A NEW MOTION COMPENSATION METHOD USING SUPERIMPOSED INTER-FRAME SIGNALS

Ka-Ho Ng, Lai-Man Po, Kwok-Wai Cheung, Xu-Yuan Xu, Ka-Man Wong

ABSTRACT

A new MCP method called Neighbor Predicted Superimposed Search (NPSS) algorithm that uses superimposed inter-frame signals to achieve higher prediction accuracy is proposed in this paper. It outperforms other Multi-Hypothesis MCP (MHMCP) methods as it does not require the transmission of multiple motion vectors. The proposed method has better prediction quality and yet having comparable computational complexity as conventional block-based MCP with no extra side-information overhead.

Index Terms— video coding, motion compensated prediction

1. INTRODUCTION

By the Multi-Hypothesis MCP (MHMCP) theory, arbitrary number of prediction signals can be linearly combined to improve the performance of MCP [1][2]. Bi-directional prediction for B-frames is one of the applications of MHMCP in which two prediction signals, one from the reference frame before and another from the reference frame after the B-frame, are superimposed to form a prediction signal with better prediction quality. MHMCP requires the estimation of multiple motion vectors. Best prediction performance can be obtained when all the motion vectors are jointly estimated but this requires very high computation complexity. Suboptimal solutions can speedup the process [3]. In [4], it is proved that the combination of MHMCP can work together with variable block-size MCP and multiple-reference frame MCP to improve the efficiency of a rate-constrained coding scheme. In [5], two-hypothesis MCP is used to enhance the error resiliency in an error-prone environment.

The problem of MHMCP is that it requires the transmission of more than one motion vectors. In view that the residual data of MCP is nowadays getting smaller and smaller due to the advance in MCP technology, the increase in the number of motion vectors is very unfavorable. Moreover, MHMCP has high computational complexity which is unfavorable especially for real-time video coding applications. To solve these problems, we start from two-hypothesis MCP because it uses only one more motion vector than conventional MCP. An algorithm called Two-Pass Superimposed Search (TPSS) is derived which demonstrates that the inter-frame prediction signals can be efficiently superimposed using a robust weighting pair to improve the prediction quality. Furthermore, we found that instead of using block-matching motion estimation to find the lowest distortion block in the first place, we can use the motion vectors of the neighboring blocks to predict it. We called this Neighbor Predicted Superimposed Search (NPSS) algorithm. NPSS can achieve better prediction quality and yet having comparable computational complexity as conventional block-based MCP with no extra side-information overhead.

2. ANALYSIS OF SUPERIMPOSITION OF INTER-FRAME SIGNALS

In natural video sequences local minima exist in the distortion error surface of block-matching motion estimation (BMME). This means that several prediction blocks can be found in the reference frame with similar distortion values. In other words, these prediction blocks with different MVs may all resemble the current block, with pixel differences here or there over the block. In conventional BMME, only one prediction block with the lowest distortion value will be selected. We believe that although this prediction block with the lowest distortion value can roughly represent the displacement of the current block in the previous frame, the other prediction blocks can also give information about how the current block should look like. Therefore we may be able to refine the prediction signal block with the lowest distortion value by linearly combining it with another prediction signal block, with different weightings applying to these different signal sources.

Consider the conventional block-based MCP, the nth image frame F_n of size W × H of a video sequence has pixel value $F_n(s)$ for each pixel location $s=(x,y)$, in which x and y are integers such that $0 = x < W$ and $0 = y < H$. The pixel value of the reconstructed frame of a hybrid video coding system is denoted as $\tilde{F}_n(s)$. A frame is segmented into K non-overlapping blocks, $\{B_{i,n}\}_{i=1}^{K}$. Each block is predicted from the previous reconstructed frame $\tilde{F}_{n=1}$ by BMME. The motion vector (MV), $\mathbf{v}_{i,n}=(u,v)_{i,n}$, specifying a spatial displacement of ith block in F_n, is determined by

$$\mathbf{v}_{i,n} = \arg\min_{d} SAD_{B_{i,n}}(F_n(s)\ ,\ \tilde{F}_{n=1}(s=\mathbf{d})) \qquad (1)$$

where SAD (sum of absolute differences) is the most commonly used distortion measure and the 2-dimensional displacement vector **d** is limited to have finite vector component within a search area. SAD is defined as

$$SAD_B(F(s)\ ,\ G(s)) = \sum_{s \in B} \left| F(s) - G(s) \right| \qquad (2)$$

After obtaining the prediction signal block with the lowest SAD value using the BMME described above, we try to refine it by superimposing the signal block with another prediction signal block in the same reference frame. A second BMME is performed. The second MV $\mathbf{r}_{i,n} = (u,v)_{i,n}$, which specifies the spatial displacement of the lowest distortion block $\tilde{F}_{n=1}(s = \mathbf{v})$ (found in the previous BMME) in the same reference frame, is determined by

$$\mathbf{r}_{i,n} = \arg\min_{\mathbf{d}} SAD_B\left(F_n(s)\ ,\ (\tilde{F}_{n-1}(s-\mathbf{v}) \cdot w_{predicted} + \tilde{F}_{n-1}(s-\mathbf{v}-\mathbf{d}) \cdot w_{candidate})\right) \qquad (3)$$

where $w_{predicted}$ is the weighting given to the lowest distortion block $\tilde{F}_{n=1}(s = \mathbf{v})$ found in the previous BMME, $w_{candidate}$ is the weighting given to the candidate signal blocks in the second BMME.

The weightings $w_{predicted}$ and $w_{candidate}$ are used to control the signal strength given to the lowest distortion block found by conventional motion estimation and the candidate block which is supposed to refine the lowest distortion block, respectively. For example if $w_{predicted}$ = 0.7 and $w_{candidate}$ = 0.3, we can interpret the superimposed signal block as a linearly combination of the lowest distortion block with signal strength of 70% and a candidate signal block with signal strength of 30%. We call the above algorithm as Two-Pass Superimposed Search (TPSS). The TPSS algorithm is summarized as follows:

Step 1) Perform block matching between the current block and the candidate blocks in the search window base on Eq. 1. The candidate block with the lowest distortion and its MV are found.

Step 2) Base on Eq. 3, perform block matching between A) the current block and B) the weighted combinations of candidate blocks in the search window and the lowest distortion block found in Step 1. The candidate block with the lowest distortion and its MV are found.

Step 3) The residue data in Step 2 together with the two MVs found in Step 1 and 2 will be coded and transmitted.

Experiments are performed for the TPSS algorithm, using different sets of weightings $w_{predicted}$ and $w_{candidate}$. Results show that the weighting pair $w_{predicted}$ = 0.8 and $w_{candidate}$ = 0.2 achieves the best prediction quality for most of the test video sequences, compared with the conventional MCP. The average prediction quality improvement in terms of peak signal-to-noise ratio (PSNR) is 0.94 dB. This shows that a robust and fixed pair of weightings can be obtained for this algorithm. From this analysis, we can observe that the inter-frame prediction signals can be superimposed to form a signal with lower distortion and this can be implemented easily by performing conventional BMME twice. Second, a high emphasis should be put on the signal found by conventional motion estimation and this can be achieved by setting a higher weighting for the prediction signal found by the first motion estimation process and a lower weighting for the signal which is used to refine the prior signal. However, the problem with TPSS is that it requires the transmission of two motion vectors. Moreover, its computation complexity is also higher than conventional MCP.

3. SUPERIMPOSED INTER-FRAME SEARCH USING MOTION VECTOR PREDICTED SIGNAL

From the analysis in the last section, we can see that prediction quality can be improved by superimposing two inter-frame prediction signals and TPSS algorithm estimates the two MVs very easily. The problem with TPSS is that it requires the transmission of two MVs. In order to reduce the number of MVs required to one, we try to omit the first-pass BMME.

It is known that there is a very high spatial correlation between neighboring blocks. Therefore the MVs of the neighboring blocks of the current block is often used to predict the motion of the current block. This technique is called motion vector prediction (MVP) and is adopted in many video coding standards. For example, H.264 adopts the Center Biased Fractional Pel Search (CBFPS) proposed in [6], which calculates the distortion of the block pointed by the median MV and the zero motion block. The position with lower distortion will be used as the starting position of the BMME. The median motion vector $\overrightarrow{MV}_{median}$ is calculated by the median value of the MVs of the three neighboring blocks: the left, top, and top-right (or top-left if top-right is not available) blocks:

$$\overrightarrow{MV}_{median} = Median(\overrightarrow{MV}_a\ ,\ \overrightarrow{MV}_b\ ,\ \overrightarrow{MV}_c) \qquad (4)$$

where $\overrightarrow{MV}_a$, $\overrightarrow{MV}_b$, and $\overrightarrow{MV}_c$ are the motion vectors of the neighboring blocks A, B, and C of the current block E respectively, as shown in Figure 1. $\overrightarrow{MV}_c$ is replaced by $\overrightarrow{MV}_d$ if it is not available (edge of a frame).

Although MVP exploits the spatial correlation between neighboring blocks efficiently, there are cases when the blocks are not correlated. Therefore most MVP algorithms search not only the positions pointed by the predicted MVs but also the zero motion position. By searching the zero motion position at the same time, they ensure that the BMME is not performed in a direction very different from the true motion. This is like a fail-safe measure.

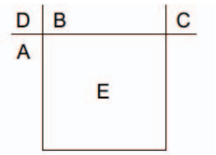

Fig. 1. Neighboring blocks.

For example the CBFPS algorithm proposed in [6] and the Motion Vector Field Adaptive Search Technique (MVFAST) algorithm proposed in [7][8] both search the zero motion position together with the positions pointed by predicted MVs.

If MVP can estimate the lowest distortion signal block without actually performing BMME, we can use this estimated signal block in Step 1 of the TPSS algorithm. Then we do not need to perform BMME in Step 1 of TPSS and can reduce the number of MVs required to one. Before we can do that, we still need to resolve one more problem. The decoder side does not know whether to use the signal block pointed by the predicted MV or the zero motion signal block.

As discussed, depending only on the signal block pointed by the predicted MV can result in low prediction quality because there are cases when the neighboring blocks are not correlated. That is why the zero motion signal block is often included for consideration. Explicitly telling the decoder side to use the signal block pointed by the predicted MV or the zero motion signal block is unfavorable because that will increase the side information overhead.

We have found a solution to this problem. By superimposing the signal block pointed by the predicted MV with the zero motion signal block, a Neighbor Predicted Signal Block (NPSB) is calculated. Eq. 5 shows how this NPSB can be calculated:

$$N(s) = \tilde{F}_{n-1}(s - \overrightarrow{\mathbf{MV}}_{median}) \cdot w_{median} + \tilde{F}_{n-1}(s) \cdot w_{zero} \quad (5)$$

where $N(s)$ is the Neighbor Predicted Signal Block (NPSB), $\overrightarrow{MV}_{median}$ is the predicted MV using the median value of the MVs of the neighboring blocks, w_{median} is the weighting given to the prediction signal block pointed by the predicted MV, w_{zero} is the weighting given to the zero motion signal block, and $w_{predicted} + w_{candidate}$=1.

This NBSP is used to substitute the signal block found in Step 1 of TPSS. Block matchings are performed between the candidate blocks and this NBSP. Motion vector $\mathbf{v}$ is found by minimizing the SAD between the candidate block & the NBSP, using Eq. 6:

$$\mathbf{v}_{i,n} = \arg\min_{\mathbf{d}} SAD_B\Big(F(s)\ ,\ (N(s) \cdot w_{predicted} + \tilde{F}_{n-1}(s - \mathbf{d}) \cdot w_{candidate})\Big) \quad (6)$$

We name this modified algorithm as Neighbor Predicted Superimposed Search (NPSS) algorithm which is summarized as follows:

Step 1) Calculate the Neighbor Predicted Signal Block (NPSB) base on the neighbor-predicted MV using Eq. 5.

Step 2) Base on Eq. 6, perform block matching between A) the current block and B) the weighted combinations of candidate blocks in the search window and the NPSB block calculated in Step 1. The candidate block with the lowest distortion and its MV are found.

Step 3) The residue data and the MV in Step 2 will be coded and transmitted.

Experiments show that NPSS has an average 0.21dB prediction quality improvement compared with conventional MCP. Although this improvement is lower than that achieved by TPSS algorithm, it is reasonable and expected. NPSS uses only one MV compared with two used in TPSS. From the results, we observe that best prediction quality is achieved when w_{median} and w_{zero} are equal to 0.5. This implies that the best prediction quality is achieved when the signal strength of the signal block pointed by the predicted MV and the zero motion signal block are the same. Moreover, experimental results show that the weighting pair $w_{predicted}$ = 0.2 and $w_{candidate}$ = 0.8 achieves the best prediction quality for most of the test video sequences, compared with the conventional MCP. This shows that a robust and fixed pair of weightings can be obtained for NPSS.

4. EXPERIMENTAL RESULTS

Experiments using seven CIF sequences: *Akiyo, Crew, Foreman, Mobile_Calendar, News, Sean,* and *Silent* are performed to analyze the performance of the proposed algorithms. These seven sequences cover a wide-range of motion contents. The block size is 16x16 pixels. The search window size is +/-16 pixels. Full search is used as the BMME algorithm. Fractional-pixel motion estimation is implemented with quarter-pixel (1/4-pixel) accuracy. Different from a real video codec that uses reconstructed reference frames, in these experiments original frames are used as reference frames. Bit-rate is not available at this stage.

Table I tabulates the peak signal-to-noise ratio (PSNR) achieved by algorithm TPSS using different $w_{predicted}$ and $w_{candidate}$ and that achieved by original MCP. The weighting pair with which the TPSS performs best in a test sequence is labeled grey. We can see that for all sequences except *Silent*, TPSS performs best with $w_{predicted}$ = 0.8 and $w_{candidate}$ = 0.2. For *Silent*, TPSS performs best with $w_{predicted}$ = 0.7 and $w_{candidate}$ = 0.3. If we select $w_{predicted}$ = 0.8 and $w_{candidate}$ = 0.2 as the weighting pair and compare the performance with original MCP, an average 0.94dB PSNR improvement can be obtained.

Table I
PSNR achieved by TPSS with different $w_{predicted}$ and $w_{candidate}$

TPSS

weight predicted	weight candidate	Akiyo	Crew	Foreman	Mobile	News	Sean	Silent
0.1	0.9	44.9385	33.5741	34.7634	27.5354	38.5420	41.1583	37.2131
0.2	0.8	45.0035	33.6628	34.8468	27.6006	38.6267	41.2605	37.3404
0.3	0.7	45.1567	33.8040	34.9796	27.7128	38.7732	41.4244	37.5406
0.4	0.6	45.2921	33.9583	35.1346	27.8305	38.9242	41.5682	37.7225
0.5	0.5	45.6304	34.1797	35.3379	27.9474	39.1340	41.8334	37.9735
0.6	0.4	45.7514	34.3280	35.4610	28.0280	39.2970	41.9521	38.1098
0.7	0.3	45.8539	34.4933	35.5752	28.0904	39.3937	42.0412	38.2055
0.8	**0.2**	**45.9321**	**34.5392**	**35.6033**	**28.1055**	**39.4050**	**42.0566**	**38.1644**
0.9	0.1	45.8401	34.3538	35.4573	27.9488	39.2190	41.8980	37.9193
Original MCP		44.9122	33.4932	34.7456	27.3228	38.5078	41.0553	37.1715
TPSS improvement over Original MCP (weight_predicted=0.8, weight_candidate=0.2)		1.0199	1.0460	0.8577	0.7827	0.8972	1.0013	0.9929
						Average Improvement:		0.9425

Table II
PSNR achieved by NPSS with different $w_{predicted}$ and $w_{candidate}$
($w_{median} = w_{zero} = 0.5$)

NPSS

weight predicted	weight candidate	Akiyo	Crew	Foreman	Mobile	News	Sean	Silent
0.05	0.95	44.9668	33.5699	34.8228	27.4319	38.5815	41.1037	37.2790
0.10	0.90	45.0469	33.6315	34.8697	27.4876	38.6497	41.1822	37.3830
0.15	0.85	45.0996	33.6667	34.8734	27.5173	38.7124	41.2196	37.4371
0.20	**0.80**	**45.1566**	**33.6713**	**34.8560**	**27.5230**	**38.7386**	**41.2600**	**37.4644**
0.25	0.75	45.2365	33.6664	34.7876	27.5136	38.7581	41.2482	37.4701
0.30	0.70	45.2384	33.5985	34.6626	27.4651	38.7325	41.2025	37.4251
0.35	0.65	45.2204	33.4980	34.4812	27.3971	38.6685	41.1411	37.3281
0.40	0.60	45.1983	33.3586	34.2597	27.3035	38.5930	41.0367	37.2068
0.45	0.55	45.1473	33.1834	33.9722	27.1865	38.4671	40.9098	37.0221
Original MCP		44.9122	33.4932	34.7456	27.3228	38.5078	41.0553	37.1715
NPSS improvement over Original MCP (weight_predicted=0.2, weight_candidate=0.8)		0.2444	0.1781	0.1104	0.2002	0.2308	0.2047	0.2929
						Average Improvement:		0.2088

Table II tabulates the PSNR achieved by algorithm NPSS using different $w_{predicted}$ and $w_{candidate}$ and that achieved by original MCP. The weighting pair with which NPSS performs best in a test sequence is labeled grey. We can see that for most sequences, the performance of NPSS peaks with ($w_{predicted}$, $w_{candidate}$) in the range of (0.15, 0.85) to (0.25, 0.75). If we select $w_{predicted}$ = 0.2 and $w_{candidate}$ = 0.8 as the weighting pair and compare the performance with original MCP, an average 0.21dB PSNR improvement can be obtained.

The experimental results match with our expectation. In TPSS the best weighting is $w_{predicted}$ = 0.8 and $w_{candidate}$= 0.2. It is because the first BMME finds the lowest distortion signal block which already resembles very much the current block. Therefore high emphasis should be put on this signal block and the weighting put on it is high. The signal strength of another inter-frame signal should be relatively lower, such that another inter-frame signal acts as a refinement signal. On the contrary, in NPSS the neighbor-predicted signal block is superimposed by the signal block predicted by the MVs of neighboring blocks and the zero motion block. No actual BMME is performed in the first step. Therefore in the subsequent step the candidate signal blocks should be given a higher strength and therefore the best weighting pair is $w_{predicted}$ = 0.2 and $w_{candidate}$ = 0.8. Although the overall PSNR improvement by NPSS is not very high, this algorithm does not need to transmit multiple MVs. Moreover, its computation complexity is also comparable to that of conventional MCP. The neighbor-predicted signal block is calculated only once. To implement the fast superimposition of the candidate blocks with this fixed neighbor-predicted signal block, a filter may be applied.

5. CONCLUSION

This paper proposes a new MCP method called NPSS which outperforms other multi-hypothesis MCP methods as it does not require the transmission of multiple motion vectors. The proposed method has better prediction quality and yet having comparable computational complexity as conventional block-based MCP with no extra side-information overhead. Future works include designing a special MVP algorithm for NPSS and dynamic weighting pairs for the different signal sources.

REFERENCES

[1] G.J. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, Apr 1993, vol. 5, pp. 437-440.

[2] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 173-183, Feb. 2000.

[3] M. Flierl, T. Wiegand, and B. Girod, "A locally optimal design algorithm for block-based multi-hypothesis motion-compensated prediction", in *Proceedings of the Data Compression Conference*, Snowbird, Utah, Apr 1998, pp. 239-248.

[4] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multihypothesis prediction for motion-compensated video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, no.11, pp. 957- 969, Nov 2002

[5] C.-S. Kim, R.-C. Kim, and S.-U. Lee, "Robust transmission of video sequence using double-vector motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 9, pp. 1011–1021, Sep 2001

[6] Z.B. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," *JVT-F017, 6th meeting*: Awaji, Japan, 5-13 Dec. 2002

[7] P. I. Hosur and K. K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation," in *Proceedings of the Second International Conference on Information, Communications and Signal Processing*, Singapore, 7-10 Dec 1999

[8] K. K. Ma and P. I. Hosur, "Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST)," in *ISO/IEC JTC1/SC29/WG11 MPEG99/m5851*, Noordwijkerhout, NL, Mar 2000