AFFINE SKIP AND DIRECT MODES FOR EFFICIENT VIDEO CODING

Han Huang[a], John W. Woods[b], Yao Zhao[a,c] and Huihui Bai[a]

[a]Institute of Information Science, Beijing Jiaotong University, Beijing, China
[b]ECSE Department, Rensselaer Polytechnic Institute, Troy, NY, USA
[c]State Key Laboratory of Rail Traffic and Safety, Beijing Jiaotong University, Beijing, China

ABSTRACT

Higher-order motion models were introduced in video coding a couple of decades ago, but have not been widely used due to both difficulty in parameters estimation and their requirement of more side information. Recently, researchers have put them back into consideration. In this paper, the affine motion model is employed in SKIP and DIRECT modes to produce a better prediction. In affine SKIP/DIRECT, candidate predictors of the motion parameters are derived from the motions of neighboring coded blocks, with the best predictor determined by rate-distortion tradeoff. Extensive experiments have shown the efficiency of these new affine modes. No additional motion estimation is needed, so the proposed method is also quite practical.

Index Terms— Video Coding, HEVC, SKIP mode, Affine Motion Model, Motion Estimation

1. INTRODUCTION

In state-of-the-art video coders, motion estimation/compensation plays a key role in coding performance. Much effort has been directed to exploiting the spatial and temporal correlation between adjacent coding blocks. From this work, SKIP and DIRECT modes have emerged as powerful tools for improving coding efficiency. In the H.264/MPEG4-AVC video coding standard [1], neither prediction residual, motion vector, or reference index is transmitted for a SKIP mode macroblock. In DIRECT mode, the block motion vector is directly derived from its temporal collocated block in the reference frame. Tourapis *et al.* introduced the spatial DIRECT mode and adaptive spatial/temporal DIRECT mode in [2], where spatial correlation is also considered for deriving the motion information. It had been adopted in the later development of H.264. Laroche *et al.* proposed a competing framework for better SKIP mode in [3], where the predictors are optimally selected in a rate-distortion sense. In [4], a model-based motion-vector predictor is developed for zoom motion. In [5], an advanced B SKIP mode is proposed with decoder-side motion estimation. Another decoder-side motion-vector derivation technique is found in [6]. In the test model HM1.0 [7] of the emerging HEVC (High Efficiency Video Coding) standard, competition-based SKIP and DIRECT modes are adopted. For a given *coding unit*, a hierarchical structure of square blocks, the HM1.0 encoder first calculates the coding cost of SKIP and DIRECT modes and then proceeds to check other modes. All of the above mentioned methods are based on a translational motion model, where the motion in a block is assumed to be uniform.

Compared to a translational model, an affine motion model is able to capture more complex motion, e.g. zooming, rotation, etc. In [8], the authors employ an in-loop post-processing method to benefit from the advantages of affine motion prediction, while still employing block motion vectors for transmission. In [9], a pre-processing method is proposed. However, the affine motion compensated prediction in these two schemes is not guaranteed to produce a smaller residual signal. In [10], Glantz *et al.* introduced a parametric SKIP mode based on estimated global motion. A feature-based, robust, and global motion estimation method was used to estimate the homography transform between the current and reference frames. Then the parameters were quantized and transmitted as side information. Some related works on higher-order motion model are also found in [11, 12, 13].

In this paper, an affine motion model is integrated into the SKIP and DIRECT modes for better motion compensated prediction. The affine motion parameter predictors are derived from motion vectors of coded spatial neighboring blocks and incorporated into the competition framework of SKIP and DIRECT modes. In which, the best predictor is chosen by a rate-distortion criterion. We evaluate this novel design in the HM1.0 software with extensive experiments. Bitrate savings up to 16.38% are reported. Different from the parametric SKIP mode in [10], an additional motion estimation process is not required in our method. Thus, the additional computational overhead is low.

We present an overview of the conventional SKIP and DIRECT modes of HM1.0 in Section 2 and then described the new affine SKIP and DIRECT modes in Section 3. Experimental results are given in Section 4. We end with a discussion and conclusions in Section 5.

2. BACKGROUND

In video coding, SKIP and DIRECT modes are powerful tools to exploit spatiotemporal redundancy. In HM1.0 SKIP/DIRECT mode, a set of motion-vector predictor candidates is derived by referring to motion parameters of nearest-neighbor already coded partitions, either spatial or temporal. Then a best predictor is chosen by checking all possible candidates and selecting the one that minimizes a Lagrangian cost. This prediction is then used as the current motion vector [7]. Thus, neither motion vector nor reference-frame index is coded for the current block. The differences between SKIP and DIRECT modes are:

- Residual signal is coded in DIRECT mode. But in SKIP mode, the residual signal is skipped and the prediction signal is used to reconstruct the current block.
- Inter-prediction direction is signaled to decoder in DIRECT mode, but bi-directional prediction is always assumed in SKIP mode.

While efficient, the potential of these two modes is limited by their translational motion assumption. If motion is changing smoothly in an area, multiple translational motion-vector predictors will be generated in the candidate set. It's possible that none of them is a good one to represent motion in the current block. In this case, the affine motion could produce a better prediction and increase the chances of SKIP/DIRECT mode winning in the mode decision process. Thus, affine SKIP/DIRECT modes are introduced in this paper.

3. AFFINE SKIP AND DIRECT MODES

3.1. Affine motion model

Let (v_x, v_y) be the apparent motion at location (x, y) in the current frame. The affine motion model can then be described as:

$$\begin{cases} v_x = (1-a)x - by - e \\ v_y = (1-c)x - dy - f \end{cases} \quad (1)$$

where a, b, c, d, e, and f are the affine parameters. Let X be the current block, a square in our case. The 6 affine parameters can be derived from the 3 motion vectors at the top-left, top-right, and bottom-left corners of X, denoted respectively as $\vec{v}_i = (v_{x_i}, v_{y_i}), i = 0, 1, 2$. We can predict these 3 motion vectors from the motion of neighboring coded partitions, $A, B, ...F$, and G, as shown in Fig. 1.

Let the top-left corner be the origin of coordinates and $S \times S$ be the size of block X. Substituting (v_x, v_y) and (x, y) into Equation (1) with $(v_{x_i}, v_{y_i}), i = 0, 1, 2$ and their corresponding coordinates, and solving the resulting equations, we get

$$\begin{cases} v_x = \dfrac{v_{x_1} - v_{x_0}}{S}x + \dfrac{v_{x_2} - v_{x_0}}{S}y + v_{x_0} \\ v_y = \dfrac{v_{y_1} - v_{y_0}}{S}x + \dfrac{v_{y_2} - v_{y_0}}{S}y + v_{y_0} \end{cases} . \quad (2)$$

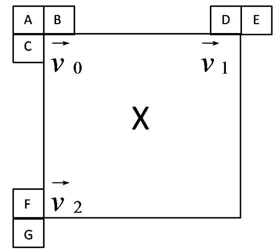


Fig. 1: The affine motion parameters represented by 3 motion vectors in a block.

In actual implementation, a scan-line algorithm can be used for faster calculation of the motion field on X. When moved from x to $x + 1$, (v_x, v_y) is changed by $(\frac{v_{x_1} - v_{x_0}}{S}, \frac{v_{y_1} - v_{y_0}}{S})$, which is constant given the affine motion model.

Motion-compensated prediction is seen as $\hat{I}_t(x, y) = \tilde{I}_{t-1}(x - v_x, y - v_y)$, where $\hat{I}_t$ is the predicted frame and $\tilde{I}_{t-1}$ is the interpolated reference frame. For the luminance component, the reference frame is pre-interpolated to $1/4$ pixel accuracy using the interpolation filter in HM1.0, then bilinear interpolation is performed if the motion is beyond $1/4$ pixel accuracy. For chroma components, the simple bilinear interpolation is adopted as in HM1.0.

3.2. Continuity in affine motion field

The affine motion field is continuous over its support. However, motion vectors for discrete pixels are calculated in the application of motion compensation. The motion difference between adjacent pixels is determined by $|\vec{v}_1 - \vec{v}_0|$ and $|\vec{v}_2 - \vec{v}_0|$. Abrupt jumps would appear if the block is over deformed. For example, the motion difference between two pixel locations (x, y) and $(x+1, y)$ is $(\frac{v_{x_1} - v_{x_0}}{S}, \frac{v_{y_1} - v_{y_0}}{S})$ by Equation (2), and a jump appears when $|v_{x_1} - v_{x_0}|$ or $|v_{y_1} - v_{y_0}|$ is large. We assume that the motion is continuous between two adjacent pixels if the motion difference is less than $1/8$ pixel. Then the X and Y components of $|\vec{v}_1 - \vec{v}_0|$ and $|\vec{v}_2 - \vec{v}_0|$ should be less than $S/8$. Such a constraint is used to check for valid affine motion parameters.

3.3. Affine SKIP and DIRECT modes

Similar to the translational SKIP/DIRECT modes, the predictors for the motion parameters in affine SKIP/DIRECT modes are derived from the motion of neighboring previously coded blocks. As shown in Fig. 1, motion vectors at A, B, and C are used to predict $\vec{v}_0$, motion vectors at D and E are used to predict $\vec{v}_1$ and motion vectors at F and G are used to predict $\vec{v}_2$. However, instead of sending the predictor index for each $\vec{v}_i$, a set of combinations of the 3 are generated. Then a single index is signaled to the decoder, thus reducing the overhead of sending predictor indices. Let $V = (\vec{v}_0, \vec{v}_1, \vec{v}_2)$ be an affine motion predictor, the best V is chosen from the set $\Omega = \{U\{\vec{v}_A, \vec{v}_B, \vec{v}_C\} \times U\{\vec{v}_D, \vec{v}_E\} \times U\{\vec{v}_F, \vec{v}_G\}\}$, where U is an operator to remove the replicas. To reduce the alphabet size of the predictor index, only those reasonable predictors are included in the candidate set. The predictor V is considered as invalid if any component of $|\vec{v}_1 - \vec{v}_0|$ or $|\vec{v}_2 - \vec{v}_0|$ is greater than $S/8$. For more efficient coding of predictor index, the candidate set should be ordered by descendant likelihood. In this paper, the candidate set is arranged in the ascending order of $D(V) = ||\vec{v}_0 - \vec{v}_1||_1 + ||\vec{v}_2 - \vec{v}_1||_1$, a measure of the deformation of the current block. A candidate predictor V is more likely to be chosen if deformation $D(V)$ is smaller. An example is shown in Fig. 2. To be consistent with the

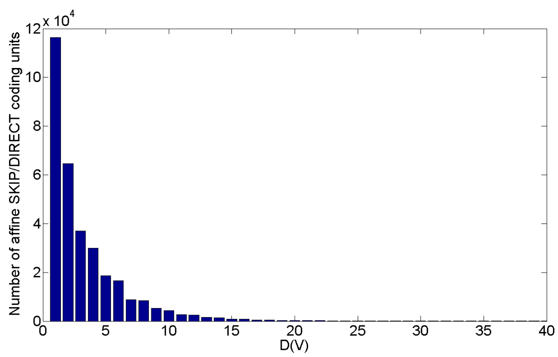


Fig. 2: Number of affine SKIP/DIRECT mode coding units in different values of $D(V)$. Results obtained from coding sequence BQSquare at low delay low complexity settings with QP equals 22

HM1.0 coder, the maximum number of affine motion predictors is set to 5 and the predictor index is unary coded. If the number of valid predictors exceeds 5, we simply discard those with higher $D(V)$ values.

The affine SKIP and DIRECT modes are embedded into the rate-distortion optimized mode decision process. When coding, we first encode the prediction mode the same as in HM1.0. If it is SKIP or DIRECT mode and the affine motion predictors are available, an additional flag is signaled to indicate whether it is affine SKIP/DIRECT mode. The encoding process for the predictor index is not changed.

Differing from the parametric SKIP using the estimated global motion [10], our affine modes exploit local motion and no additional motion estimation process is required. Our method is also different from the local affine prediction in [8] and [9]. In their schemes, the motion of the current block is first estimated or predicted in the conventional way. Then an affine motion field is derived if the motion vector of the current block and its neighboring blocks satisfy some predefined conditions. Although no extra overhead is needed, such schemes are not guaranteed to produce better predictions. And they are designed specifically for the macroblock structures in H.264/AVC.

Table 1: Low complexity encoder parameter settings.

	Low delay	Random access
Max. coding unit (CU) size	64	
CU depth	4	
Residual Quadtree (RQT) size (min./max.)	4/32	
Max. RQT depth INTER	2	
Max. RQT depth INTRA	1	
Number of reference frames	2	
Luma interpolation	12-tap directional interpolation filter	
Chroma interpolation	Bilinear interpolation	
Bit-depth	8	
Entropy coding	Variable length coding	
Generalized P-slice to B-slice	On	
Merge mode	On	
Adaptive loop filter	Off	
Motion search range	64	
Fast search	On	
Fast encoder decision	On	
Rate distortion optimized quantization	On	
Period of I-Frame	only first	32
GOP Size	1	8
Hierarchical B coding	off	on
Low-delay coding structure	on	off

Table 2: Bit Rate Reductions. LL: low delay low complexity; RL: random access low complexity.

Type	Sequence Name	Resolution	Frames	*VS.* Anchor 1 (%)		*VS.* Anchor 2 (%)	
				LL	RL	LL	RL
Class A	Traffic	2560x1600	150 @ 30fps	N/A	-0.27	N/A	-0.13
	PeopleOnStreet		150 @ 30fps	N/A	-0.06	N/A	-0.03
Class B	Kimono	1920x1080	240 @ 24fps	-0.46	-0.43	-0.52	-0.36
	ParkScene		240 @ 24fps	-1.09	-0.89	-1.85	-0.92
	Cactus		500 @ 50fps	-1.34	-1.36	-2.22	-1.33
	BQTerrace		600 @ 60fps	-0.73	-0.72	-2.27	-0.58
	BasketballDrive		500 @ 50fps	-0.42	-0.30	-0.82	-0.49
Class C	RaceHorses	832x480	300 @ 30fps	-0.59	-0.56	-1.37	-0.62
	BQMall		600 @ 60fps	-1.19	-1.02	-2.31	-1.12
	PartyScene		500 @ 50fps	**-5.08**	**-5.13**	**-8.70**	**-6.18**
	BasketballDrill		500 @ 50fps	-0.46	-0.34	-0.80	-0.42
Class D	RaceHorses	416x240	300 @ 30fps	-0.88	-0.59	-1.52	-0.65
	BQSquare		600 @ 60fps	-4.47	**-8.97**	**-16.38**	**-10.46**
	BlowingBubbles		500 @ 50fps	-3.26	-2.27	-4.72	-3.08
	BasketballPass		500 @ 50fps	-0.45	-0.36	-0.74	-0.39
Class E	Vidyo1	1280x720	600 @ 60fps	-1.30	N/A	-1.50	N/A
	Vidyo3		600 @ 60fps	-3.47	N/A	-3.79	N/A
	Vidyo4		600 @ 60fps	-1.35	N/A	-1.73	N/A
Others	Foreman	352x288	300 @ 30fps	-1.35	-1.30	-3.67	-1.42
	Mobile		300 @ 30fps	-2.23	-3.27	**-6.53**	-3.94
	Flower		250 @ 30fps	-3.46	-3.25	**-5.76**	-3.56
	Flowervase	832x480	300 @ 30fps	-2.48	-2.31	-4.81	-2.58
	Desert	720x400	240 @ 25fps	-0.35	-0.10	-0.44	-0.22
	Entertainment	720x576	250 @ 25fps	-0.97	-1.27	-2.26	-1.31
	City720p	1280x720	600 @ 60fps	-1.97	-1.35	-3.83	-1.21
	BlueSky	1920x1080	218 @ 25fps	-2.04	-1.74	-2.67	-2.08
	Station2		250 @ 25fps	-3.22	-2.69	**-5.69**	-3.38

4. EXPERIMENTAL RESULTS

The affine SKIP and DIRECT modes are incorporated into the HEVC test model HM1.0 for evaluation. The experiments were conducted with two test conditions, low-delay low-complexity (LL) and random-access low-complexity (RL), on five classes of HEVC test sequences [14]. In addition, nine other clips are used, some of which are provided by [15]. Two versions of HM1.0 anchors are generated for comparison. In anchor 1, the default encoder configuration files in the HM1.0

software [16] are used. Some major encoder parameter settings are shown in Table 1. In anchor 2, the MERGE mode is turned off and the number of reference frames is set to one per reference list. The same encoder parameter settings are used for the new coder with affine SKIP and DIRECT modes when compared to the anchors. We will refer to our new coder as *HM1.0+Affine*. Bitrate savings are measured in terms of Y BD-rate [17], calculated by 4 rate points generated with QP values $\{22, 27, 32, 37\}$.

4.1. Rate distortion performance

As we can see in Table 2, where negative values indicate average bitrate reduction, *HM1.0+Affine* outperforms the original HM1.0 coder in all cases. Bitrate savings rang from 0.06% to 8.97% in anchor 1 settings, and 0.03% to 16.38% in anchor 2 settings. Though the performance is very sequence dependent, no performance loss is observed and the maximum 16.38% bitrate saving is quite significant. The lower bounds are at the test sequence *PeopleOnStreet*, which is a surveillance video of a street by a static camera. On one hand, the background motion can be well described by the translational motion. On the other hand, the motion for the foreground objects, numerous moving people, is hard to estimate.

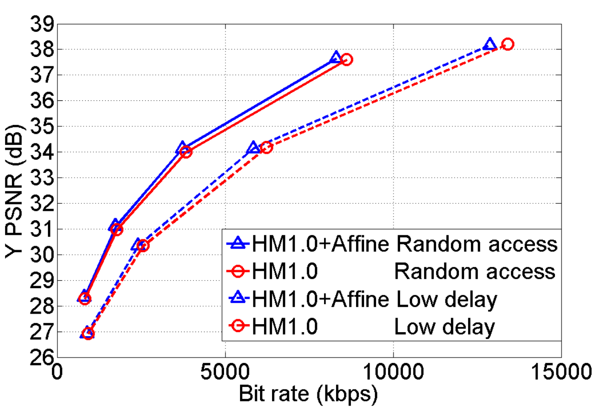


Fig. 3: RD curves of coding the sequence *PartyScene* at anchor 1 encoder settings

Table 3: Comparison with the results in [10]. The numbers represent bit rate savings in terms of Y BD-rate.

	HM1.0+PSKIP		*HM1.0+Affine*	
Sequence	LL	RL	LL	RL
BQSquare	-2.8	-3.6	**-16.38**	**-10.46**
PartyScene	-2.3	-2.8	**-8.70**	**-6.18**
BQTerrace	-2.0	-1.4	**-2.27**	-0.58
Cactus	-1.2	0.1	**-2.22**	**-1.33**
Desert	1.2	1.8	**-0.44**	**-0.22**
Entertainment	0.0	0.0	**-2.26**	**-1.31**
City720p	-3.6	-0.5	**-3.83**	**-1.21**
BlueSky	-8.2	-0.9	-2.67	**-2.08**
Station2	-29.1	-9.7	-5.69	-3.38

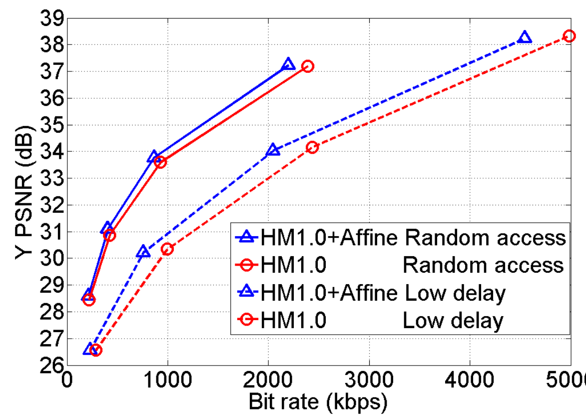
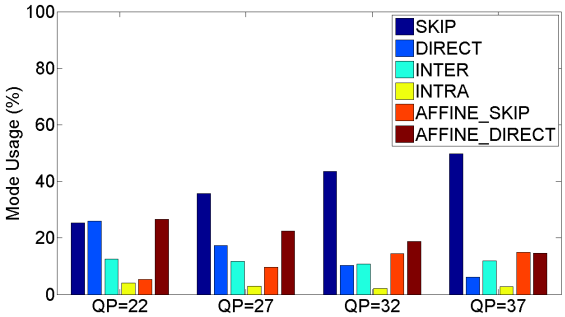


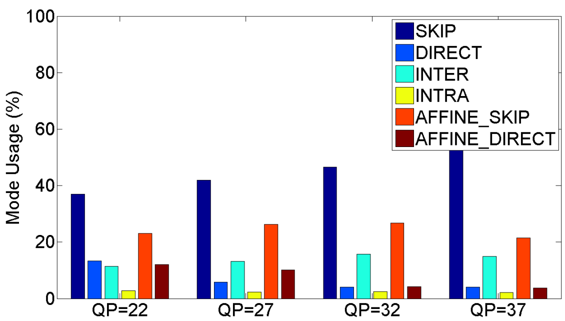
Fig. 4: RD curves of coding the sequence *BQSquare* at anchor 2 encoder settings

In Fig. 3, the R-D curves for coding the sequence *PartyScene* at anchor 1 settings are presented. The average bitrate reduction compared to the HM1.0 anchor 1 is 5.08% in low-delay condition and 5.13% in random-access condition. The R-D curves for coding the sequence *BQSquare* at anchor 2 settings are shown in Fig. 4. The average gain relative to the HM1.0 anchor 2 is 16.38% and 10.46% BD rate savings, respectively. It can be seen from the corresponding percentage coverage of all modes shown in Fig. 5 that the affine SKIP and DIRECT modes take on significant percentages in the new coder. To further demonstrate the effectiveness of the new affine modes, increases in SKIP/DIRECT mode coverage for all HEVC test sequences are shown in Fig. 6. By adding the affine SKIP and DIRECT modes, SKIP/DIRECT mode coverage generally increases, by around 3% for *BQSquare* in both low-delay and random-access conditions.

Note that coding of the new affine modes is not optimized in this paper and the HM1.0 coder already achieves bitrate reduction by $30-40\%$ when compared to H.264/AVC [10]. The results of PSKIP mode proposed in [10], using the same HM1.0 coder, are listed in Table 3 for reference. We will refer to this coder as *HM1.0+PSKIP*. The encoder parameter settings in *HM1.0+PSKIP* are similar to our anchor 2 settings, but some differences may exist. Nevertheless, the comparison shown in Table 3 still can reveal the advantages of our method. Cases when our *HM1.0+Affine* coder achieves more bitrate saving are highlighted. Motions in sequence *Station2* and *BlueSky* are mostly camera motion, which are well fit by the global motion model proposed in [10]. So the *HM1.0+PSKIP* coder performs much better than *HM1.0+Affine*.

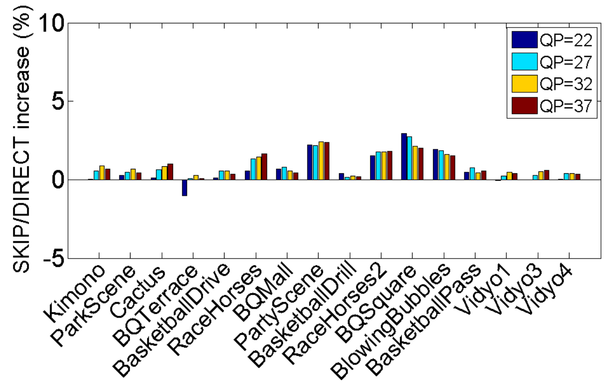


(a) Low delay

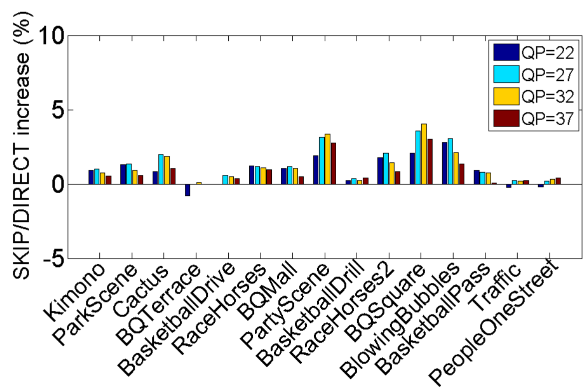


(b) Random access

Fig. 5: Mode usage coverage in the *HM1.0+Affine* coder, depending on the QP values by using the BQSquare sequence at anchor 2 settings.



(a) Low delay



(b) Random access

Fig. 6: Percentage increases of SKIP/DIRECT mode coverage, depending on the QP values at anchor 2 settings.

4.2. Complexity analysis

The computational complexity added by the affine SKIP and DIRECT modes comes from three parts:

1. Derivation of the affine motion predictors;
2. Extra interpolations required by affine motion compensation;
3. Additional coding/decoding process for calculating the mode costs.

The maximum number of candidates is 12. Calculation of $D(V)$ for each candidate and ordering are trivial. In affine motion compensation, the additional computation is the bilinear interpolation, which is fairly cheap. The degree of deformation is limited by the continuous constraint described in Section 3.2. Therefore, the additional computation and memory requirements for the pre-interpolation process are low. Note that if some early termination techniques are employed in the mode decision process, the affine SKIP/DIRECT mode may increase the chance of early termination. To sum up, compared to the motion estimation process, the additional computational load of affine SKIP/DIRECT is low.

5. CONCLUSIONS AND DISCUSSION

In this paper, an affine motion model is introduced to the powerful SKIP and DIRECT prediction modes. In addition to translational motion vectors, affine motion parameters are derived for the current block by exploiting the motion of neighboring coded blocks. Such new modes, affine SKIP and affine DIRECT, are evaluated in the test model HM1.0 of the emerging HEVC standard. Extensive results prove the efficiency of the proposed method. Since no additional motion estimation process is required, the computational load added by the new modes is low.

More generally, we introduce the concept of affine motion predictor in this paper. It can be further developed for affine motion estimation. As in traditional block matching, some update can be estimated to refine the motion. In this way, not only we can reduce the complexity of estimating affine parameters, but also provide a method for efficient coding. Since translational motion vectors are a special case of affine motion, instead of two separate predictor candidate sets for translational and affine motion parameters, we can derive a mixed candidate set with both models for the SKIP and DIRECT modes. This will be the direction of our future work.

Acknowledgements

This work was supported in part by National Basic Research Program of China (No. 2011CB302204), National Science Foundation of China (No. 61025013), Jiangsu Provincial Natural Science Foundation(BK2011455) and Sino-Singapore JPR(2010DFA11010).

6. REFERENCES

[1] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul 2003.

[2] A.M. Tourapis, Feng Wu, and Shipeng Li, "Direct mode coding for bipredictive slices in the h.264 standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 119 – 126, Jan. 2005.

[3] G. Laroche, J. Jung, and B. Pesquet-Popescu, "Rd optimized coding for motion vector predictor selection," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 12, pp. 1681 –1691, Dec. 2008.

[4] Hui Yuan, Yilin Chang, Zhaoyang Lu, and Yanzhuo Ma, "Model based motion vector predictor for zoom motion," *Signal Processing Letters, IEEE*, vol. 17, no. 9, pp. 787 –790, Sept. 2010.

[5] T. Murakami and S. Saito, "Advanced b skip mode with decoder-side motion estimation," Tech. Rep. VCEG-AK12, ITU-T Q.6/SG16 VCEG, Yokohama, Japan, April 2009.

[6] Yue Wang, Li Zhang, Siwei Ma, and Wen Gao, "Advanced spatial and temporal direct mode for b picture coding," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, Nov. 2011, pp. 1 –4.

[7] K. McCann, B. Bross, and S. Sekiguchi, "High Efficiency Video Coding (HEVC) Test Model 1 (HM 1) Encoder De-scription," Tech. Rep. JCTVC-C402, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Guangzhou, China, Oct 2010.

[8] R.C. Kordasiewicz, M.D. Gallant, and S. Shirani, "Affine motion prediction based on translational motion vectors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 10, pp. 1388 –1394, Oct. 2007.

[9] Hoi-Kok Cheung and Wan-Chi Siu, "Local affine motion prediction for h.264 without extra overhead," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, June 2010, pp. 1555 –1558.

[10] A. Glantz, M. Tok, A. Krutz, and T. Sikora, "A block-adaptive skip mode for inter prediction based on parametric motion models," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept. 2011, pp. 1201 –1204.

[11] H. Lakshman, H. Schwarz, and T. Wiegand, "Adaptive motion model selection using a cubic spline based estimation framework," in *IEEE International Conference on Image Processing*, 2010, pp. 805–809.

[12] R. Mathew and D. S. Taubman, "Quad-tree motion modeling with leaf merging," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1331–1345, 2010.

[13] Han Huang, John W. Woods, and Yao Zhao, "Motion compensated prediction using partial mesh generation," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, sept. 2011, pp. 1677 –1680.

[14] Frank Bossen, "Common test conditions and software reference configurations," Tech. Rep. JCTVC-C500, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Guangzhou, China, Oct 2010.

[15] "TUB test sequences," `http://ftlabsrv.nue.tu-berlin.de/review/sequences/`.

[16] JCT, "HM software 1.0," SVN, `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-1.0`.

[17] Gisle Bjontegaard, "Improvements of the BD-PSNR model," Tech. Rep. VCEG-AI11, ITU-T Video Coding Experts Group(VCEG), Berlin, Germany, July 2008.