Published in IEE ELECTRONICS LETTERS, Feb. 2003

Efficient Video Coding with Fractional Resolution Sprite Prediction Technique

Yan Lu, Wen Gao and Feng Wu

An efficient algorithm for dynamic sprite-based video coding with fractional resolution motion compensations is presented. Different from the traditional sprite coding, the global motion and local motion are jointly compensated at two stages. The new techniques developed in JVT codec are also utilized. Experimental results demonstrate that the proposed algorithm can averagely save 8% bitrate compared to JVT codec for the typical test sequences.

I. INTRODUCTION

Sprite coding is a promising technique for highly efficient video coding. A sprite is an image composed of pixels visible throughout the video sequence. Thus, it can be effectively used for the predictive video coding. The technique, known as dynamic sprite coding, has been developed in the literature [1]. For dynamic sprite coding, the sprite for providing prediction is generated with the previously decoded frames both in the encoder and the decoder. In this way, the macroblocks (MBs) in the current frame can be coded with either local motion compensation (LMC) or sprite prediction. The advantages of sprite prediction are twofold. First, the sprite prediction can compensate the global motion such as panning, tilting and zooming, and second, the bits for coding the motion vectors can be saved. However, the advantages of sprite prediction have not been fully utilized in the traditional sprite coding techniques. In these methods, sprite prediction of an MB is directly from the corresponding block at the same position in the warped sprite. In this way, the bits for coding motion vectors (MVs) can be saved. However, the sprite cannot always provide very accurate predictions due to the small distortion of global motion estimation (GME). Thus, the coding efficiency improved by sprite prediction is not fully exploited since only a limited number of MBs are coded with sprite prediction.

To tackle the above problem, in this Letter we propose a two-stage motion compensation scheme for dynamic sprite coding. At the first stage, the sprite is warped towards the current frame using the estimated global motion parameters. At the second stage, the warped sprite as well as the previously decoded frame serves as the reference frame of local motion compensation. Different from the traditional LMC, The local motion between the current frame and the warped sprite is only searched within a very small range. In this way, the sprite can provide more accurate prediction, and meanwhile the bits for coding the MVs are still very few. Rate-distortion optimisation (RDO) is applied to select the reference frame and coding mode.

To further improve the efficiency of sprite coding, in this Letter we also adopt some new techniques developed in JVT codec [2]. The JVT codec intend to be a new video coding standard named as H.264 in ITU and Part 10 of MPEG-4 in ISO. The baseline of the proposed sprite coding is derived from JVT codec. Since JVT codec supports fractional resolution motion compensations, the sprite is also stored and updated at the fractional resolution. An interpolation algorithm is developed for this purpose by improving the algorithm in JVT codec.

II. Proposed Sprite Coding

The proposed scheme aims at efficiently compressing the video by fully exploiting the correlations among multiple frames. An improved dynamic sprite coding algorithm is employed for this purpose. Fig. 1 shows the proposed sprite coding architecture. Since JVT codec supports multiple frame motion compensation, the baseline of the proposed scheme is based on it. Some modules, such as Intra prediction, integer transform, quantisation and entropy coding, are the same as used in JVT codec. However, to compensate global motion such as panning, tilting and zooming, a sprite image is dynamically generated to serve as the alternative reference of motion compensation. In this Letter the six-parameter affine motion model is used to represent the global motion.

As shown in Fig. 1, the video can be coded with either Intra mode or Inter mode. Switch S_0 controls the Intra/Inter mode selection by employing the rate-distortion optimisation scheme. For Intra mode coding, directional spatial prediction is utilised. For Inter mode coding, the prediction can be from either the previously decoded frame or the sprite warped towards the current frame using the estimated affine motion parameters. The motion trajectories of three vertex points are instead of the affine motion parameters encoded into the bitstream. The warped sprite is stored at a high resolution, and hence it can be directly used for the LMC without performing the fractional sample interpolation. For sprite prediction, the local motion is only searched within a small range to speed up the entire encoding process. Similar to the

Intra/Inter mode selection, Switch S_1 controls the prediction frame selection by employing the rate-distortion optimisation scheme.

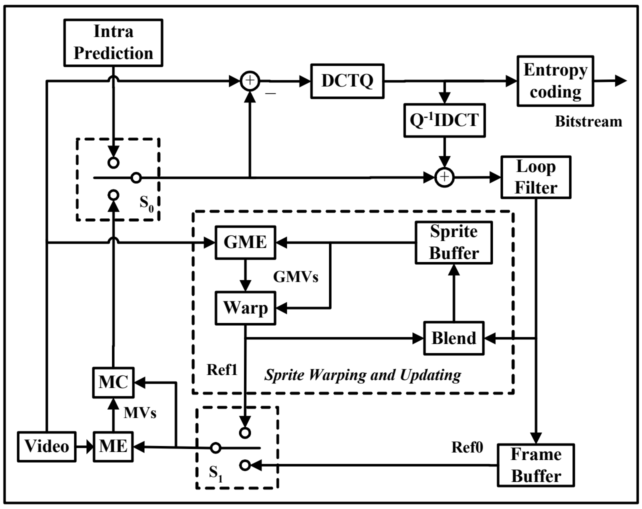
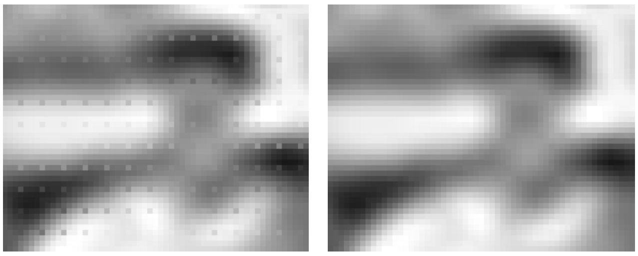


Fig. 1 Block diagram of the proposed sprite-based video coding scheme.

After one frame is coded, the sprite is updated with the new decoded frame. The decoded frame is first interpolated with an improved algorithm developed from JVT codec. This Letter adopts quarter sample interpolation as an example. Samples at integer positions are available at the beginning of interpolation. Samples at half positions are first obtained by applying a six-tap filter, same as that in JVT codec. Afterwards, quarter samples are obtained by averaging with rounding the four nearest samples at half or integer positions. In JVT codec, the bottom-right sample at quarter position is obtained by averaging the four nearest samples at integer positions. In this way, the bottom-right sample presents noisy in viewing the entire interpolated image, as shown in Fig. 2*a*. Since the bottom-right samples are together used for prediction in JVT codec, they have little effects on coding efficiency. However, after being warped in the proposed scheme, the positions of these samples become irregular, which may affect the accuracy of sprite prediction. As shown in the image of Fig. 2*b*, the noisy samples produced by JVT interpolation have been eliminated. The decoded frame after quarter sample interpolation is averagely blended with the sprite in the buffer. The sprite buffer is empty at the beginning, and then updated frame by frame.



(a) (b)

Fig. 2 Enhanced Images interpolated with (a) JVT codec and (b) the proposed algorithm.

III. Experimental Results

To evaluate the performance of the proposed algorithm, we performed the experiments on some MPEG test sequences. Since JVT codec outperforms other video coding standards, the comparisons are between the proposed codec and JVT codec. JM3.9 was adopted as the reference software. Some common test conditions were the same. For instance, I frame was only used once and followed with all P frames; motion vector resolution was quarter pixel; rate-distortion optimisation was switched on; and CABAC was used for entropy coding. To achieve the typical results, we used five and two reference frames in JM3.9, respectively. In the proposed algorithm, only one reference frame as well as the sprite in the buffer provides interframe prediction.

Fig. 3 shows the PSNR-Bitrate curves achieved from the testing on Stefan sequence in CIF format at 15 fps. JM-Ref*n* indicates that *n* reference frames are used in JM3.9. The experiments demonstrate that the proposed codec averagely saves 8% bit rate compared to JM3.9 with the best settings. Complexity comparison between JM3.9 with five reference frames and the proposed algorithm is also analysed. The sprite buffer can be restricted within the constant size (e.g. 2.25 times the frame size). Therefore the memory cost in the proposed sprite coding is less than that in JM3.9. For computing complexity analysis, we only consider the computing time of motion estimation. In JM3.9, the local motion estimation (LME) is performed five times (i.e. once for each reference frame). In the proposed codec, LME is performed only once. The total time of motion estimation in the proposed codec is for one LME and one GME. By utilizing the fast GME algorithm, the total time for LME plus GME is less than five times LME in JM3.9. Similar to the traditional dynamic sprite coding techniques, the proposed codec has the disadvantage that sprite warping has to been performed in the decoder. However, considering the significant improvement of coding efficiency, the extra computing complexity for sprite warping is acceptable.

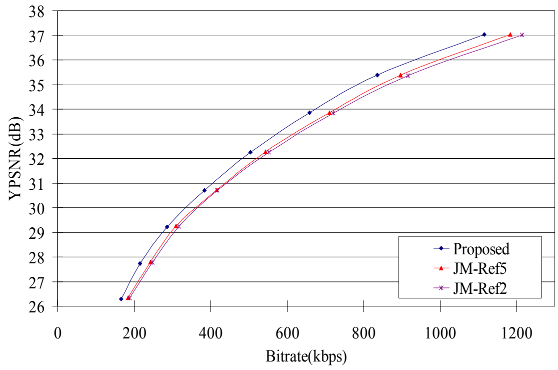


Fig. 3 PSNR-Bitrate curves achieved from the testing on Stefan sequence.

IV. Conclusions

This paper has presented a highly efficient algorithm for dynamic sprite coding. The high coding efficiency is achieved due to two reasons. First, the new techniques developed in JVT codec are utilised; and secondly, the fractional resolution sprite prediction is incorporated into the proposed algorithm.

References

1. Smolic, A., Sikora, T. and Ohm, J-R.: 'Long-term global motion estimation and its application for sprite coding, content description, and segmentation', IEEE Trans. On Circuits Syst. Video Technol., Dec. 1999, vol. 9, pp. 1227-1242

2. Joint Video Team of ISO/IEC MPEG and ITU-T VCEG: 'Joint committee draft (CD)', May 2002, Fairfax, USA

Authors' affiliations:

Yan Lu (Department of Computer Science, Harbin Institute of Technology, Harbin, 150001, China. Email: ylu@jdl.ac.cn or ylu@ieee.org)

Wen Gao (Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China)

Feng Wu (Microsoft Research Asia, Beijing, 100080, China)