MOTION VECTOR FIELDS BASED VIDEO CODING

Amin Zheng, Yuan Yuan*, Hong Zhang†, Haitao Yang†, Pengfei Wan* and Oscar C. Au**

*The Hong Kong University of Science and Technology
Emails: {amzheng, yyuanad, leoman, eeau}@ust.hk
†Huawei Technologies Co., Ltd

ABSTRACT

Motion vector fields (MVF) are able to produce more accurate prediction image than conventional block based motion compensation. However, the MVF is not used in conventional video coding standards due to the difficulty of efficient estimation and compression. In this work, we propose a MVF based video coding framework. We formulate the estimation of the MVF as a discrete optimization problem by both optimizing the residual energy and MVF smoothness, which can be efficiently solved by graph cut algorithm with initialized motion vectors for each pixel. We then propose a modified rate distortion optimization approach for the MVF compression. Experimental results show that the proposed method has comparable performance in terms of object quality compare to the state-of-art of HEVC, while it is with better subjective performance by overcoming the block artifacts problem.

Index Terms— motion vector fields, motion compensation, rate distortion optimization, video coding

1. INTRODUCTION

Inter prediction is essential for a video coding system to exploit temporal redundancy. In the conventional hybrid video coding system, the current frame is predicted from the previous decoded frame, then the prediction residual is converted to bit-stream by transform coding and entropy coding. The predicted frame is obtained by warping the previous decoded reference frame using motion vectors (motion compensated prediction). The motion vectors also need to be written into the bit-stream. On one hand, more accurate prediction means less residual energy, and thus less amount of resultant bits. On the other hand, the cost of coding the motion vectors should not overweight the benefit bring by the accurate prediction.

The residual energy of each pixel can be minimized independently by assigning each of them an motion vector to indicate the most similar pixel in the reference frame. However, as each pixel may have very different motion vectors, the cost of coding the motion vectors is too heavy to afford. One compromise is to use the block based motion vectors, which is adopted in most modern video coding standards. In H.264 [1], each block (the block size is from 16×16 to 4×4.) has a constant motion vector. The motion vectors are computed by searching a matching block in the reference frame with minimum mean square error (MSE). The block based scheme greatly reduces the total number of motion vectors, which makes the motion vectors compressible. However, it fails to find an accurate match for blocks containing edges and complex texture. The transform coefficients of these blocks tend to contain large high-frequency components which are difficult to compress. Besides, the reconstructed frame has block artifact, especially at the low bit-rate situation.

There have been recent efforts on motion vector fields (MVF) based video coding [2–4]. They all try to find a smooth-like dense MVF to make it both compressible and also be able to get a small residual energy. In [2], the authors apply a variation of Horn and Schunk's optical flow field [5] to get a coarse MVF, but not the pixel level motion vector. In [3], when computing the MVF, the authors not only consider the smoothness of the MVF but also the smoothness of the residual. In [4], the MVF are represented based on orthogonal wavelets. The compression efficiency of the MVF is wisely considered in the cost function for computing the MVF. A common problem of these methods is that they don't consider the influence of using compressed and distorted MVF for motion compensation in the final texture frame coding.

In this paper, we propose a new MVF based video coding framework. Firstly, we formulate the estimation of the MVF as a pixel-labeling discrete optimization problem which optimizes both the residual energy and MVF smoothness. A hierarchical block matching (HBM) approach is used to get the motion vector candidates for each pixel, then each pixel is assigned one motion vector from the corresponding candidates by efficiently solving the pixel-labeling problem using graph cut algorithm [6]. Secondly, we propose a modified rate distortion optimization approach to compress the MVF jointly considering the bit-rate and distortion of the texture frame. Experimental results show that the proposed method has comparable performance in terms of object quality compare to the state-of-art of HEVC [7], while it has better subjective performance by getting rid of the block artifacts.

The rest of the paper is organized as follows. We introduce our MVF estimation scheme in Section 2. In section 3, we show how to compress the MVF and residual jointly. Experimental results and further analysis are given in Section 4. Section 5 concludes this paper.

2. MOTION VECTOR FIELDS ESTIMATION

In this section, we aim to estimate a compressible MVF to minimize the residual energy. Computational efficiency is also considered for estimating the MVF, as it is a significant practical issue.

2.1. Problem formulation

Let $\boldsymbol{I}$ and $\boldsymbol{I_0}$ be the current frame and the reference frame, respectively. Let $\mathcal{L}$ denotes the set of all pixels and $\mathbf{x} \in \mathcal{L}$ is the two-dimension pixel index. The MVF is denoted by $\boldsymbol{u}$, then $\boldsymbol{u}(\mathbf{x})$ is the two-dimension motion vector of pixel $\mathbf{x}$. Let $R(\boldsymbol{u})$ denotes the bit-rate of coding the MVF. For each pixel $\mathbf{x}$, we denote the set of feasible motion vectors as $\mathcal{F}(\mathbf{x})$.

The MVF should ideally optimize the encoder's operational rate distortion performance [8]. Given the bit-rate budget B of coding the MVF, we wish to minimize the residual energy as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{u}} & \sum\limits_{\mathbf{x}\in\mathcal{L}} |\boldsymbol{I}(\mathbf{x}) - \boldsymbol{I_0}(\mathbf{x}+\boldsymbol{u}(\mathbf{x}))|^2 \\ \text{s.t.} & R(\boldsymbol{u}) \leq B, \boldsymbol{u}(\mathbf{x}) \in \mathcal{F}(\mathbf{x}) \end{array} \qquad (1)$$

As it is impractical to get the exact $R(\boldsymbol{u})$ without coding the MVF, we use the smoothness of MVF, i.e, the total variation of MVF $|\nabla \boldsymbol{u}|$, to approximate $R(\boldsymbol{u})$. This is a reasonable assumption that smooth fields are easy to compress [2]. Based on the Rate-Distortion Optimization (RDO) scheme [8], we rewrite (1) as Lagrangian:

$$\begin{array}{ll} \min\limits_{\boldsymbol{u}} & \sum\limits_{\mathbf{x}\in\mathcal{L}} |\boldsymbol{I}(\mathbf{x}) - \boldsymbol{I_0}(\mathbf{x}+\boldsymbol{u}(\mathbf{x}))|^2 + \lambda \sum\limits_{\mathbf{x}\in\mathcal{L}} |\nabla \boldsymbol{u}(\mathbf{x})| \\ \text{s.t.} & \boldsymbol{u}(\mathbf{x}) \in \mathcal{F}(\mathbf{x}) \end{array} \qquad (2)$$

where λ is the Lagrangian multiplier. By allowing the elements of $\mathcal{F}(\mathbf{x})$ be real numbers, (2) is a typical optical flow problem which can be solved by some iterative methods with Taylor-expansion [9]. These methods always have high computational complexity and the real numbers are also difficult to compress.

In this problem, we assume $\boldsymbol{u}(\mathbf{x})$ can only be integer or quarter precision, because higher precision motion vectors have negligible improvements for motion compensated residual compression [7]. Then the problem can be regarded as a pixel-labeling problem that each pixel $\mathbf{x}$ need to be labeled a motion vector from the candidates set $\mathcal{F}(\mathbf{x})$ in order to minimize the cost function. However, even with integer pixel precision, the possible candidates of $\boldsymbol{u}(\mathbf{x})$ for each pixel have $M \times N$ choices, where M and N denotes the frame height and width, respectively. It is time consuming to solve this pixel-labeling problem directly. Besides, each candidate set contains too many redundancy motion vectors which increases the probability of getting outliers. Next we employ the HBM approach which reduces the number of candidates to nine for each pixel, and then the reduced labels pixel-labeling problem is solved efficiently by graph cut algorithm.

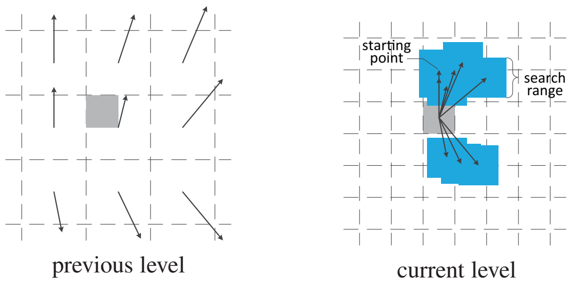


Fig. 1. Search area (blue) of current level derived from previous level.

2.2. Motion vector initialization via hierarchical block matching

In [10], the motion is hierarchically estimated using decreased block sizes in each level. We use the same idea in this paper to get the motion vector candidates for each pixel.

The hierarchical block matching starts at the block size of 64×64 which is halved in five iterations until to the 2×2 level. At each hierarchy level, the motion vectors are computed by using conventional block matching algorithm by minimizing the MSE. The search area of current level is dependent on the motion vectors of previous level, as illustrated in Fig. 1. The set of starting search points of current block is formed on the nine neighboring motion vectors of previous level, and the search range around each starting points is decreased with each level. Thus, the motion of current level is able to follow the motion of the previous level.

To reduce the effect of noise caused by small block matching, the search strategy is switched to a candidates based approach when the block size is smaller than 8×8. The motion vector of current block is directly chosen from the nine neighboring motion vectors of previous level, which means the search range is set to 0. This forces the small blocks to decide to which motion object they belong.

The hierarchical strategy is able to prevent local minimum compared to the conventional block matching approach and it also imposes the smoothness constraint implicitly through predictive search. After getting the motion vectors of each 2×2 block, the candidates set $\mathcal{F}(\mathbf{x})$ for each pixel is then defined by the nine neighboring 2×2 blocks' motion vectors. As each pixel has only nine candidates, (2) can be efficiently solved by some discrete optimization algorithm.

2.3. Energy minimization via graph cut

We choose graph cut algorithm [6] to solve the pixel-labeling problem with reduced labels. The standard form of energy function in graph cut problem is as follows:

$$E(\boldsymbol{u}) = \sum_{\mathbf{x}\in\mathcal{L}} D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x})) + \sum_{\mathbf{x},\mathbf{y}\in\mathcal{N}} V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})) \qquad (3)$$

where $\mathcal{N} \in \mathcal{L} \times \mathcal{L}$ is a neighbor system on pixels. $D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x}))$ measures the cost of assigning the label $\boldsymbol{u}(\mathbf{x})$ to the pixel $\mathbf{x}$. $V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y}))$ measures the cost of assigning the labels $\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})$ to the adjacent pixels $\mathbf{x}, \mathbf{y}$ and is used to impose spatial smoothness.

In our problem (2), $D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x})) = |\boldsymbol{I}(\mathbf{x}) - \boldsymbol{I_0}(\mathbf{x}+\boldsymbol{u}(\mathbf{x}))|^2$. As $\lambda \sum_{\mathbf{x}\in\mathcal{L}} |\nabla \boldsymbol{u}(\mathbf{x})|$ can be written as $\sum_{\mathbf{x},\mathbf{y}\in\mathcal{N}} \lambda|\boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y})|$, $V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})) = \lambda|\boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y})|$. After obtaining the candidate set $\mathcal{F}(\mathbf{x})$ in section 2.2, we use the graph cut algorithm solve (3) to get the final MVF.

3. MOTION VECTOR FIELDS COMPRESSION

Same as the depth image, the MVF is also piecewise smooth. By representing the MVF as two grey images corresponding to the horizontal components and the vertical component respectively, we can use some depth compression methods to compress the MVF, such as the intra depth coding of 3D-HEVC [11] and the graph transform based depth coding [12]. With these methods, the RDO function to choose the best coding mode for compressing the MVF will be as follows:

$$J_{mvf}(m) = D_{mvf}(m) + \lambda R_{mvf}(m) \qquad (4)$$

where m denotes the coding modes (containing prediction modes, block partition modes, etc.) of the current coding unit. $D_{mvf}(m)$ and $R_{mvf}(m)$ denote the distortion and bits of coding the current unit with mode m, respectively. And λ is the predefined Lagrangian multiplier.

However, the distortion of the MVF is not our concern to design the coding system. The distortion of the reconstructed texture frame denoted by D_{rec} is what we really care about, which is the same as the distortion of the texture residual. It is also necessary to consider the bit-rate of coding the texture residual denoted by R_{resi} when compressing the MVF. Thus we get the following modified RDO function:

$$J_{mvf}(m) = D_{rec}(m) + \lambda(R_{mvf}(m) + R_{resi}(m)) \qquad (5)$$

where $D_{rec}(m)$ and $R_{resi}(m)$ are computed by applying the compressed MVF with mode m for motion compensation. When compressing the MVF using the modified RDO fuction, the compression of the texture frame is also considered to make the total rate distortion cost minimized.

In this paper, we modify the depth intra coding algorithm of 3D-HEVC [11] to compress the MVF. The Depth Modeling Modes (DMM) in [11] is efficient to compress the piecewise smooth signal which is characterized by sharp edges and large areas of nearly constant or slowly varying samples.

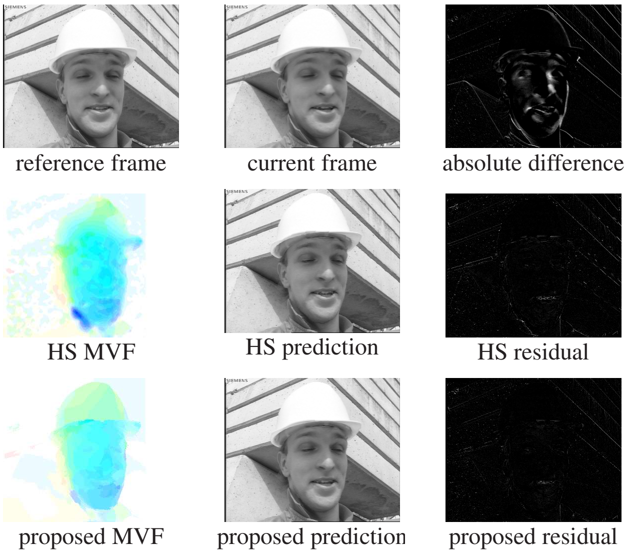


Fig. 2. MVF examples of `Foreman`. First row: first two frames and their absolute difference. Second and third row: MVF, prediction and residual obtained with Horn and Schunk's and proposed algorithm. The colored MVF is obtained based on the Middlebury optical flow benchmark [9].

We summarize here our proposed framework: Firstly, the MVF sequence is computed based on the original input sequence by using the proposed algorithm in section 2. The computed MVF sequence is represented by two sequences of gray images corresponding to the horizontal components and vertical components, respectively; Secondly, for each texture frame, the corresponding two gray images of MVF are first compressed by the proposed approach in section 3 and then the reconstructed MVF are used to perform motion compensation for the current texture frame compression; The final total coding bits are the sum of the MVF coding bits and the texture frame coding bits.

4. EXPERIMENTS

We modify the 3D-HEVC software HTM 10.0 [13] depth intra coding algorithm and texture frame coding algorithm to compress the MVF and texture frame separately. The proposed framework is compared with HEVC (HM 12.0 [14]), which is a fair comparison as the residual compression of 3D-HEVC is the same as that in HEVC. The difference between our proposed framework and HEVC is that the estimation and compression of the motion information.

The coding structure is IPPP... and the previous one frame is used as reference frame. In this paper, only the performance of integer-precision MVF is tested compared with HEVC integer-precision motion vector. Our framework can be extended to quarter-precision MVF by replacing the integer block matching in section 2.2 with sub-pixel block

Table 1. BD-Rate results versus HM12.0

ΔBD-Rate[%]		Prop-NR	Prop
Class C (832×480)	BasketballDrill	32.35	-5.69
	BQMall	41.92	-7.11
	PartyScene	10.51	-2.80
	RaceHorses	38.11	-2.92
Class D (416×240)	BasketballPass	46.32	-2.71
	BQSquare	2.28	-0.54
	BlowingBubbles	32.82	-1.85
	RaceHorse	51.64	2.32
CIF (352×288)	Foreman	68.39	1.50
	Mobile	12.37	0.68
Average		34.67	-1.91

Table 2. Ratio of motion information bits in total bits.

QP		MV Bits/Total Bits[%]			ΔBD-Rate[%]	
Texture	MVF	HEVC	Prop-NR	Prop	Prop-NR	Prop
BasketballDrill						
22	24	8.48	7.47	11.37	32.35	-5.69
27	29	13.74	7.99	17.64		
32	34	20.45	8.53	25.70		
37	39	29.45	9.16	36.44		
BQMall						
22	24	6.33	3.98	7.07	41.92	-7.11
27	29	10.73	4.49	12.74		
32	34	16.50	4.33	19.41		
37	39	24.23	4.32	28.42		

matching. Test sequences in our experiments include 10 sequences with various resolutions, seeing detail in Table 1. The test quantization parameters (QP) of texture frame are 22, 27, 32 and 37 and the QP of MVF is two larger than that of corresponding texture frame. Weighted BD-Rate [15] is used to measure the objective performance.

Fig. 2 shows one result of our MVF estimating algorithm. Compared to the conventional Horn and Schunk's optical flow field, the residual obtained by proposed algorithm has smaller energy and the predicted edge is also much sharper. Besides, based on HBM and discrete optimization algorithm with reduced labels, our approach is computational efficient.

Table 1 gives the numerical results of the proposed framework. Compared to HM 12.0, the proposed approach (denoted as Prop in the table) has an average of 1.91% and up to 7.11% BD-Rate reduction. We also test the results without modified RDO when compressing the MVF, denoted by Prop-NR. While, the BD-Rate increase is large for the Prop-NR approach which shows the necessity of considering the influence of compressed MVF for motion compensation. Fig. 3 plots the rate distortion curves of BQMall and BasketballDrill. We see that the proposed approach outperforms HEVC at different quantization levels. As shown in Fig. 4, the proposed approach has better subjective quality by getting rid of block artifacts compared to HEVC, especially at the low bit-rate situation.

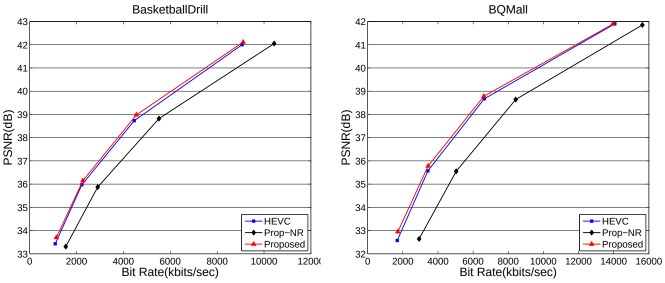
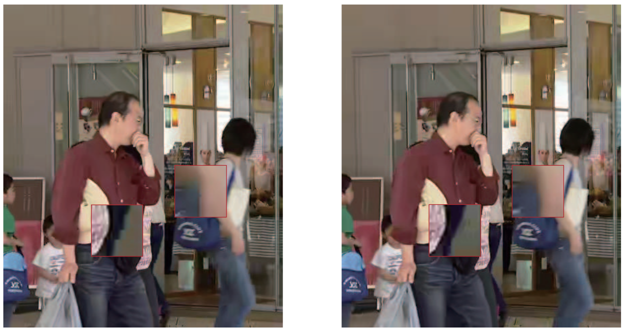


Fig. 3. Rate distortion curves of BasketballDrill and BQMall.



HEVC Proposed

Fig. 4. Decoded images of the third frame of BQMall at $QP = 37$.

The ratio of motion information bits in the total bits for BQMall and BasketballDrill is given in Table 2. Compared to HEVC, the proposed approach takes a few more percent bits to compress the motion information, but the overall performance is better, which shows that the texture residual of proposed approach is compressed more efficiently than that of HEVC. While although the Prop-NR approach takes much fewer percent bits to compress the MVF, the total BD-Rate increases a lot.

Compared to HEVC, one drawback of our framework is the computational complexity, which mainly comes from the extra MVF compression. In the future, we will try to exploit the correlation between the two components of the MVF to design a more efficient MVF compression algorithm.

5. CONCLUSION

We present a motion vector fields based video coding framework in this paper. In contrast to HEVC with block based motion vectors, the proposed framework is capable of making a better trade-off between the bit-rate of motion information and the accuracy of motion compensated prediction by using pixel based motion vectors. Experimental results demonstrate that the proposed framework has comparable performance in terms of object quality and overcomes the block artifacts, compared to HEVC. Our future work is to extend the current integer-precision MVF to quarter-precision MVF.

6. REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services*, ISO/IEC 14496-10, ITU-T Rec. H.264, 2005.

[2] P. Moulin, R. Krishnamurthy and J. W. Woods, "Multiscale modeling and estimation of motion fields for video coding," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1606–1620, Dec 1997.

[3] S.-C. Han and C. Podilchuk, "Video compression with dense motion fields," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1605–1612, Nov 2001.

[4] G. Ottaviano and P. Kohli, "Compressible Motion Fields," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2013, pp. 2251–2258.

[5] B. K. P. Horn and B. Schunk, "Determining optical flow," in *Artif. Intell.*, 1981, vol. 17, pp. 185–203.

[6] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999, vol. 1, pp. 377–384.

[7] G.J. Sullivan, J. Ohm, W.-J Han and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[8] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag., IEEE*, vol. 15, no. 6, pp. 74–90, Nov 1998.

[9] S. Baker, S. Roth, D. Scharstein, M.J. Black, J.P. Lewis and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," in *Proc. IEEE Int. Conf. Computer Vision*, Oct 2007, pp. 1–8.

[10] S. Klomp, M. Munderloh and J. Ostermann, "Decoder-side hierarchical motion estimation for dense vector fields," in *Proc. IEEE Int. Picture Coding Symp.*, Dec 2010, pp. 362–365.

[11] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken and T. Wiegand, "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *IEEE Trans. Image Processing*, vol. 22, no. 9, pp. 3366–3378, Sept 2013.

[12] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, L. Jaejoon Lee and W. Hocheon, "Edge-adaptive transforms for efficient depth map coding," in *Proc. IEEE Int. Picture Coding Symp.*, Dec 2010, pp. 566–569.

[13] "3D HEVC Test Model," `https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/`.

[14] "HEVC Test Model," `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/`.

[15] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, pp. 1–9, Apr. 2001.