Backward Adaptive Pixel-based Fast Predictive Motion Estimation

Xiaolin Chen, *Student Member, IEEE*, Nishan Canagarajah, *Member, IEEE*, and Jose L. Nunez-Yanez, *Member, IEEE*

***Abstract*—This letter presents a novel Backward Adaptive pixel-based fast Predictive Motion Estimation (BAPME) scheme for lossless video compression. Unlike the widely used block-matching motion estimation techniques, this method predicts the motion on a pixel-by-pixel basis by comparing a group of past observed pixels in two adjacent frames, eliminating the need of transmitting side information. Combined with prediction and a fast search technique, the proposed algorithm achieves better entropy results and significant reduction in computation than pixel-based full search for a set of standard test sequences. Experimental results also suggest that BAPME is superior to block-based full search in terms of speed and zero-order entropy.**

***Index Terms*—Fast search, lossless video compression, motion estimation, pixel-based prediction, predictive search.**

I. Introduction

MANY motion estimation schemes [1]–[3] are based on Block Matching Algorithms (BMA). They divide each video frame into blocks and search the reference frame for the block that most closely matches the block to be coded in the current frame, using certain matching criteria. These schemes require transmission of side information indicating the motion vectors.

Alternatively, pixel-based algorithms (PBA) [4]–[7] are used to avoid including motion vectors in the output files. While PBA is capable of delivering good motion prediction accuracy, its high complexity makes it less popular than BMA. In this paper, we propose a new Backward Adaptive pixel-based fast Predictive Motion Estimation (BAPME) scheme for lossless video compression. Our goal is, without transmitting any overhead, to obtain high motion prediction accuracy, which is essential for lossless compression, and to reduce the complexity so that it is practical and hardware amenable. BAPME works in two stages. Firstly, an initial motion vector of the current pixel is predicted with our new prediction scheme. Secondly, starting from this initial motion vector, we employ a target window, which is the neighborhood of the current pixel, to search within a certain range in the previous frame and locate the motion vector that minimizes the Sum of Absolute Differences (SAD) between the target windows in the previous and current frame. Besides the prediction set in the first stage, the novelty of BAPME is that its search is conducted on an enhanced fast Diamond Search (DS) pattern [8], [9]. The fast search technique is normally applied in BMA to obtain a tradeoff between the number of search steps and motion estimation accuracy, but we use it in PBA not only to reduce the complexity but also to increase the prediction accuracy. This is possible due to the directional prediction function given by the initial motion vector prediction and the diamond search. Experimental results show that BAPME achieves better zero-order entropy than the pixel-based Full Search (FS) for a set of standard test sequences. Moreover, BAPME performs significantly better and faster than block-based full search. The reduced complexity and simple calculation involved make our scheme suitable for hardware implementation.

This letter is organized as follows. Section II describes the proposed pixel-based fast predictive motion estimation. Sections III and IV give the experimental results of our algorithm and the comparison with other algorithms, as well as the complexity analysis. Conclusions are drawn in Section V.

II. Pixel-Based Fast Predictive Motion Estimation

The proposed motion estimation scheme is conducted on a pixel-by-pixel basis. Although most research effort nowadays tends to use BMA, there are some natural difficulties in block-matching. First, the irregular shapes of the video objects make the division into blocks difficult; second, the rigid motion (e.g., rotation and zoom) and nonrigid motion (e.g., elastic or deformable motion) are hard to model by rigid blocks; third, the transmission of motion vectors is undesirable. To circumvent this, we introduce a fast pixel-based scheme which preserves the simplicity of block-matching, and achieves better motion accuracy without sending any side information. It works in two stages: initial motion vector prediction and fast search. We describe them in the following two subsections.

A. Initial Motion Vector Prediction

The first stage of the proposed scheme is initial motion vector prediction. Motion vector prediction has been used to reduce search steps in block-matching algorithms. However, we use it in pixel-based method not only to speed up the search but also to make better motion prediction and to reduce the likelihood of being trapped in a suboptimal minimum point.

There are some analyses on the average motion vector probability for block-matching algorithms in the literature [10], [11]. It is revealed that using full search with SAD as the block distortion measurement, for CCIR601 RGB sequences there are about 23.34% motion vectors (MV) at zero (0,0), 46.79% on the cross region (Fig. 1), 47.62% on the diamond region and 51.85% on the square region within a small absolute distance. For CIF and

Manuscript received November 06, 2008; revised February 01, 2009. Current version published March 18, 2009. This work was supported by EPSRC under Grant EP/D011639/1. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Athanassios Skodras.

The authors are with the Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1UB, United Kingdom (e-mail: Xiaolin.Chen@bristol.ac.uk; Nishan.Canagarajah@bristol.ac.uk; J.L.Nunez-Yanez@bristol.ac.uk).

Digital Object Identifier 10.1109/LSP.2009.2016475

QCIF sequences, MVs concentrate even more on the cross region. These analyses suggest that most of the motion in natural sequences at 30 frames/sec is slow and smooth and MVs are often center-biased and cross-biased. In light of this, we design a set of four MV predictors—west, neighbor, median and center predictor $(\{\vec{\mathrm{MV}}_w, \vec{\mathrm{MV}}_{\mathrm{nei}}, \vec{\mathrm{MV}}_m, \vec{\mathrm{MV}}_0\})$. Among them, the neighbor predictor is newly designed and the other three have been introduced in literature. Here we assume that each frame is scaned in RASTER order and pixels above and on the left of the current pixel are known.

1) West predictor. Based on a commonly agreed conclusion that the motion in the horizontal direction is much heavier than that in the vertical direction for natural sequences [12], we take the MV of the pixel to the west of the current pixel, $\vec{\mathrm{MV}}_w$, as one of the predictors.
2) Neighbor predictor. This new predictor makes use of the intraframe spatial correlation among pixels for motion estimation. It is based on the assumption that if the current pixel belongs to the same object with one of its known neighboring pixels, they are very likely to move together from the previous frame, in other words, be in the same relative position in the previous frame. As illustrated in Fig. 2, for example, if pixel $p_i(x, y-1)$ is identified as being in the same group with the current pixel $p_i(x, y)$ in the current frame i, we predict that $p_i(x, y)$ moves together with $p_i(x, y-1)$. So the MV of $p_i(x, y-1)$ is chosen to predict the MV of $p_i(x, y)$. However, to define which neighboring pixel is in the same group with the current pixel is a problem. Based on the observation that pixels in the same group are generally connected and have similar color or brightness, represented by pixel intensity, we decide whether a neighboring pixel (west, northwest, north and northeast to the current pixel) is a "pair" with the current pixel by comparing their intensity, and the one with smallest difference is chosen. Since the value of the current pixel $p_i(x, y)$ is unknown in a backward adaptive scheme, we consider making a good prediction of $p_i(x, y)$ and using the predicted value $\tilde{p}_i(x, y)$ for the comparison. The Gradient-Adjusted Prediction (GAP) from CALIC [13] is very effective in spatial prediction, and hence we adopt it to predict $p_i(x, y)$. Note that using this prediction does not put more burden on the computational complexity since there are often areas where temporal prediction does not perform well and spatial prediction is an alternative in a complete video coding system.
3) Median predictor. Due to the continuity and smoothness of motion in most sequences, we take the median value of the MVs of the pixels to the west, north and northeast of the current pixel as predictor $\vec{\mathrm{MV}}_m$.
4) Center predictor. There are often static areas in a video and the above analysis shows that MV is often center-biased, so we use (0,0) as an initial MV predictor.

Having determined the candidates of initial MV, we choose the initial MV with the minimum SAD in the target window.

$$\vec{\mathrm{MV}}_{\mathrm{init}} = \mathrm{argmin}_{(x,y)}\mathrm{SAD}(\mathrm{tw}) \tag{1}$$

tw denotes the target window in the pixel-based motion estimation and will be defined in the next subsection.

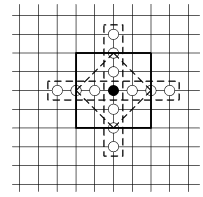

Fig. 1. Search areas: cross, diamond and square region.

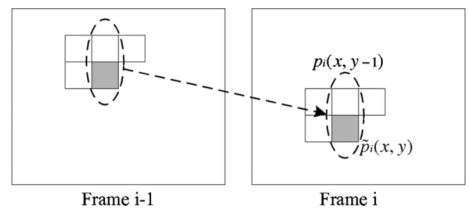

Fig. 2. Neighbor Predictor as initial MV of pixel $p_i(x, y)$. The motion vector of $p_i(x, y-1)$ is chosen if $\tilde{p}_i(x, y) \approx p_i(x, y-1)$ [see Section II-A 2)].

TABLE I
PROBABILITY OF INITIAL MVS BEING THE FINAL MVS (IN %)

sequence	$\vec{MV}_w$	$\vec{MV}_{nei}$	$\vec{MV}_m$	$\vec{MV}_0$	none
Football	68.37	61.01	56.60	19.53	20.54
Mobile	67.61	57.33	55.66	4.41	20.77
Susie	61.51	51.47	45.94	7.72	25.82
Missa	54.74	40.86	36.86	8.35	31.62
Claire	72.90	66.68	55.41	44.74	16.19
Average	65.03	55.47	50.09	16.95	22.99

Table I shows the average probability of these initial MVs being the final chosen MVs after motion estimation and the probability of none of them being the final one. If the probability is high, we regard the predictor as good because: a) it gives good $\vec{\mathrm{MV}}_{\mathrm{init}}$ prediction; b) it saves search steps in the next stage. The $\vec{\mathrm{MV}}_w$ has the highest probability, and $\vec{\mathrm{MV}}_{\mathrm{nei}}$ comes second. Although sometimes the MVs from different predictors are the same, e.g., $\vec{\mathrm{MV}}_w = \vec{\mathrm{MV}}_m$, our scheme can be more robust if it keeps the whole prediction set, especially when dealing with more complex motion. The order of the predictors is arranged according to this probability from high to low, so as to reduce the number of search steps in the fast search. Applied on the test sequences, this prediction contributes a 0.06 bpp (bits per pixel) reduction to the entropy on average comparing with the same setup without prediction.

B. Pixel-Based Fast Search

As distinct from the pixel-based full search predictor in [7], we propose to conduct pixel-based prediction on a fast search pattern in the second stage of our scheme, with the initial motion vector from the first stage as the search center. This new approach enhances the accuracy of pixel-based prediction.

The proposed predictor uses a *target window* to investigate the motion between the current frame *i* and the previous frame *i*-1. We define the *target window* as the upper-left pixels of the current pixel $p_i(x, y)$, as in Fig. 3(a). The target window size can be adjusted. Our experiments on the test set show that choosing 18 neighboring pixels is adequate and gives the best results compared to a bigger or smaller target window. For each pixel to

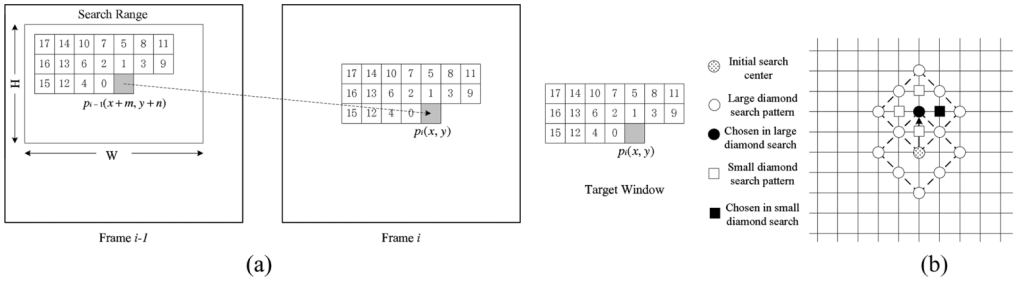

Fig. 3. Pixel-based adaptive motion estimation, see Section II-B (a) Target windows in current frame i and previous frame i-1, within search range $W \times H$. (b) Diamond Search Pattern.

be coded, the predictor searches within search range $W \times H$ in frame *i*-1 for the target window which obtains the minimal SAD, given by

$$\mathrm{SAD(tw)} = \sum_{(x,y)\in \mathrm{tw}} |p_i(x,y) - p_{i-1}(x+m, y+n)| \quad (2)$$

where tw denotes the *target window*, and $p_i(x,y), p_{i-1}(x,y)$, $(x,y) \in \mathrm{tw}$ denote the pixels in the target window in frame *i* and frame *i-1*, respectively. Motion vector of the target window is denoted by (m,n), in the region $-W/2 \le m \le W/2, -H/2 \le n \le H/2$. Using SAD as the criteria of selecting the best predictor is due to its simplicity in computation and effectiveness in our experiments. The predicted $p_i(x,y)$ is

$$\hat{p}_i(x,y) = p_{i-1}(x+m_0, y+n_0) \quad (3)$$

where (m_0, n_0) is the chosen motion vector of $p_i(x,y)$.

However, the computational complexity of pixel-based FS impedes its wide application in video coding. We consider applying the pixel-based predictor on a fast search pattern so the target window only examines a few search points instead of exhaustively examining every position within the search range. Although many advanced fine-tuned fast search patterns have been proposed, we choose the simple DS pattern for a) a very large portion of "best motion vectors" resides on the cross area [11], which is covered directly by the DS pattern; b) the DS pattern covers eight directions with similar step size so that the evaluation of different directions is adequate; c) it can achieve good accuracy with small amount of search steps. With the initial MV as search center, the target window is checked on the large DS pattern first, as in Fig. 3(b). If the MV that minimizes the SAD is not at the center, move the center to the minimum point and carry on searching until the best MV is found at the center. Then a one-step small DS pattern is applied and the MV that minimizes the SAD is chosen, as (m_0, n_0) in (3). The benefit of this method, especially when combined with prediction of the initial motion vector, is not only reducing the complexity, but also providing directional prediction on the object movement in the video. The performance results in the next section demonstrate this.

III. Performance Comparison

Since BAPME is designed for lossless video compression, the most straightforward way to evaluate its performance is to calculate the entropy of the sequence residue after motion estimation. First, we compare BAPME with other *pixel-based* motion estimation methods on the green color components of a set of CCIR601 RGB sequences in Table II. FS, hexagonal-based search (HEXBS) [14] and DS are chosen for comparison because of their benchmark status as search patterns. The search range is restricted to ± 32 and our experiment shows that enlarging the search range does not improve the result for FS. Table II shows that HEXBS and DS sometimes outperform FS, which does not occur in block-matching algorithms. This is because in BMA FS always obtains the global minimum error of each block and these error are directly coded, which naturally leads to minimal entropy. However, in PBA, although FS can always obtain the minimum SAD, it does not guarantee the best prediction since the target window does not include the pixel to be coded and the minimum SAD might be located at a suboptimal point. On the other hand, HEXBS and DS can estimate the motion directions, and BAPME even takes advantage of the intraframe spatial correlation to help with motion estimation, resulting in the best performance among the four. Compared the performance of DS and BAPME, we can also see the improvement brought by the initial MV prediction. In [7] there is a refined search approach using initial MV and smaller target window but for FS without prediction. The complete video compression system with BAPME has 0.11 bpp improvement on average over [7] on a set of YUV sequences.

TABLE II
Zero-Order Entropy of Different Pixel-Based Algorithms, in Bits Per Pixel (BPP)

sequence	FS	HEXBS	DS	BAPME
Football	4.91	5.25	5.05	4.91
Mobile	4.56	4.60	4.53	4.47
Susie	3.68	3.82	3.71	3.64
Missa	3.66	3.71	3.63	3.66
Claire	2.59	2.52	2.52	2.52
Average	3.88	3.98	3.90	3.84

We also compare BAPME to *block-based* FS with different block size. The search range of both is ± 32. Table III indicates that the compression ratio of BAPME is placed between block size 16×16 and 8×8, before taking into account the side information for encoding MV in a block-based scheme. The uncompressed side information requires $2\log_2(2 \times \mathrm{search\_range})/(\mathrm{block\_size})^2$ bits per pixel. If the side information is compressed, e.g., with JPEG-LS as in [2], the amount of bits required is shown in Table III. In our experiment BAPME is superior to conventional block-based FS algorithm in terms of overall zero-order entropy.

TABLE III
ZERO-ORDER ENTROPY COMPARISON OF BLOCK-BASED FS WITH DIFFERENT BLOCK SIZE AND THE PROPOSED METHOD, IN BPP

sequence	16×16	8×8	4×4	BAPME
Football	5.25 +0.02	4.68 +0.11	4.54 +0.62	4.91
Mobile	4.54 +0.01	4.44 +0.09	4.12 +0.59	4.47
Susie	3.81 +0.03	3.64 +0.14	3.22 +0.67	3.64
Missa	3.87 +0.03	3.65 +0.16	3.12 +0.71	3.66
Claire	2.56 +0.01	2.48 +0.09	2.11 +0.60	2.52
Average	4.01 +0.02	3.79 +0.12	3.42 +0.64	3.84

TABLE IV
COMPUTATIONAL COMPLEXITY COMPARISON OF BAPME AND BLOCK-BASED FULL SEARCH, ON AVERAGE

sequence	BAPME search points	BAPME computation	FS computation (±16)	FS computation (±32)	SIR (±16, in %)	SIR (±32, in %)
Football	15.01	270.13	1089	4225	75.19	93.61
Mobile	15.37	276.62	1089	4225	74.60	93.45
Susie	15.65	281.75	1089	4225	74.13	93.33
Missa	17.48	289.96	1089	4225	73.37	93.14
Claire	14.55	261.96	1089	4225	75.94	93.80
Average	15.34	276.09	1089	4225	74.65	93.47

IV. COMPLEXITY ANALYSIS

Previous research favors BMA over PBA for video coding, mostly because of the simplicity and speed of BMA. Here we demonstrate in Table IV that the proposed pixel-based scheme can also "enjoy both worlds" of being effective and efficient. We compare the computational complexity of BAPME and *block-based* FS in Table IV. The number of search points per pixel in BAPME is calculated by

$$\mathrm{SP}_{\mathrm{BAPME}} = \mathrm{SP}_{\mathrm{init}} + 8 + M \times (S_{\mathrm{ds}} - 1) + 4 \tag{4}$$

where $\mathrm{SP}_{\mathrm{init}}$ stands for the search points in the initial MV prediction, followed by eight search points in the first large DS pattern. S_{ds} is the search steps in DS. M is either 3 or 5, when the search direction is towards the edge or the corner of the diamond pattern respectively, in addition to four search points in the small DS. The search points for block-based FS is

$$\mathrm{SP}_{\mathrm{FS}} = (2 \times \mathrm{search\_range} + 1)^2. \tag{5}$$

The amount of SAD computation per pixel, shown from the third to fifth column of Table IV, is calculated by

$$C_{\mathrm{BAPME}} = \mathrm{SP}_{\mathrm{BAPME}} \times \mathrm{tw\_size} \tag{6}$$

$$C_{\mathrm{FS}} = \mathrm{SP}_{\mathrm{FS}}. \tag{7}$$

From the above equations we obtain the Speed Improvement Ratio (SIR) of our algorithm over block-based FS. BAPME is slightly faster than a block-based FS scheme when the search range is ±8, but is about 75% faster when the search range of FS is ±16, and is about 93% faster when the search range is ±32. For block-based FS, to reduce the block size would significantly increase the side information as described in the previous section, and to enlarge the search range would increase both the side information and computation amount.

V. CONCLUSION

In this letter, a novel Backward Adaptive pixel-based fast Predictive Motion Estimation (BAPME) scheme is proposed. It takes advantage of a new prediction set for the initial motion vector and applies pixel-based motion estimation on the fast diamond search. Contrary to the case of block-matching algorithms where the full search achieves better compression ratio than the fast search, BAPME outperforms the pixel-based full search in our experiment. Because we use "prediction" rather than "matching," the initial motion vector prediction and the diamond search provide better estimation on motion direction and reduce the chances of the motion vectors being trapped in a suboptimal point. BAPME also obtains better compression ratios than block-based FS when taking into account the side information. Moreover, BAPME is faster than block-based FS especially when the search range is large, and its simplicity in computation allows for its hardware implementation. Although mainly designed for lossless video compression, we conjecture that BAPME can be adapted in near-lossless or lossy compression. It can also be extended to fractional-pel accuracy and combined with spatial prediction into a complete video compression system.

REFERENCES

[1] N. Memon and K. Sayood, "Lossless compression of video sequences," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1340–1345, Oct. 1996.

[2] D. Brunello, G. Calvagno, G. Mian, and R. Rinaldo, "Lossless compression of video using temporal information," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 132–139, Feb. 2003.

[3] E. Carotti and J. Martin, "Motion-compensated lossless video coding in the CALIC framework," in *Proc. IEEE Symp. Signal Processing and Information Technology*, Dec. 2005, pp. 600–605.

[4] K. Yang and A. Faryar, "A context-based predictive coder for lossless and near-lossless compression of video," in *Proc. IEEE Int. Conf. Image Processing*, Sept. 2000, vol. 1, pp. 144–147.

[5] E. Carotti, J. Martin, and A. Meo, "Backward-adaptive lossless compression of video sequences," in *Proc. IEEE Int. Conf. Audio, Speech, Signal Processing*, Apr. 2002, pp. 3417–3420.

[6] E. Carotti, J. Martin, and A. Meo, "Low-complexity lossless video coding via adaptive spatiotemporal prediction," in *Proc. IEEE Int. Conf. Image Processing*, Sept. 2003, vol. 2, pp. 197–200.

[7] Y. Li and K. Sayood, "Lossless video sequence compression using adaptive prediction," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 997–1007, Apr. 2007.

[8] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[9] J. Tham, S. Ranganath, M. Ranganath, and A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 369–377, Aug. 1998.

[10] C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.

[11] C.-H. Cheung and L.-M. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimedia*, vol. 7, no. 1, pp. 16–22, Feb. 2005.

[12] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 264–290, Apr. 2006.

[13] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[14] C. Zhu, X. Lin, and L. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.