Adaptive Motion Estimation Order for Frame Rate Up-Conversion

Chengzhou Tang, Ronggang Wang, Wenmin Wang
School of Electronic and Computer Engineering
Peking University Shenzhen Graduate School
Shenzhen, China
Email: cztang@sz.pku.edu.cn, rgwang@pkusz.edu.cn, wangwm@pkusz.edu.cn

Abstract— **This paper proposes an adaptive motion estimation (ME) order for frame rate up-conversion (FRUC). Almost all existing FRUC methods adopt a raster scan order for ME. The ME is performed from top-left blocks to bottom-right blocks in raster scan order. Such an order can propagate some wrongly estimated motion vectors (MV) through a frame. The proposed method first detects the blocks rich in features (feature blocks) and estimates their MVs. Then ME is performed on the other blocks according to their distance to feature blocks. The closer a block to feature blocks is; the earlier the ME is performed on it. In this adaptive order, MVs of feature blocks are propagated to its neighbors. It makes the estimated motions of a frame close to the true motions. In order to demonstrate the efficiency of the proposed method, we estimate the MVs with diamond search in the proposed adaptive ME order. In the experiments, the quality of frame rate up converted videos have been significantly improved compared with the ones using traditional raster scan order. Moreover, the adaptive ME order can be easily combined with various ME methods applied in previous FRUC.**

I. Introduction

Frame rate up-conversion (FRUC) is the technique that generates videos with high frame rate from the ones with low frame rate. The simplest solution is to add a new frame by duplicating its previous frame or averaging its previous and successive frames. However, these methods usually lead to unacceptable artifacts in the video sequences with complex motions.

In order to improve visual experience for videos with high motions, various motion compensated interpolation (MCI) based methods [1-4] have been proposed for FRUC. In these methods, the added frames are interpolated according to the motion vectors (MV) between its neighbor frames. So accurate MVs are critical to achieve high-quality frame interpolation. Then block based motion estimation (ME) is adopted in various FRUC methods. One of the most classical ME methods for FRUC is 3DRS [5]. In 3DRS, spatial-temporal candidates are used to initialize a block's motion vector. Apart from 3DRS, other ME methods for video coding such as DS [6], HS [7], UMHS [8] and EPZS [9] can also be used for FRUC. All of these ME methods are designed for single block. They are always performed in raster scan order, where blocks within a frame are scanned from top-left to bottom-right.

Other works like [10] and [11] take feature correspondences into account. They track feature points between frames and use the points' trajectories as initial MVs. In these feature concerned methods, blocks' motions are still estimated in raster scan order.

The traditional ME methods for FRUC improve the accuracy of MVs either by a robust matching function, better motion search strategy or better motion initialization. In these methods, the ME is performed block by block in raster scan order, and the MV of current block is utilized as a candidate MV for its not estimated neighbors. The true MV of textureless block is hard to get, and once the MV of a block is wrong, it tends to be propagated to the not estimated blocks in this frame.

In order to alleviate the above drawbacks of traditional ME, this paper proposes an adaptive order for ME other than raster scan order. This adaptive order is constructed as follows: First, ME is performed on the blocks rich in features (feature blocks). Secondly, the ME order of the other blocks is constructed according to their distance to feature blocks. During the ME phase, the closer a block to feature blocks is, the earlier the ME is performed on it. Experimental results testified that the quality of interpolated frames is significantly improved by performing ME in the proposed adaptive order.

The rest of the paper is organized in three sections. In Section II, we will describe motion estimation process in the proposed adaptive order detailedly. In Section III, the experimental results are given to prove the enhancement generated from the proposed method. At last, we give a brief conclusion in Section IV.

II. Motion Estimation in Adaptive Order

In previous FRUC algorithms, ME is performed over blocks in traditional raster scan order. As shown in Fig.1, spatial-temporal candidates are used to initialize current block's MV. The candidate minimizing motion cost (SAD or SSD) value is assigned to the current block as initial MV. The final MV can be obtained by performing ME in a search range. In such order, MVs are propagated from top-left blocks to the ones bottom-right.

For the blocks with blank textures, their MVs tend not to be close to the true motions. It is caused by the limitation of motion cost measurement for textureless regions. If we perform ME in raster scan order, some wrongly estimated MVs will be propagated to the not estimated blocks and reduce the visual quality of interpolated frames.

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

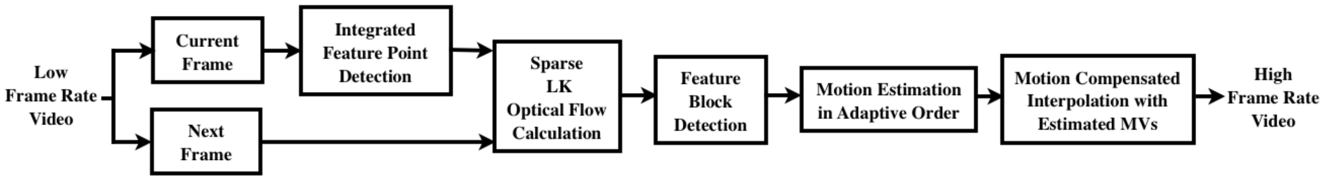
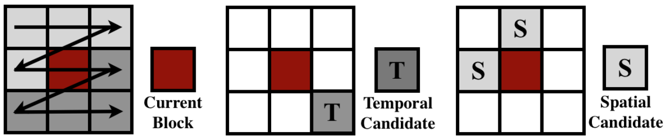


Fig. 2. Flow diagram of proposed adaptive ME order construction method



(a) (b) (c)

Fig. 1. Traditional raster scan order and spatial-temporal candidates. In raster scan order (a), top-left blocks are estimated ahead of the ones bottom-right. In (b), MVs of current block 's right-down neighbor is taken as temporal candidate, it has been estimated in previous frame. In (c), MVs of left and up neighbors are taken as spatial candidates, they are estimated in current frame.

In order to alleviate this drawback, we construct an adaptive ME order based on feature blocks. The flow diagram of the proposed method is shown in Fig.2. In this method, feature points are first detected using a fast integrated feature point detector. After that, the trajectories between frames of the feature points are tracked as sparse Lucas-Kanade optical flow [12]. Then feature blocks are detected and initialized with the feature points and feature trajectories. At last, a FIFO queue is implemented. The blocks closer to feature blocks are inserted into the queue ahead of the farther ones. The head block of the FIFO queue is estimated and popped out one by one. The ME will be terminated until the queue is empty.

Feature blocks are crucial in this adaptive order construction. Their MVs are close to true motions. It benefits from their distinguishable textures. On the other hand, neighbor blocks of feature blocks tend to have similar motions. Then the blocks closer to feature blocks should be estimated ahead of the farther ones. Thus, MVs are propagated from blocks closer to feature blocks to the farther ones. Performing ME in such adaptive order will not only increase the accuracy of MVs but also smooth the motion field.

A. Integrated feature point detection

Since local regions around feature points are usually rich in textures, feature point detection is widely used for image/video processing. Two of the most famous detectors are FAST detector [13] and Harris detector [14]. The FAST detector is fast because it compares pixel intensity through a binary decision tree. The Harris detector computes a matrix of second-order image derivatives, it can filter out the points along edges. In order to combine these different characteristics of the two detectors, they are integrated to get better responses as shown in Fig.3. After feature point detection, feature points' trajectories are tracked as sparse Lucas-Kanade optical flow.

(a) (b) (c)

Fig. 3. A sample of integrated corner points detection. (a) is a frame from sequence Foreman, (b) is the feature points detected by FAST detector, (c) is the final feature points after filtered out by Harris detector. All the feature points are drawn as circles with diameter of 9 pixels.

B. Feature block detection

As feature points' locations are arbitrary and in subpixel accuracy, they should be converted to block based form for ME. So we detect and initialize feature blocks using feature points and feature trajectories. The 9×9 image patch around a feature point is defined as a feature patch, and a frame is divided into 8×8 blocks. As shown in Fig.4, a block may be covered with multiple feature patches. We calculate the weight of a feature patch P based on its overlapped area $S_{overlap}(B,P)$ with block B as follows:

$$w(B,P) = \begin{cases} S_{overlap}(B,P) : S_{overlap}(B,P) > \alpha S(B) \\ 0 : otherwise \end{cases} \tag{1}$$

In Eq.1, α is a threshold ratio and $S(B)$ is the area of B. Then we sum the weight $w(B,P)$ of every feature patch that covers B, and detect B as a feature block if this sum is larger than 0. The MV of a feature block $B_{feature}$ is initialized by Eq.2. In Eq.2, $T(P_i)$ is the optical flow of the feature point centering at patch P_i, and P_i is one of the n patches that cover $B_{feature}$.

$$MV_{initial}(B_{feature}) = \frac{\sum_{i=1}^{n} w(B_{feature},P_i)T(P_i)}{\sum_{i=1}^{n} w(B_{feature},P_i)} \tag{2}$$

In some cases, $T(P_i)$ has low confidence. If the corresponding SAD value is larger than a threshold, $T(P)$ is recalculated as follows:

$$T_{initial}(P) = \frac{\sum_{i=1}^{4} w(P,B_i)MV_{previous}(B_i)}{\sum_{i=1}^{4} w(P,B_i)} \tag{3}$$

In Eq.3, $MV_{previous}(B_i)$ is the MV of B_i estimated in previous frame, and B_i is one of the four blocks that overlap with patch P. After obtaining $T_{initial}(P)$, we further estimate $T(P)$ as:

$$T(P) = T_{initial}(P) + T_{me}(P) \tag{4}$$

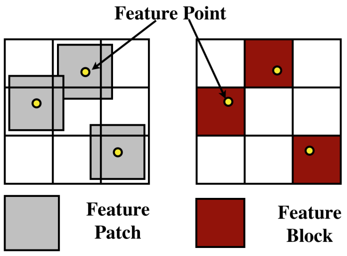
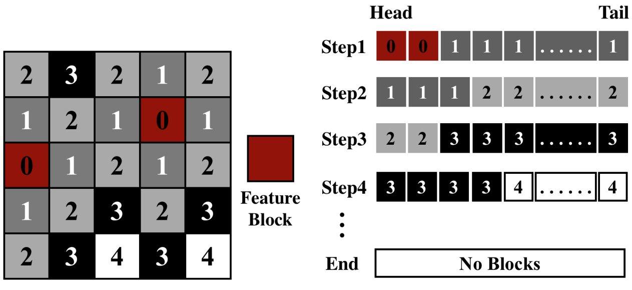


Fig. 4. Overlap between feature patches and feature blocks.



(a) (b)

Fig. 5. A sample of blocks' distance from feature blocks (a) and blocks' arrangement in the FIFO queue (b).

where the added trajectory $T_{me}(P)$ is from ME operation on feature patch P.

C. Motion estimation in adaptive order

After initializing all the feature blocks, ME is performed on them at first. The ME order for the other (non-feature) blocks is decided by their distance to nearest feature block shown in Fig.5 (a). The closer a block to its nearest feature block is, the earlier the ME is performed on it. For the blocks with the same distance, they are estimated in raster scan order.

Once the MV of a block has been estimated, it is used as additional candidate for its neighbor blocks. With the method described above, well estimated MVs are propagated from feature blocks to non-feature blocks in an adaptive order. In order to arrange the blocks, a FIFO queue is adopted, which is shown in Fig.5 (b). For a better understanding of our method, the motion estimation process in the proposed adaptive order is given in Algorithm 1.

Algorithm 1: Motion Estimation in Adaptive Order

Input: Set of feature blocks $S_{feature}$ and Set of all blocks S_{all}
Output: Motion vectors of blocks MV

Initialize a FIFO queue Q;
foreach *block* b *in the set* S_{all} **do**
  Set $D(b)$ to MAX;
end
foreach *feature block* f *in the set* $S_{feature}$ **do**
  Set $D(f)$ to 0;
  Set $MV_{initial}(f)$ using Eq.2;
  Insert f into Q;
end
while Q *is not empty* **do**
  Pop the head block q from Q;
  Estimate $MV(q)$ using optional ME method around $MV_{initial}(q)$;
  foreach *neighbor block* n *of* q **do**
    if $D(n) > D(q) + 1$ **then**
      Set $D(n)$ to $D(q) + 1$;
      Add $MV(q)$ as additional candidate;
      Add MVs of n's not estimated neighbors as temporal candidates;
      Select the candidate that minimize SAD as $MV_{initial}(n)$;
      Insert n into Q;
    end
  end
end

III. Experimental Results

In order to evaluate the performance of the proposed method, the odd frames of the test sequences are dropped. Then these frames are interpolated by MCI with estimated MVs. The MVs are estimated by diamond search in proposed adaptive order and diamond search in traditional raster scan order respectively. We compare the quality of interpolated frames by the above two methods to testify the effectiveness of our proposed adaptive ME order.

To measure the quality of interpolated frames, PSNR and SSIM [15] value of luminance (Y) component between the interpolated and original odd frames are calculated. PSNR value is the most common quality metrics to evaluate similarity between frames. However, it is not always consistent with subjective quality. So SSIM is developed to generate objective values, which are more consistent with human visual sense. In the experiments, frame interpolation is accomplished using raw MVs without any post-refinement. Then influences from other factors can be prevented.

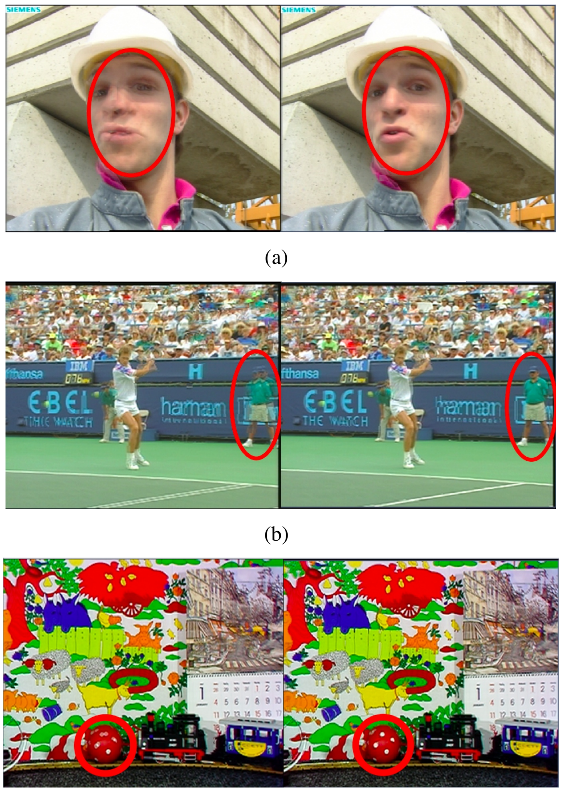
The experiment is done on several typical CIF sequences, including Foreman, Coastguard, Stefan, Flower and Mobile. Shared parameters such as block size and search range are all set the same for FRUC with traditional raster scan order and FRUC with proposed adaptive order. It guarantees that the different order is the only factor that makes differences to performance. The performance comparison between the two methods is summarized in Table I, from which we can see that the PSNR of interpolated frames is improved by 1.87dB in average over the 5 sequences by the proposed adaptive order. Fig.6 shows frame by frame comparison from three test sequences. We also give the comparison in SSIM scores in Table II, where SSIM scores are increased by 0.0468 in average.

TABLE I
QUALITY COMPARISON ON PSNR

Sequence	Traditional Order	Adaptive Order	Improvement
Forman	30.58	32.97	2.39
Coastguard	31.45	33.20	1.75
Stefan	25.93	28.12	2.19
Flower	28.17	29.80	1.63
Mobile	28.10	29.50	1.40
Average	28.85	30.72	1.87

TABLE II
QUALITY COMPARISON ON SSIM

Sequence	Traditional Order	Adaptive Order	Improvement
Forman	0.8665	0.9265	0.0600
Coastguard	0.8851	0.9341	0.0490
Stefan	0.8617	0.9270	0.0653
Flower	0.9143	0.9572	0.0429
Mobile	0.9427	0.9599	0.0172
Average	0.8941	0.9409	0.0468



(a)

(b)

(c)

Fig. 6. All the right images are the results from the proposed method while the left images are from FRUC with ME in raster scan order. (a) is from sequence Foreman, (b) is from Stefan and (c) is from Mobile.

IV. CONCLUSION

In this paper, a novel adaptive ME order based on feature information has been proposed to improve existing ME methods for FRUC. The main contribution is to estimate the blocks that tend to provide accurate MVs at first and propagate these MVs to their neighbors. As a consequence, MCI artifacts caused by wrong MVs from blank texture blocks are alleviated. Experimental results on FRUC have proven the effectiveness of the proposed adaptive order. Although the implementation in this paper adopts diamond search as its ME method, the proposed adaptive order can be easily combined with other ME algorithms for FRUC. In the future, we will concentrate on improving the adaptive order construction with more appropriate criterion besides blocks' distance to nearest feature block.

REFERENCES

[1] T.-Y. Kuo, J. Kim, and C.-C. J. Kuo, *Motion-compensated frame interpolation scheme for h.263 codec*, Proc. IEEE International Symposium on Circuits and Systems, pp. 491-494, 1999.

[2] J. Zhai, K. Yu, J. Li, and S. Li, *A low complexity motion-compensated frame interpolation method*, Proc. IEEE International Symposium on Circuits and Systems, pp. 4927-4930, 2005.

[3] B.-D. Choi, S.-H. Lee, and S.-J. Ko, *New Frame Rate Up-conversion Using Bi-directional Motion Estimation*, IEEE Trans. on Consumer Electronic, vol. 46, pp. 603-609, Aug. 2000.

[4] T. Ha, S. Lee, and J. Kim, *Motion compensated frame interpolation by new block-based motion estimation algorithm*, IEEE Trans. on Consumer Electronic, vol.50, pp. 752-759, May 2004.

[5] G. Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, *True motion estimation with 3-D recursive search block matching*, IEEE Trans on Circuit System and Video Technology, vol. 3, pp. 368-379, Oct. 1993.

[6] S. Zhu and K.-K.Ma, *A new diamond search algorithm for fast block matching motion estimation*, IEEE Trans on Image Processing, vol. 9, pp. 287-290, Feb. 2000.

[7] C. Zhu, X. Lin, and L. P. Chau, *Hexagon-based search pattern for fast block motion estimation*, IEEE Trans on Circuits System and Video Technology, vol. 12, pp. 349-355, May 2002.

[8] Cheng. Z, Peng. Z, He. Y, *Faster integer Pel and fractional Pel motion estimation for H. 264/AVC*, Journal of Visual Communication and Image Representation, vol. 17, pp. 264-290, Apr. 2002.

[9] A. M. Tourapis, *Enhanced predictive zonal search for single and multiple frame motion estimation*, Proc. of the SPIE Conference on Visual Communications and Image Processing, pp. 1069-1079, 2002.

[10] Y. H. Cho, H. Y. Lee, D. S. Park and C. Y. Kim, *Enhancement for temporal resolution of video based on multi-frame feature trajectory and occlusion compensation*, Proc. of the IEEE International Conference on Image Processing, pp. 389-392, 2009.

[11] R. Braspenning and G. de Haan, *True-motion estimation using feature correspondences*, Proc. of the SPIE Conference on Visual Communications and Image Processing, pp. 396-407, 2004.

[12] B. D. Lucas and T. Kanade, *An iterative image registration technique with an application in stereo vision*, Proc. of the International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.

[13] E. Rosten and T. Drummond, *Machine learning for high-speed corner detection*, Proc. of the European Conference on Computer Vision, pp. 430-443, 2006.

[14] C. Harris and M. Stephens, *A combined corner and edge detector*, Proc. of the Alvey Vision Conference, pp. 147-151, 1988.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Trans on Image Processing, vol. 13, pp. 600-612, Apr. 2004.