Region of Interest Based Frame Rate Up-Conversion Using Encoded Bit-stream

Kyungah Kim and Won Woo Ro
School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
E-mails: akyungah.kim@yonsei.ac.kr; bwro@yonsei.ac.kr (the corresponding author)

Abstract. This paper proposes an algorithm to find the region of interest (ROI) by using motion vector from the encoded bit-stream and perform frame rate up conversion (FRUC). First, the motion vector (MV) in the frame is collected by decoding the bit-stream and the ROI is defined in the frame based on the MV. The interpolated frame is created by reusing MV, which enables compact operations. The proposed method can provide a more comfortable visual experience to the user by performing FRUC using decoding information of the video bit-stream. The proposed technique effectively resolves the artifact and shows an excellent visual quality and execution speed compared to the block based motion compensated algorithms.

1. Introduction
Recently, the demand for ultra-high definition (UHD) and 3D video is rapidly increasing, and the needs for display technology for high quality multimedia are also increasing. Frame Rate Up-Conversion (FRUC), one of the important display technologies, has been proposed to visually smoothen the sequence by creating additional frames between the original ones. Thus, more dynamic motion effects can be realized due to interpolated frames.

Previously, most FRUC algorithms were developed based on MCI (Motion Compensated Interpolation) method [1][2][3][4]. They divide the frame into blocks and consider the Motion Vector (MV) of the blocks obtained by Motion Estimation (ME) [5]. An inserted frame is generated by referring to the interpolated MV and pixel values of an adjacent frame. Therefore, the image quality and execution performance of the motion compensated interpolation based FRUC depends largely on the performance of the ME. ME find a MV that minimizes the Sum of Absolute Differences (SAD) between the two blocks.

However, MCI-based technology generates blocking artifacts and creates holes and overlapped regions, because it moves the blocks on a pixel-by-pixel to produce an interpolated image. Choi et al. proposed Bidirectional Motion Estimation (BDME) which uses a Bidirectional Motion Vector (BMV) to solve hole and overlapped problems [6]. In addition, BDME performs additional spatio-temporal smoothing processing to mitigate blocking artifacts. However, the BDME method still represents a limitation of blocking artifacts and can hardly account for the actual motion of the object within the video sequence, resulting in poor visual quality of the interpolated frame.

In this paper, we propose an algorithm that finds the Region Of Interest (ROI) from the encoded bit-stream and interpolates the frame based on it. Through the proposed algorithm, it is possible to generate interpolated frames with better quality than the existing FRUC. First, the MV is extracted by decoding the bit-stream. The frame is divided into a 64×64 Coding Tree Unit (CTU) applied in the HEVC main profile, and the ROI is defined using the Square Sum of MV (SSMV) of the CTU. In ROI,
it is possible to efficiently generate interpolated image of ROI part by reusing MV defined in each CTU. The proposed FRUC efficiently solves the blocking artifacts, improves the image quality of 6.37 dB in PSNR, and improves the image quality of 13.24% in SSIM compared to existing algorithms.

The paper is organized as follows. In Section 2, we describe the backgrounds for the proposed technique. Then the extraction of the region of interest using motion vector is described in Section 3. In Section 4, the interpolation method using motion vector is explained. Performance evaluation is represented in Section 5. Finally, the paper is concluded in Section 6.

2. Backgrounds and Related Works

2.1 Coding Tree Unit of High Efficiency Video Coding (HEVC)

HEVC is released in April 2003 and the latest video compression standard [7][8]. Compared with the previous standard H.264/AVC, as the demand for high-resolution images such as HD and FHD increases, elements that can effectively encode them have been added. The biggest change is the definition of the encoding unit. In the previous H.264/AVC, 16×16 sized macro block is used for basic encoding unit [9]. However, as the resolution of the image has increased, basic encoding unit should be redefined [10]. Therefore, in HEVC, a quad-tree structured 64×64 sized CTU is proposed.

One CTU is basically 64×64 size and encoder recursively searches four depths up to 8×8 sized CU. First, a prediction mode indicating a maximum compression efficiency is determined for a 64×64 CU. The current best at 64×64 size is compared with the summation of sub-CU at lower depth as represented in Figure 1. That is, an operation that determines optimized prediction mode for each depth is called recursively and determines whether to divide it in comparison with the case where it is and sum of lower depths.

![Figure 1. Example of CTU in HEVC](image)

![Figure 2. Prediction Mode in HEVC](image)

2.2 Prediction Unit of HEVC

There are INTRA and INTER prediction mode. INTRA prediction mode search a spatial locality within the frame. It finds most similar data from the adjacent CUs. It uses only 2N×2N and N×N modes in Figure 2. INTRA prediction finds angles for luminance and chrominance signals to define data that is most similar to the current PU. INTER mode search temporal locality. It finds similar data from adjacent frames. The encoder finds the best mode among the 8 modes shown in Figure 2. In INTER prediction mode, the inter-frame data which is most similar to the current PU is defined through the MV obtained by motion estimation.

3. Extraction of ROI using Motion Vector

The proposed method analyses the MV in the bit-stream and defines the ROI in the frame. Since the MV described in the bit-stream is already searched by the HEVC encoder, additional searching operation such as motion estimation is not required. Therefore, it is much more efficient in terms of speed than existing MCI based technologies.
Figure 3 shows the CU decoding process. For each CU, skip_flag is decoded if it is not I-Slice. The skip_flag indicates whether the current CU is in the merge skip mode. If the CU is in the merge skip mode, it passes to the PU decoding without additional decoding process in the CU. If it is not the skip mode, it further decodes pred_mode_flag (inter or intra frame prediction) and part_moe (prediction modes in Figure 2) and then goes to decode in PU.

Figure 4 shows the PU decoding process. If the current PU is in skip or merge mode, it decodes merge_idx. Otherwise, it further decodes the motion information. At this time, the bidirectional or unidirectional MV is also decoded to collect information.

We also extracted the ROI to perform FRUC more efficiently. First, the frame is divided into 64×64 size in CTU unit, and square sum of MV (SSMV) is obtained in CTU unit. When the SSMV is large, there exists large number of MVs. And how interpolate such large movement portion affects the quality of the interpolated frame. SSMV is defined as shown in Figure 5. If SSMV exceeds the threshold, the corresponding CTU is regarded as ROI.

4. Interpolation with reused MV

This section describes the process of interpolation based on the extracted MV and ROI. Interpolation is performed by using the Equation 1.

\[
I(x,y) = Ref_{t-1} \left( x + \frac{1}{2} MV_{L,1}, i, y, y + \frac{1}{2} MV_{L,1}, j \right) + Ref_{t+1} \left( x + \frac{1}{2} MV_{L,0}, i, y, y + \frac{1}{2} MV_{L,0}, j \right)
\]  

(1)

We explain with the Figure 6. Ref refers to frames that are referenced when creating an interpolated frame. Assuming that an interpolated frame is a frame generated at time t, Ref_{t-1} and Ref_{t+1} are used for interpolation. Each reference frame has a bidirectional MV (based on the HEVC’s main profile, clean
random access configuration (CRA)). In this case, at the time t-1, only MV\(_{L1}\), which is an MV generated in the direction of the interpolated frame, is used. MV\(_{L0}\) is an MV generated through motion estimation of frame\((t-1)\) with frame\((t-2)\), which correlation with the interpolated frame is low. Similarly, at time t+1, only MV\(_{L0}\) is used for interpolation.

![Figure 6. Interpolation description](image)

The non-ROI part does not display important moves or matching object information. Therefore, the proposed method interpolates the background area using the BDME method, which shows good visual quality when the size of the motion vector is small. In addition, blocking artifacts, which are also problems with BDME, can be minimized.

5. Experimental Results

In this section, the experiments of the proposed technique are performed. We evaluated the proposed technique using SSIM and PSNR. We performed the FRUC by a factor of 2 and adopted BDME and DME as the baseline for comparison [6][11].

Figure 7, Figure 8, and Figure 9 show the evaluation results of PSNR and SSIM. The **Kimono** sequence has a single object and the static background. Compared with the BDME, the proposed technique obtained 6.37 dB PSNR image quality improvement and 3.99% of SSIM improvement. The proposed method also improves compared with DME PSNR by up to 1.99 dB and 1.20% of SSIM.

![Figure 7. PSNR and SSIM performance of Kimono sequence](image)

![Figure 8. PSNR and SSIM performance of BasketballPass sequence](image)
The sequence of BasketballPass has multiple objects with complex motion which move towards different directions. The proposed technique achieves the image quality improvements as 4.44 dB in PSNR and 13.24% in SSIM compared to BDME. Compared with DME, average of 2.21 dB and 5.51% in SSIM is improved.

The BasketballDrive sequence has multiple objects with complex motion and the resolution is higher than BasketballPass sequence. The proposed technique improves 3.07 dB in PSNR and 7.48% in SSIM of compared to BDME and 1.31 dB in PSNR and 3.40% in SSIM compared to DME.

Table 1 shows the execution time to interpolate one frame. All of the experiments are conducted on a hexa-core system which operates frequency of 3.20 GHz and 24 GB main memory. The proposed technique runs faster than BDME and DME. Proposed method extracts MV from bit-stream and reuses it, so it takes much less time because MV is skipped. Therefore, the proposed method performs outstanding visual quality and also shows improved performance as respects of execution time.

Table 1. Execution time (sec/frame)

|                | Kimono | BasketballPass | BasketballDrive |
|----------------|--------|----------------|-----------------|
| Proposed       | 0.78   | 0.05           | 0.89            |
| BDME           | 1.17   | 0.07           | 1.17            |
| DME            | 25.10  | 1.06           | 25.04           |

6. Conclusion

In this paper, we proposed the interpolation algorithm for frame rate up-conversion using motion vector which is described in encoded bit-stream. With the proposed technique, we resolved blocking artifact problems of FRUC and outperforms as respects of speedup. Our technique was evaluated with both PSNR and SSIM for image quality assessment and achieved improvements of 6.37 dB in PSNR and 13.24% in SSIM compared with the BDME algorithm.

Acknowledgments

This research was partially supported by the Graduate School of YONSEI University Research Scholarship Grants in 2018 and Multi-Ministry Collaborative R&D Program (R&D program for complex cognitive technology) through the National Research Foundation of Korea (NRF) funded by MSIT, MOTIE, KNPA (2018M3E3A1057248) and supported by the MOTIE (Ministry of Trade, Industry & Energy) (No. 10080590, Technology Development of Unified Memory System for Heterogeneous System Architecture) and KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device.

Reference

[1] Chen, Yen-Kuang, et al. “Frame-rate up-conversion using transmitted true motion vectors.” Multimedia Signal Processing, 1998 IEEE Second Workshop on. IEEE, 1998.
[2] Wang, Demin, Liang Zhang, and André Vincent. “Motion-compensated frame rate up-conversion—Part I: Fast multi-frame motion estimation.” IEEE Transactions on Broadcasting 56.2 (2010): 133-141.

[3] Kim, Kyungah, et al. “True motion compensation with feature detection for frame rate up-conversion.” Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015.

[4] Huang, Chang-Lin, and Teh-Tzong Chao. “Motion-compensated interpolation for scan rate up-conversion [also Erratum 35 (5)(May1996)].” Optical Engineering 35.1 (1996): 166-177.

[5] Dane, Gökçe, and Truong Q. Nguyen. “Motion vector processing for frame rate up conversion.” Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP’04). IEEE International Conference on. Vol. 3. IEEE, 2004.

[6] Choi, Byung-Tae, Sung-Hee Lee, and Sung-Jea Ko. “New frame rate up-conversion using bi-directional motion estimation.” IEEE Transactions on Consumer Electronics 46.3 (2000): 603-609.

[7] Maugey, Thomas, et al. “Evaluation of side information effectiveness in distributed video coding.” IEEE Transactions on Circuits and Systems for Video Technology 23.12 (2013): 2116-2126.

[8] Sullivan, Gary J., et al. “Overview of the high efficiency video coding(HEVC) standard.” IEEE Transactions on circuits and systems for video technology 22.12 (2012): 1649-1668.

[9] Wiegand, Thomas, et al. “Overview of the H. 264/AVC video coding standard.” IEEE Transactions on circuits and systems for video technology 13.7 (2003): 560-576.

[10] Ohm, J-R., et al. “Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC).” IEEE Transactions on circuits and systems for video technology 22.12 (2012): 1669-1684.

[11] Kang, Suk-Ju, Sungjoo Yoo, and Young Hwan Kim. “Dual motion estimation for frame rate up-conversion.” IEEE Transactions on Circuits and Systems for Video Technology 20.12 (2010): 1909-1914.