longer bits to rare characters. Advantages of Arithmetic coding, replace the entire input sequence with a single floating-point number, does not require the probability distribution, easily adaptive to code changes, no need to keep and send codeword table and fractional codeword length [1] Lecture 5.

VI. ANALYSIS OF RESULTS

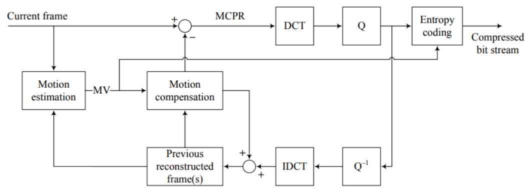

Fig. 1. Block-based video coder

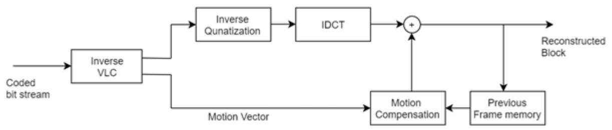

Fig. 2. Block-based video decoder

During the encoding process, individual frames are extracted from the video files and are fed into the block based video coder in Fig. 1. Two frames, a ‘previous reconstructed’ and ‘current’ frames undergo motion estimation process where motion vectors are calculated. During the first iteration, since no reconstructed frames have been created yet, the first and second frames of the video file are used. The motion vectors are then provided to the entropy coding to be compressed into bit files and to the motion compensation to form predicted frames and from where Motion Compensated Prediction Residual (MCPR) is generated. Finally, the MCPR frame is transformed using DCT, quantized and encoded using Arithmetic encoder. The quantized MCPR frame is dequantized and Inverse Discrete Cosine Transform (IDCT) is performed on it and added to the predicted frame to form a ’previous reconstructed frame’ which is used in the next iteration of the process.

Coded bit stream is decoded using arithmetic decoder in which it separates the frame bits from motion vector bits to complete the process of decoding. Then frame bits undergo inverse quantization and IDCT while motion vector bits are processed through motion compensation. Finally, the sum of both results creates the reconstructed block in which it is also used as previous frame for subsequent frames.

For this investigation two 360-degree videos ‘Natalie.mp4’ and ‘Kristine.mp4’ were used to compare the encoding and decoding process. The block searching algorithms tested during this investigation include the Exhaustive Search and Diamond Search. The block matching cost operations used are MAD and SAD and the size of the search range will vary between +/- 16 and +/- 32 pixels. The original video samples were cut into 4 seconds video segments, since the frame rate for both videos was the same, 30 frames per second, this meant encoding and decoding 120 frames.

From Fig. 3, the operational rate-distortion curve using a search range of +/- 32 pixels and Exhaustive - SAD block search algorithms, is able to achieve the highest PSNR at all qualities as expected. Having double the search range in comparison with the other methods, the process is able to find a better match. Following closely, both SAD and MAD Exhaustive search methods resulted in very similar PSNR values through out the encoding process. Similarly, both Diamond methods are very closely related to one another, however resulting in significantly lower PSNR in all cases, as expected. Diamond search although being very close to Exhaustive PSNR values it was much less resource intensive.

The PSNR per frame plot in Fig. 4 for the video file ‘Kristine.mp4’ displays an almost periodic behaviour where the keyframes, also known as intra-frames are seen as the peaks in PSNR. This is due to the nature of the video file. Most of the frames in the video file ‘Kristine.mp4’ are very similar to each other, this makes the DS and ES find a very low PSNR as ‘best matching block’ and translate in dips in the plot. In contrast some frames are very different from the rest in the video file and result in relatively high PSNR ‘best matching blocks’, this consequently translates into peaks in the plot.

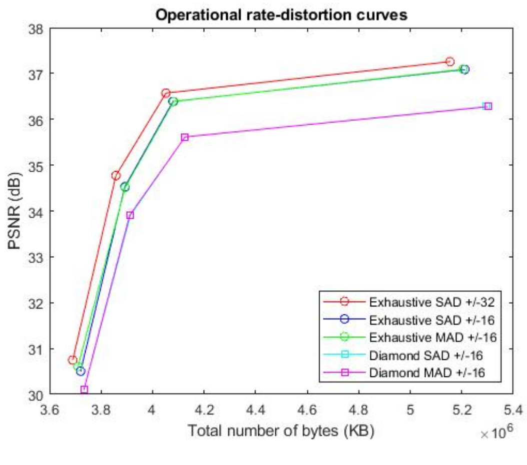

Fig. 3. Operational rate-distortion curves - Kristine

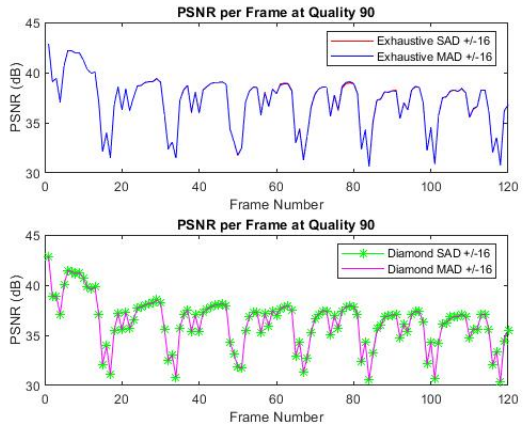

Fig. 4. PSNR per Frame at Quality 90 - Kristine

The operational rate-distortion curves for 'Natalie.mp4' are similar to the ones observed for 'Kristine.mp4' in Fig. 5 due to the same reason. ES is more effective at producing high PSNR frames than DS, however it is more computational expensive. We can clearly see this in the significantly lower PSNR from the DS at all qualities in Fig. 5 for this video file.

The plots in Fig. 6 are different from the ones produced for 'Kristine.mp4'. We observe no periodic behaviour, in contrast, the PSNR per frame for all cases is high at first but significantly decreases and remains constant for the last half for all frames in the video file. This indicates several differences between the 'current' and the 'previously reconstructed frames' early in the encoding process, but these differences decrease due to frames being more and more similar to each other in the second half of the video file.

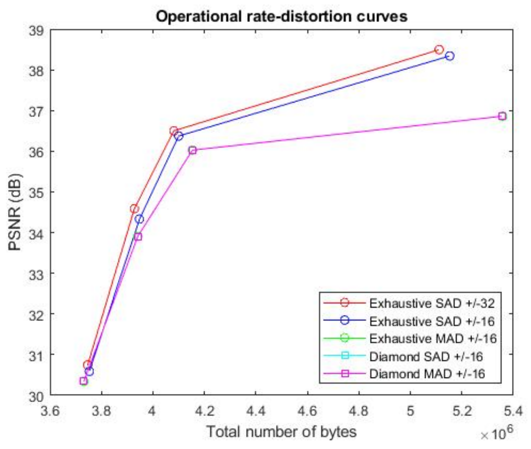

Fig. 5. Operational rate-distortion curves - Natalie

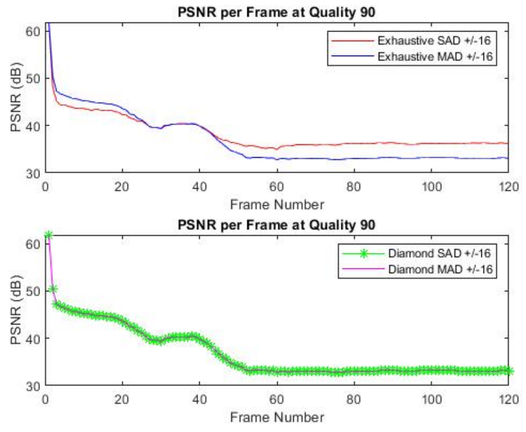

Fig. 6. PSNR per Frame at Quality 90 - Natalie

VII. Conclusion

The objective of our investigation was to fully encode and decode 360-degree videos using ES and DS using different search ranges and qualities values 30, 50, 70, and 90. The 360 video was successfully encoded and decoded using both algorithms. However, The results highlight the key advantage and disadvantages of each method. The ES produces high quality encoding which translates into high PSNR per frame, but at a great cost. This algorithm has a big delay and is very computational intensive and therefore renders it very inconvenient for high resolution videos. In contrast, DS is much faster at encoding producing a lower but close PSNR per frame values to the ones obtained from ES. For high resolution 360-degree videos, DS is definitely one of the best block searching algorithms. Reconstructed videos for all cases can be found in the same folder as this document.

References

[1] P. Muralidhar, C.B.Rama Rao. "Analysis of Block Matching Motion Estimation Algorithms", ICCCNT 2013, USA, July 4-6, 2013, Departments of Electronics and Communication Eng., National Institute of Technology. Warangal, July 5, 1987.

[2] A. Barjatya. "Block Matching Algorithms For Motion Estimation", DIP 6620 Final Project Paper in Digital Image Processing, Utah State University, pp. 16. April 2004.

[3] J. Y. Tham, S. Ranganath, M. Ranganath and A. A. Kassim, (Aug. 1998) *A Novel Unrestricted Center-Biased Diamond Search Algorithm For Block Motion Estimation,* IEEE Transactions on Circuits and Systems for Video Technology, Volume 8, pp. 369377.

[4] I. Bajic, Topic 5:Arithmetic Coding 1, ENSC 424 - Multimedia Communications Engineering 2018. pp. 1-21.