

Received August 25, 2018, accepted September 28, 2018, date of publication October 12, 2018, date of current version November 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875688

Hierarchical Motion Estimation for Small Objects in Frame-Rate Up-Conversion

NGUYEN VAN THANG¹, JIWOONG CHOI¹, JU-HYEON HONG¹,
JIN-SUNG KIM², AND HYUK-JAE LEE¹

¹Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea

²Department of Electronic Engineering, Sun Moon University, Asan 31460, South Korea

Corresponding author: Jin-Sung Kim (jinsungk@sunmoon.ac.kr)

This work was supported in part by LG Display, in part by the Ministry of Trade, Industry, and Energy under Grant 10080568, and in part by the Korea Semiconductor Research Consortium Support Program for the Development of the Future Semiconductor Device.

ABSTRACT Block-based hierarchical motion estimations are widely used for frame interpolation in frame-rate up-conversion and are successful in generating high-quality interpolations. However, it still fails in the motion estimation of small objects when a background region moves in a different direction. This is because the motion of small objects is neglected by the down-sampling and over-smoothing operations at the top level of image pyramids in the maximum a posteriori (MAP) method. Consequently, the motion vector of small objects cannot be detected at the bottom level, and therefore, the small objects often appear deformed in an interpolated frame. This paper proposes a novel algorithm that preserves the motion vector of the small objects by adding a secondary motion vector candidate that represents the movement of the small objects. This additional candidate always propagates from the top to the bottom layers of the image pyramid. Experimental results demonstrate that the intermediate frame interpolated by the proposed algorithm significantly improves the visual quality when compared with conventional MAP-based frame interpolation.

INDEX TERMS Frame interpolation, FRUC, MEMC, motion estimation, small object.

I. INTRODUCTION

Motion-compensated frame-rate up-conversion (MC-FRUC) is widely used for Liquid Crystal Display Televisions (LCD TVs) to increase the frame rate during video display [1], [2]. Hierarchical block-based motion estimation is widely used for MC-FRUC thanks to its relatively small complexity because a real-time operation is required for LCD TVs. The visual quality of a generated intermediate frame is heavily dependent on the accuracy of the motion vectors between two temporally consecutive original frames. Conventional block-based hierarchical motion estimation suffers from a fundamental limitation in handling motion details, such as the diverse movements of a small object in the background. In general, it is not easy to define a general size to classify an object into small one or not. Because a part of a large object in a block can also be defined as a small object when the motion vector of the small part is different from the motion vector of the block that includes the small part. The two primary reasons for the fundamental limitation in handling motion of small objects are image down-sampling and motion over-smoothing.

As down-sampling reduces the size of an object at the top pyramid level, it has little effect on the motion estimation for a block that includes the object. Therefore, the motion of a small object is often neglected in the motion estimation at the top level. The over-smoothing of a motion vector occurs in a conventional maximum a posteriori (MAP) [9], [11]–[16], [22], including the operation to increase the smoothness of a motion vector field. Over-smoothing occurs when a motion vector of a block is different from the motion vectors of neighboring blocks. MAP replaces the motion vector of a small object block with a background motion vector, which often results in the selection of a wrong motion vector of a block that includes the small object.

Fig. 1 shows an example that illustrates the limitation of the conventional block-based hierarchical motion estimation. In this example, an area of 16×16 pixels in an input image is used as the unit for motion estimation and the number of pyramid levels is three. The scaled image of each level is partitioned into 4×4 blocks, and each of these blocks is represented by a square. A small object is represented by a shaded area while the background is represented by

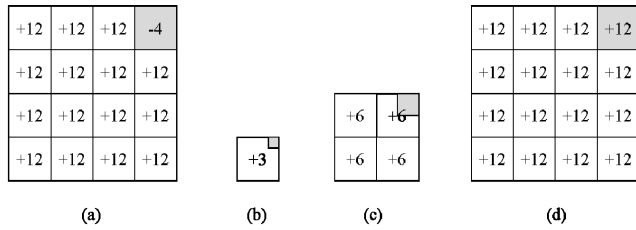


FIGURE 1. Example of an inaccurate motion vector of a small object that is represented by the shaded area in a hierarchal motion estimation: (a) the ground-truth motion vector at the bottom level. (b) a block with a motion vector at top level; (c) blocks with motion vectors at lower levels; (d) error of the estimated motion vector of a small object at the bottom level.

a white one. For simple illustration, the number in each block represents only the horizontal component of the estimated motion vector. In this example, a small object moves in a different direction from the direction of the movement of the background. The ground truth motion vectors are shown in Fig. 1 (a), in which a shaded block corresponds to a small object. In block-based hierarchical motion estimation, an input image is down-sampled and motion vectors are estimated from the top level. Fig. 1 (b) shows a top level block obtained by down-sampling the image area in Fig. 1 (a). The portion of the small object in the block at the top level is small; therefore, the motion of the small object has little effect on the motion estimation of the block. Thus, the estimated motion vector only represents the motion of the background whereas the motion of the foreground is ignored. As shown in Fig. 1 (c) and (d), the motion vector of the small object is not propagated from the higher level. Consequently, the motion vector of a small object is inaccurately determined as shown in Fig. 1 (d), which is different from the ground-truth motion vector shown in Fig. 1 (a). The wrong motion vector may cause a small object to appear distorted or to disappear in the interpolated frame.

Extensive research efforts have been made to handle the motion estimation for challenging cases in frame-rate up-conversion, from repetition pattern objects [17]–[19] to small objects [3]. A previous study proposes a SIFT feature-based optical flow in order to explore the motion vector of a small object [3]. The method proposed in this study successfully improves the accuracy but increases the computational complexity. For example, the time required to estimate the flow of an urban test sequence in the Middlebury test bench is 342 s. Another method employs variable block sizes at motion boundary blocks to provide a dense motion vector field [4]. This method succeeds in deriving accurate motion vectors at boundary blocks but it also requires extensive computation. In the method proposed in [5], a pixel-based motion vector selection is derived from neighboring block-based motion vectors. The motion vectors of the pixels are generated from the estimated motion vectors of the blocks that include them. The pixel-based estimation further improves the accuracy of motion estimation although small objects may remain undetected if the motion estimation for

a block is inaccurate. Recently, Jeong, Lee and Kim propose a use of video segmentation for estimating motion vectors of pixels [6]. The method can generate a dense motion vector field and successfully reduce block artifacts. However, its computational complexity is high owing to the derivation of video segmentation solving a graph cut algorithm. Variable block size approaches have been also studied in previous works [4], [6], [7]. The method proposed in [9], increases the density of a motion vector field in a hierarchical manner. In other words, the motion vector of a sub-block is derived from the motion vector of a parent block and those of the neighboring blocks of the parent block. This method successfully reduces the computational complexity while offering a reasonable level of accuracy. However, it cannot reduce the motion vector errors owing to the disappearance of the motion vector of small objects from the motion estimation of the original blocks.

This study addresses the difficulty in the motion estimation of a small object described in Fig. 1 and proposes a new hierarchical motion estimation algorithm for MC-FRUC with two primary contributions.

- The hidden motion information of a small object at the top level is represented by an *alternative motion vector* candidate. The alternative candidate is propagated to the lower levels and used for the motion estimation of small objects at the bottom layer.
- A matching algorithm for determining the alternative motion vector is proposed. If pixels with high residual costs are detected in a block, the matching algorithm is performed for the high cost pixels, otherwise it maintains the motion vector estimated by a full search block matching algorithm as the alternative motion vector.

The rest of the paper is organized as follows. Section II introduces previous works and the motivations of the paper. The proposed method is presented in Section III and experimental results are discussed in Section IV. Section V concludes this paper.

II. PREVIOUS WORKS

A. MAXIMUM A POSTERIOR FRAMEWORK

A recent proposal of MAP-based motion estimation has achieved better performance than the conventional block matching algorithm (BMA) because it exploits smoothness constraints of motion fields [9], [12], [13]. The smoothness constraint on neighboring motion vectors can improve the estimation accuracy thanks to the property that motion vectors in an object do not change abruptly. The smoothness constraint is a key contribution of many optical flow methods [20], [21], and block based motion estimation methods [9], [13], [22]. Horn and Schunck [20] propose an algorithm that uses a smoothness constraint as a penalty for pixel-matching scores in dense motion field estimation. Zach et al. [21] compute the smoothness term with an L1 norm of motion vector difference between neighboring ones. In recent block based motion estimation [9], [13], [22],

smoothness constraint is used as a key approach to find true motion vectors.

The BMA is an unconstrained optimization; meanwhile, MAP applies prior probability to the optimization in order to make a smooth change in motion vector field. In MAP, the objective function is to minimize an energy function that is composed of two components. The first term is a data cost that represents the block matching value or likelihood and the second one is a smoothness cost that encodes a prior probability of the motion vector field. The combination of two components, which are likelihood and prior probability, is to estimate the posterior probability of the motion vector field as shown in the following equations:

$$E(\mathbf{u}) = \text{SAD}(\mathbf{u}) + \lambda * \Sigma P_c(\mathbf{u}, \mathbf{v}_c) \quad \text{with } \mathbf{v}_c \in N_c \quad (1)$$

$$E(\mathbf{u}) = \sum_{x \in \text{Block}} |I_c(x) - I_r(x + \mathbf{u})| + \lambda * \sum_{\mathbf{v}_c \in N_c} \{ \|\mathbf{u} - \mathbf{v}_c\| * \theta(\mathbf{u}, \mathbf{v}_c) \} \quad (2)$$

where $\mathbf{u} = (u_x, u_y)$, is the motion vector variable, $\text{SAD}(\mathbf{u})$ is the sum of absolute difference of the block that corresponds to \mathbf{u} , $P_c(\mathbf{u}, \mathbf{v}_c)$ is the smoothness function that corresponds to the motion vector difference between neighboring blocks, N_c represents the neighboring motion vectors of the current motion vector \mathbf{u} , and λ is a weighting parameter. Eq. (2) is a specific formulation of Eq. (1), where $I_c(x)$ is the intensity of a pixel at position x in the current block. $I_r(x + \mathbf{u})$ is the intensity of the corresponding pixel $(x + \mathbf{u})$ in the reference block, $\theta(\mathbf{u}, \mathbf{v}_c)$ is a threshold continuity function that is equal to zero when the difference between \mathbf{u} and \mathbf{v}_c is larger than a predefined threshold, otherwise, it is equal to one. During the optimization of the energy function $E(\mathbf{u})$, \mathbf{u} varies within the search range S that is a 2-D value table, i.e. $(\pm 16, \pm 16)$. The final estimated motion vector $\hat{\mathbf{u}}$ that optimizes the energy function $E(\mathbf{u})$ is defined as follows:

$$\hat{\mathbf{u}} = \underset{\mathbf{u} \in S}{\text{argmin}} E(\mathbf{u}) \quad (3)$$

In general, MAP-based methods outperform the conventional BMA method. However, in areas with small objects in which the motion vectors are different from the motion of the surrounding background, over-smoothing in motion vector typically occurs. In these areas, BMA tends to yield more accurate motion vectors. Thus, this paper proposes an algorithm for using motion vectors obtained by BMA for the motion estimation of small object areas.

B. HIERARCHICAL MOTION ESTIMATION

For real time operation of LCD TVs, an FRUC algorithm must be sufficiently fast to process 60 frames per second. In order to satisfy this strict requirement, a hierarchical motion estimation has been often used for reducing the computational complexity [8]–[10]. To obtain precise motion vector field, the MAP method is used at the top level [9]. Subsequently, the top motion vectors are propagated to the bottom level to produce finer motion vector fields. In this manner, the images at all pyramid levels are partitioned into

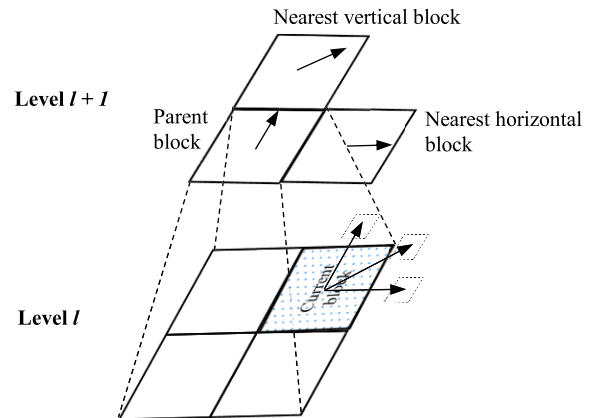


FIGURE 2. Hierarchical motion estimation [9].

blocks of the same size. To estimate motion vector of a block at a lower level, three motion vectors from the upper level are used as its initial motion vectors. The first one is from its parent block, and the other two are from the blocks both horizontally and vertically adjacent to the parent block. Fig. 2 illustrates an example. The full search around the three initial motion vectors with a search distance of $\pm d$ pixels are performed to choose the best motion vector in three search windows. The motion estimation for each layer is recursively performed in this manner from the top to the bottom levels in the image pyramid. If there are missing motion vectors at the top level, the propagation cannot discover the missing ones at the bottom level. This is the primary drawback of the conventional hierarchical motion estimation. Thus, this paper proposes a new hierarchical motion estimation algorithm that discovers the missing motion vector of small objects at the top level and propagates it into the bottom level. With this manner, the proposed algorithm successfully preserves the motion vector of small objects in hierarchical motion estimation framework.

III. HIERARCHICAL MOTION ESTIMATION FOR A SMALL OBJECT

In hierarchical motion estimation, an input image is down-sampled for generating the top pyramid layer in which the size of an object becomes smaller than that in the bottom layer. Consequently, a small object typically occupies only a small part of a block at the top level. Therefore, the small object may be ignored in motion estimation, and the remaining region of a block contributes more significantly to motion estimation than the small object does. If the motion information for a small object is ignored at the top level, it cannot be recovered at the bottom level. Thus, hierarchical motion estimation often fails in the generation of a correct motion vector for a small object. However, this small object may be sufficiently large to occupy an entire block at the bottom level, and the erroneous motion vector of a small object can deteriorate the image quality in MC-FRUC. Therefore, it is necessary to store the motion information of a small object at the top level and to pass it to be used for motion estimation at the bottom level.

This paper aims to propose a novel algorithm for hierarchical motion estimation that avoids the artifact in the region that includes a small object. In the algorithm proposed in [9], each block at a lower level has three motion vector candidates: one from the motion vector of the parent block, and the other two from the motion vectors of the nearest neighboring blocks of the parent block in the horizontal and vertical directions. This paper proposes the use of an additional motion vector candidate that represents the motion information of a small object at the top level. The additional candidate is propagated to the lower level and used for motion estimation of a small object.

A. AN ALTERNATIVE MOTION VECTOR FOR HIGH COST PIXELS

The proposed algorithm attempts to detect a small object that has a motion vector different from that of the block that includes a small object. In this case, it is possible to have a case that the movement of a small object is different from that of the surrounding area in the block. The matching error of the block may be high because small object pixels may not have matching pixels in a reference block. In this case, the matching error of the pixels that belong to a small object is high. The pixel difference, ΔI , is defined by the following equation:

$$\Delta I = |I_c(i, j) - I_r(i + u, j + v)| \tag{4}$$

where $I_c(i, j)$ is the intensity of the pixel at position (i, j) in the current frame. $I_r(i + u, j + v)$ is the intensity of the corresponding pixel $(i+u, j+v)$ in the reference frame. Vector (u, v) is the motion vector of the current block to be derived.

Herein, a pixel with a large pixel difference is referred to as a *high-cost pixel* that has a potential to be a pixel of a small object. If the pixel difference is larger than the predefined threshold, it is determined as a high-cost pixel.

When a block contains high-cost pixels, the second full search motion estimation for the high-cost pixels is performed to estimate the motion vector of a small object that consists of these pixels. In the second motion estimation, only the matching cost of the high-cost pixels is considered, and thus, a motion vector of a small object can be found. A motion vector that represents the motion of a small object is referred to as an *alternative motion vector*.

Fig. 3 shows an example of motion estimation for a 4x4 block at the top level. In Fig. 3 (a), the current block includes a part of a small object that is represented in black, and it does a rest part of a background that is in white. In the first motion estimation, a motion vector of a block is estimated as +3. Fig. 3 (b) shows a matching block in the reference frame. When the current block is compared with the matching block, the pixel difference of the small object is high, and thus, these pixels are determined as high-cost pixels which are represented by shaded pixels in Fig. 3 (c). In the second motion estimation, only high-cost pixels are used for computing SAD, and an alternative vector of -1 is derived in this example. In the proposed algorithm,

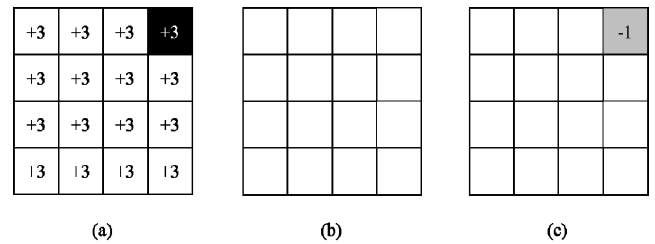


FIGURE 3. Example of the alternative motion vector: (a) a current block with a small object in black pixels; (b) a matched block in a reference frame; (c) high-cost pixels and the alternative vector obtained by the second motion estimation.

two motion vectors of +3 and -1 are to be propagated to the lower layers. If the number of layers is three as shown in Fig. 1, the motion vectors of +12 and -4 are obtained at the bottom layer. For each block at the top level, two motion vectors are derived. The first one represents the motion of the block, and the second one represents a motion of a small object in the block. The propagation of both motion vectors to the finer levels allows the motion of the small object to be preserved from the top layer to the bottom layer.

Each block has two motion vectors. One is the motion vector of the block from the first motion estimation, and the other is the motion vector of high-cost pixels from the second motion estimation. Even when no high-cost pixel exists in a block, the motion vector of a block can be wrong owing to over-smoothing of the MAP-based methods. This case may occur for a block in which all pixels belong to a small object with its size almost the same as the block size. In this case, the BMA can obtain a true motion vector of the block. However, the true motion vector can be replaced with a false one by MAP when the small object moves in a direction different from that of the background. In the proposed algorithm, the motion vector from the BMA is assigned to an alternative vector for the blocks that do not contain high-cost pixels. This ensures that all potential motion vectors for a small object are propagated to the lower layers.

B. MODIFIED HIERARCHICAL MOTION ESTIMATION

In the proposed algorithm, four motion vectors from the upper level of an image pyramid are used as the initial motion vectors for motion estimation. The three motion vectors are the same as those of the conventional algorithm [9]. The additional candidate is the alternative motion vector discussed in the previous subsection. If a block at the top level includes high-cost pixels, the alternative motion vector is the motion vector of the high-cost pixels. Otherwise, a motion vector obtained by the BMA for a block at the top level is used for an alternative motion vector. Four motion vectors are propagated to the lower level, and then the full-search BMA around the four motion vectors with a search distance of $\pm d$ pixels are performed to choose the best among the four search windows. Even when the alternative motion vector is not selected as the best one, it is still propagated to the next lower level to preserve the motion vector of the small object.

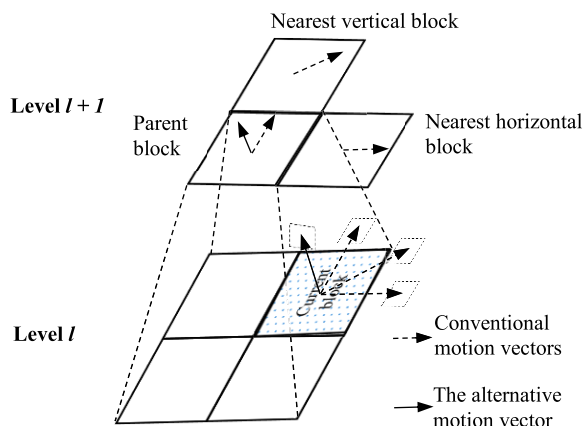


FIGURE 4. Modified hierarchical motion estimation.

Motion estimations for the lower layers are performed in this manner again in the image pyramid as shown in Fig. 4. At a finer layer or level l , in each current block (pattern fill block in Fig. 4), three dashed arrows represent the three conventional motion vectors, the other is the alternative motion vector.

C. THE PROPOSED ALGORITHM

The flowchart of the proposed algorithm is shown in Fig. 5. From input frames, image pyramids are constructed for hierarchical motion estimation. Then, a conventional full search BMA is performed at the top pyramid level and the high-cost pixels of each block are detected. In the next step, two operations are performed in parallel. One is a MAP-based motion estimation that is performed as a refinement of the BMA [9]. The other is a full-search motion estimation for high-cost pixels. When a block does not include high-cost pixels, the motion vector estimated by BMA is used for an alternative motion vector. Therefore, all blocks have two motion vectors. One is estimated by the MAP-based motion estimation and the other is the alternative motion vector. These two motion vectors of the top level are propagated to the lower level in which these vectors are used for generating search windows. After the motion estimation of the level is completed, the motion vector from BMA and the scaled alternative motion vector for each block are propagated to the next level. The alternative motion vector is propagated to the next pyramid level irrespective of whether it is chosen as the motion vector of the block or not, thereby guaranteeing that the motion vector of the small object is propagated to the bottom layer.

IV. EXPERIMENTAL RESULTS

For experiment, the proposed algorithm is evaluated with four full-HD video sequences that contain small objects: tennis ball, rim ball, basketball and soccer ball. For video frames in the dataset, odd frames are removed and these frames are used as the ground truth frames. Motion compensated frame-rate up-conversion algorithms are applied to even frames to generate intermediate frames, which are compared to the

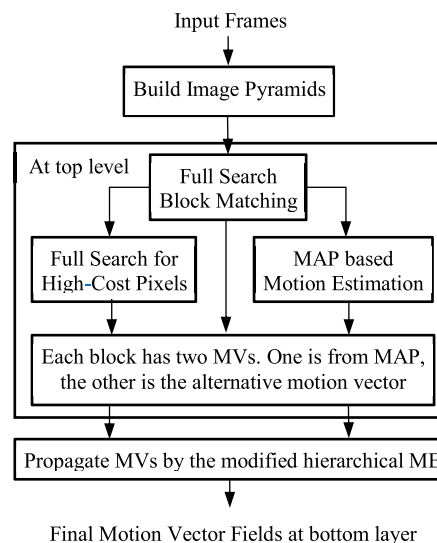


FIGURE 5. Motion estimation of the proposed algorithm.

corresponding ground-truth frames. The performance of the proposed motion estimation is compared to that of the previous method that uses the MAP algorithm at the top pyramid level and conventional hierarchical motion estimation [9]. For motion estimation, experiment is carried out with the previous and proposed algorithms under identical conditions as follows: three temporally consecutive original frames are used for estimating both forward and backward motion vector fields as suggested by [9], the number of pyramid levels is four, and the block size is fixed to 8×8 for all pyramid levels. One block at level l is a parent of four blocks at level $l + 1$. At the top pyramid level, the search range is ± 16 pixels in the horizontal direction and ± 8 pixels in the vertical direction in order to reduce search space in the vertical direction. At the other levels, the small search range d is ± 1 for both horizontal and vertical directions. The image size at the top level is 240×135 pixels, at the bottom level is 1920×1080 pixels. For frame interpolation, the algorithm in [23] is used for both previous and proposed motion estimations. The peak signal-to-noise ratio (PSNR) values of interpolated frames are used for objective comparison. In addition, subjective visual image quality is also compared.

A. EFFECT OF AN ALTERNATIVE VECTOR

Fig.6. presents an example of over-smoothing of MAP approach. Figs. 6 (a) and (b) show two consecutive input frames. Fig. 6 (c) shows a magnified input image that includes a small object. In this figure, white lines represent the blocks corresponding to the blocks at the top pyramid level. The blue arrows represent the motion vectors estimated by BMA, the red arrows represent the motion vectors estimated by MAP. For the two center blocks that contain a part of the ball, BMA estimates correctly the motion of the part of the ball while MAP over-smooths it to make the motion vector of the ball similar to those of neighboring blocks that belong to the background with different movement. Fig. 6 (d) shows

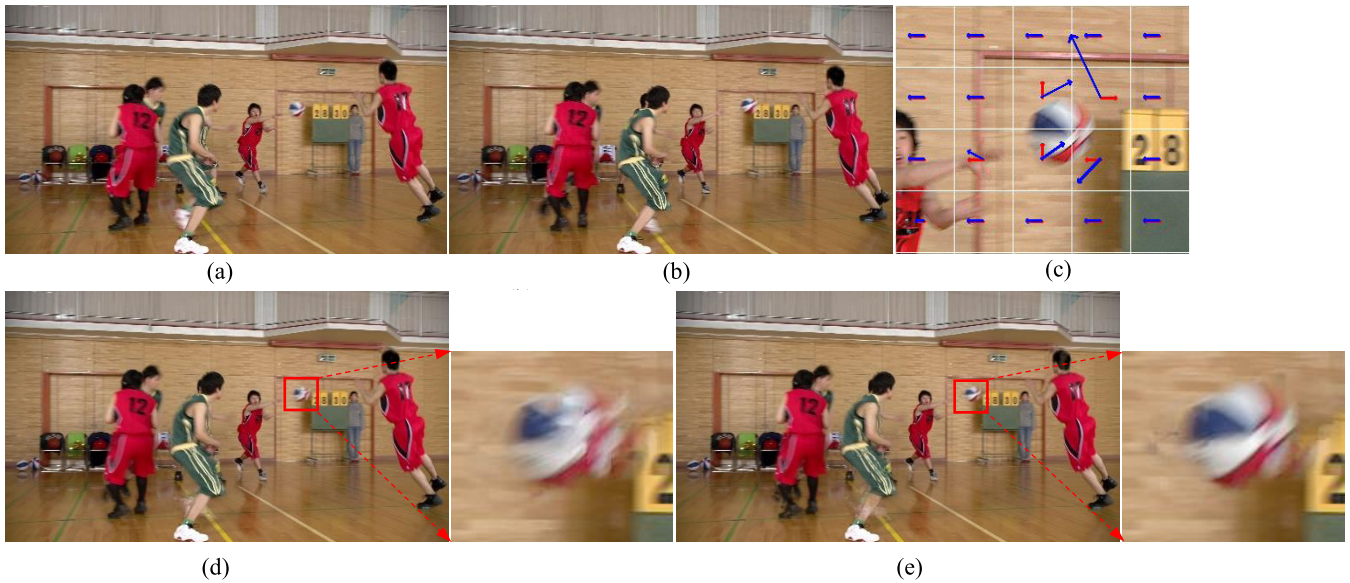


FIGURE 6. Effect of over-smoothing of MAP and the alternative motion vector with BMA. (a) the first frame; (b) the second frame; (c) the scaled motion fields with the alternative motion vectors obtained by BMA (the blue arrows are the alternative motion vectors, the red arrows are the motion vectors of blocks obtained by MAP); (d) the interpolated frame without the alternative motion vector; (e) the interpolated frame with the alternative motion vector.

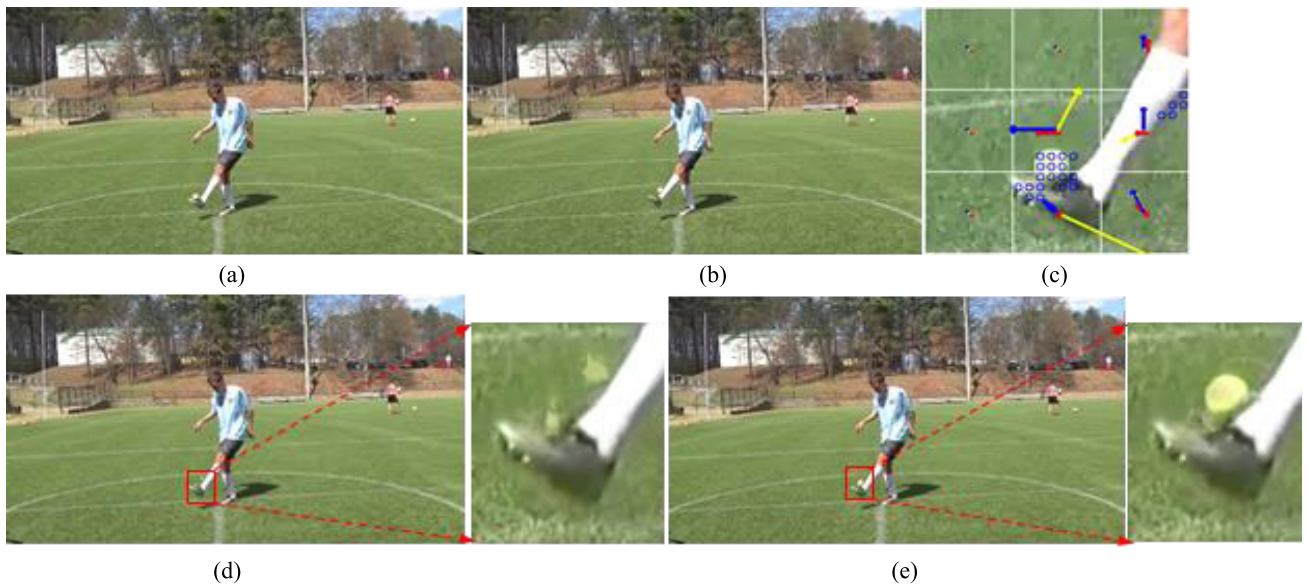


FIGURE 7. Effect of down-sampling and the alternative motion vector with the detected high cost pixels. (a) the first frame; (b) the second frame; (c) the scaled motion fields with the alternative motion vectors obtained by the second motion estimation for the high cost pixels (the yellow arrows are the alternative motion vectors, the blue arrows are the motion vectors of the blocks obtained by BMA, the red arrows are the motion vectors of the blocks obtained by MAP); (d) the interpolated frame without the alternative motion vector; (e) the interpolated frame with the alternative motion vector.

the interpolated frame when MAP is used and broken artifact is generated owing to some parts of the object generated with the erroneous motion vectors. Fig. 6 (e) shows the interpolated one when the alternative motion vector with BMA is used. The interpolated frame with the alternative motion vector preserves well the shape of the ball. This proves that the efficiency of the preservation of the motion vector of small objects with the alternative motion vector obtained by the BMA.

Fig. 7 presents an example of the alternative motion vector for the detected high cost pixels. Figs. 7 (a) and (b) show

two original frames. In Fig. 7 (c), each block represents a top level block at the top pyramid level, and it is scaled to a corresponding 64×64 block at the bottom pyramid level. The blue and red arrows represent the motion vectors obtained by the conventional BMA and MAP, respectively. The blue dots denote high-cost pixels. The yellow arrows show the alternative vectors represent the movement of the detected high cost pixels. The small tennis ball moves to the upper right corner. However, this movement is dismissed by the dominance of the background (grass) in the block. If only the motion vector obtained by BMA or MAP is propagated,

TABLE 1. PSNR comparisons between the MAP algorithm [9] and the proposed method.

Test sequence	PSNR (dB)		
	MAP algorithm [9]	Proposed method	Improvement
Rim ball	35.30	35.83	0.53
Tennis ball	34.71	34.96	0.25
Basketball	29.06	29.46	0.40
Soccer ball	38.97	39.48	0.51
Average			0.42

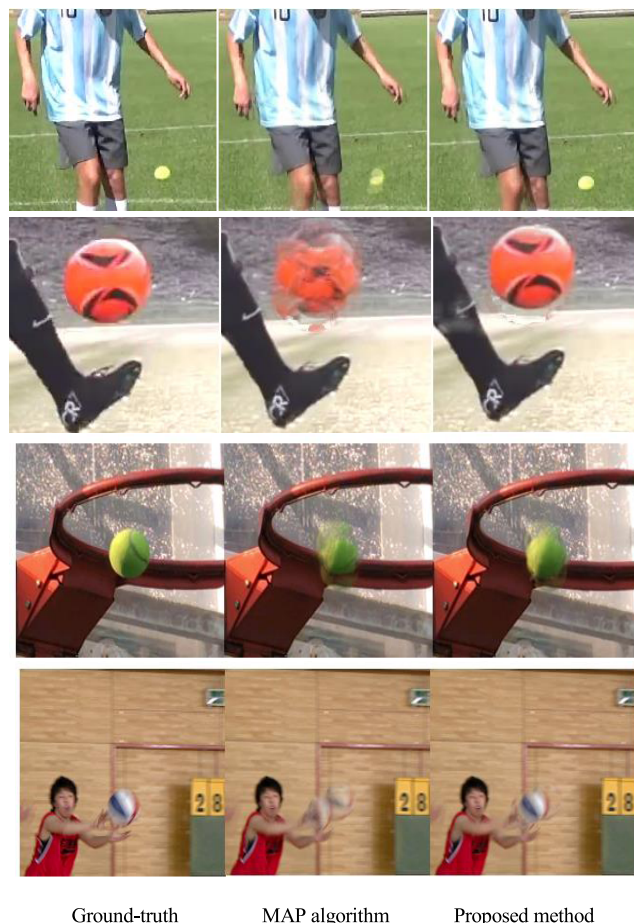


FIGURE 8. Comparison between the ground truths, the interpolated frames obtained by the MAP algorithm [9] and the proposed algorithm.

the true motion of the tennis ball cannot be found at the bottom layer. Then, the tennis ball can be missed or exist with the deformed shape in the interpolated frame as shown in Fig. 7 (d). With the proposed alternative vector, the true motion vector of the tennis ball is persevered and propagated to the bottom pyramid level. Therefore, it guarantees that the correct motion vector of the tennis ball can be used for frame interpolation that generates the intermediated frame as shown in Fig. 7 (e).

B. PERFORMANCE EVALUATION

The objective quality of the proposed algorithm is compared to that of the MAP algorithm in [9] in Table 1 which shows the comparison of the PSNR. The improvement achieved by

the proposed algorithm is about 0.42 dB on average. Fig. 8. presents the comparison of the subjective image qualities of the previous work in [9] and the proposed method. The first column represents the ground truth frames, the second column shows the interpolated frames of the previous work in [9], and the third one shows the frames generated by using the proposed algorithm with an alternative vector. In the MAP algorithm, there are broken artifacts in the interpolated frames because the motion vectors of some parts of the small balls are lost. The proposed algorithm reduces the broken artifacts significantly in comparison with the MAP algorithm. The proposed algorithm preserves the shapes of the small object because the motion vectors of the whole parts of the small object are estimated and preserved by the alternative vectors.

V. CONCLUSION

In conventional block-based hierarchical motion estimation, a motion vector of a small object is not detected at the top level, and thereby resulting in object deformation in interpolated frames in MC-FRUC. This paper proposes a new algorithm for estimating the motion of a small object in a hierarchical motion estimation framework, which improves the image quality of an interpolated frame. The proposed algorithm detects high-cost pixels for each block, and estimates the motion vector of high-cost pixels. This motion vector is used as an additional motion vector candidate in hierarchical motion estimation. The additional motion vector is propagated to the bottom level, and thus enabling a motion vector of a small object to be discovered at the bottom level. Experimental results for MC-FRUC show that the proposed algorithm achieves a better performance than the MAP algorithm in terms of both subjective image quality and objective measurements. The PSNR is improved by 0.42 dB on average by using the proposed algorithm.

REFERENCES

- [1] Y.-N. Liu, Y.-T. Wang, and S.-Y. Chien, "Motion blur reduction of liquid crystal displays using perception-aware motion compensated frame rate up-conversion," in *Proc. IEEE Workshop Signal Process. Syst.*, Oct. 2011, pp. 84–89.
- [2] T.-H. Tsai and H.-Y. Lin, "Hybrid frame rate upconversion method based on motion vector mapping," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 11, pp. 1901–1910, Nov. 2013.
- [3] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2012.
- [4] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.
- [5] T.-H. Tran and C.-T. LeDinh, "Frame rate converter with pixel-based motion vectors selection and halo reduction using preliminary interpolation," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 2, pp. 252–261, Apr. 2011.
- [6] S.-G. Jeong, C. Lee, and C.-S. Kim, "Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4497–4509, Nov. 2013.
- [7] G. G. Lee, C.-F. Chen, C.-J. Hsiao, and J.-C. Wu, "Bi-directional trajectory tracking with variable block-size motion estimation for frame rate up-converter," *IEEE J. Emerging Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 29–42, Mar. 2014.

- [8] J.-G. Kim and D. Lee, "Frame rate upconversion using pyramid structure and dense motion vector fields," *J. Electro. Imag.*, vol. 25, no. 3, p. 033004, May 2016.
- [9] D. Wang, L. Zhang, and A. Vincent, "Motion-compensated frame rate up-conversion—Part I: Fast multi-frame motion estimation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 133–141, Jun. 2010.
- [10] A. Heinrich, R. J. van der Vleuten, and G. de Haan, "Perception-oriented methodology for robust motion estimation design," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 3, pp. 463–474, Jun. 2014.
- [11] S. Dikbas and Y. Altunbasak, "Novel true-motion estimation algorithm and its application to motion-compensated temporal frame interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2931–2945, Aug. 2013.
- [12] N. Van Thang and H.-J. Lee, "An efficient non-selective adaptive motion compensated frame rate up conversion," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)* May 2017, pp. 1–4.
- [13] D. Choi, W. Song, H. Choi, and T. Kim, "MAP-based motion refinement algorithm for block-based motion-compensated frame interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1789–1804, Oct. 2016.
- [14] N. Jacobson, Y.-L. Lee, V. Mahadevan, N. Vasconcelos, and T. Q. Nguyen, "A novel approach to FRUC using discriminant saliency and frame segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2924–2934, Nov. 2010.
- [15] A.-M. Huang and T. Q. Nguyen, "A multistage motion vector processing method for motion-compensated frame interpolation," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 694–708, May 2008.
- [16] H. Liu, R. Xiong, D. Zhao, S. Ma, and W. Gao, "Multiple hypotheses Bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1188–1198, Aug. 2012.
- [17] N. Van Thang and H.-J. Lee, "A semi-global motion estimation of a repetition pattern region for frame interpolation," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 2563–2566.
- [18] S.-H. Lee, O. Kwon, and R.-H. Park, "Motion vector correction based on the pattern-like image analysis," *IEEE Trans. Consum. Electron.*, vol. 49, no. 3, pp. 479–484, Aug. 2003.
- [19] S.-G. Kim, T.-G. Ahn, and S.-H. Park, "Motion estimation algorithm for periodic pattern objects based on spectral image analysis," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jan. 2013, pp. 310–311.
- [20] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, 1981.
- [21] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L¹ optical flow," in *Proc. 29th DAGM Conf. Pattern Recognit.*, 2017, pp. 214–223.
- [22] C. Bartels and G. de Haan, "Smoothness constraints in recursive search motion estimation for picture rate conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1310–1319, Oct. 2010.
- [23] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion—Part II: New algorithms for frame interpolation," *IEEE Trans. Broadcast.*, vol. 56, no. 2, pp. 142–149, Jun. 2010.



JIWOONG CHOI received the B.S. degree in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2015, and the M.S. degree in electrical and computer engineering from Seoul National University, Seoul, in 2017, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include hardware accelerator design, computer architecture, and parallel processing for multimedia applications.



JU-HYEON HONG received the B.S. and M.S. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests are in the areas of image/video processing and visual display.



JIN-SUNG KIM received B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 1996, 1998, and 2009, respectively. From 1998 to 2004 and from 2009 to 2010, he was with Samsung SDI Ltd., Cheonan, South Korea, as a Senior Researcher, where he was involved in driver circuit and discharge waveform research. From 2010 to 2011, he was a Post-Doctoral Researcher with Seoul National University. In 2011, he joined the Department of Electronic Engineering, Sun Moon University, Asan, South Korea, where he is currently an Associate Professor. His current research interests include algorithm and architecture for video compression and computer vision.



HYUK-JAE LEE received the B.S. and M.S. degrees in electronics engineering from Seoul National University, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1996. From 1998 to 2001, he was with the Server and Workstation Chipset Division, Intel Corporation, Hillsboro, OR, USA, as a Senior Component Design Engineer. From 1996 to 1998, he was with the Faculty of the Department of Computer Science, Louisiana Tech University, Ruston, LS, USA. In 2001, he joined the School of Electrical Engineering and Computer Science, Seoul National University, where he is currently a Professor. He is a Founder of Mamurian Design, Inc., a fabless SoC design house for multimedia applications. His research interests are in the areas of computer architecture and SoC design for multimedia applications.



NGUYEN VAN THANG received the B.S. degree in electrical engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2010, and the M.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2012, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests are in the areas of image/video processing, focus on motion analysis and video frame rate up conversion.