Modified Predictive Motion Estimation for Reduced-Resolution Video from High-resolution Compressed Video

Justy W. C. Wong and Oscar C. Au

Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China Email: eejusty@ee.ust.hk, eeau@ee.ust.hk

ABSTRACT

To convert a compressed video sequence to a lower-resolution compressed video, one typically needs to decompress the original sequence, down-sample each frame, and recompress it. It involves motion estimation in the reduced sequence which is computational intensive. In [1], predictive motion estimation was proposed to generate the motion vectors of the reduced video by using the original motion information in the compressed bitstream. However, it was described only for the case of 4 to 1 down-conversion. In this paper, we propose a modified predictive motion estimation to work on other down-conversion ratios in lower complexity.

1. INTRODUCTION

Nowadays, delivery of digital video through different kinds of network, such as internet, local area network, wireless network, cable TV network, is very common. Owing to the huge storage size of video, most stored video is only available in compressed format. For applications like previewing of a movie from Internet, transcoding to a lower bitrate in wireless network and down-conversion of HDTV video for low-resolution display, there is a need to reduce the resolution of the compressed video sequence which is stored in the video server and encode the reduced video into a compatible bitstream for the decoder.

In this paper, we are concerned of video compressed by ITU-T H.261, H.263, MPEG-1, MPEG-2. One thing common to these standards is that they employ motion estimation and compensation to reduce temporal redundancy between successive frames in a video sequence in order to achieve high compression efficiency.

A straight forward way to scale down the compressed video sequence is to decompress the whole sequence, apply anti-aliasing lowpass filtering, down-sample the video and then re-encode the reduced sequence by conventional video encoders which have to perform the computational intensive motion estimation again. If brute force motion estimation is used, more than 60% of the computational power of the encoder can be used by the motion estimation. Therefore, fast motion estimation is highly desirable.

As the original motion information is available in the compressed bitstream and should be highly correlated to the motion of the reduced video, it is possible to estimate the new motion vectors from the original motion vectors in order to save much of the computational cost. There are some existing algorithms for this purpose [1,2].

In [1], predictive motion estimation (PME) was proposed for down-conversion. However, it only evaluated the performance of 4-to-1 conversion. In this paper, we will evaluate the performance of PME in other down-conversion cases (e.g. 9-to-1 & 16-to-1). For downscaling from n to 1 using PME, if the n is large, the search points of PME will increase resulting in increased computation. In order to reduce the computations, we propose a modified predictive motion estimation (MPME) by selectively reduction of search points.

2. EXISTING ALGORITHMS

For *n*-to-1 video down-conversion, each side of a frame is reduced by \sqrt{n} . Let $\{MB_{o,i}\}_{i=1}^n$ be n macroblocks of the original video with corresponding the motion vectors $\{V_{o,i}\}_{i=1}^n$. The *n* macroblocks are reduced to form one macroblock

MB_r of the reduced video with corresponding motion vector V_r .

The simplest algorithm of getting the new motion vectors is to take the mean of the motion vectors of the n corresponding macroblocks in the original video and reduce it by \sqrt{n} and we call this MEAN which can be represented by the following equation:

$$V_r = \frac{1}{\sqrt{n}} truncate \left(\frac{\sum_{i=1}^{n} V_{o,i}}{n} \right)$$

The MEAN is a good estimate if the n macroblocks have motion vectors pointing to similar directions with similar magnitude. However, if the n motion vectors have different directions and/or magnitude, the resulting MEAN can be poor and meaningless. The simple arithmetic mean of the n different motion vectors is inadequate to describe all situations. In [2], algorithm called adaptive motion vector resampling (AMVR) was proposed to estimate the motion vector of the reduced video using a weighted mean of the original motion vectors.

In the compressed video bitstream of MPEG-1, MPEG-2, H.261 and H.263, only the inter-coded macroblocks in the interframes (P or B frames) contain motion vector information. AMVR applies only to these inter-coded blocks. AMVR uses a weighted mean of the n V_o to estimate V_r .

$$V_r = \frac{1}{\sqrt{n}} \frac{\sum_{i=1}^{n} V_{o,i} \cdot A_i}{\sum_{i=1}^{n} A_i}$$

where A_i denotes the activity measurement of residual macroblock i (in the original video). In [2], the number of non-zero AC coefficients was used as A_i . The complexity of AMVR is also very low but it is not necessarily good. Actually, it gives a rather low peak signal-to-noise ratio (PSNR) when compared with full search in the reduced video, as we will show in section 5.

3. REVIEW ON PREDICTIVE MOTION **ESTIMATION**

In this section, we review the algorithm PME as presented in [1] for the 4-to-1 down-conversion. PME constructs a weighted motion vector as a

candidate and also considers the four original motion vector as four other candidates. All five candidates are tested in search of the best motion vector.

For each reduced macroblock, if the 4 reference V_o $(V_{o,1}, V_{o,2}, V_{o,3}, V_{o,4})$ are the same, PME uses the common value and reduces it by one half to be V_r . Otherwise, it goes through the following steps to get the V_r .

Step 1: Compute n candidate vectors $(V_{r,i})$ as $V_{r,i} = \text{truncate}(V_{o,i})/2$

for
$$i = 1, 2, ..., n$$

 $\label{eq:for_i} \textit{for } i = 1, \ 2, ..., \ n$ The $V_{r,i}$ will be in half pixel precision.(n=4)

Compute the mean absolute difference (MAD_i) in the reduced video for each of the four candidates $\{V_{r,i}, i = 1, 2, ..., n\}$. If one of the MAD_i value is equal to zero, assign the corresponding $V_{r,i}$ as the new V_r and stop searching. Otherwise, go to step 3.

Step 3: Compute a new candidate vector $V_{r,n+1}$ by

$$V_{r, n+1} = \frac{\sum_{i=1}^{n} \left[V_{r, i} \cdot \left(\frac{1}{MAD_i} \right) \right]}{\sum_{i=1}^{n} \left(\frac{1}{MAD_i} \right)}$$

If the $V_{r,n+1}$ is not equal to one of the first n candidates, compute MAD_{n+1} using $V_{r,n+1}$ as the motion vector.

Step 5: The $V_{r,i}$ with the minimum MAD value is chosen as the final V_r .

For the MB_o that are skipped blocks or intra blocks which do not provide any motion information, we set the V_o of these blocks to be zero.

For better performance, PME can be followed by a local half-pixel search and is called PME-LS2.

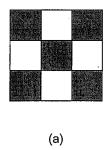
4. MODIFIED PREDICTIVE MOTION **ESTIMATION**

For downscaling from n to 1 using PME, if the n is large (e.g. 9 or 16), the number of search points of PME will be increased resulting in increased computation. To reduce the computations,

propose a modified predictive motion estimation (MPME). In MPME, we selectively reduce the possible candidates of PME.

Note that PME performs a search only when the candidate motion vectors disagree with each other. We observe that most of the disagreement is caused by having two or more objects moving to different locations and the probability of this situation increases with n. This situation always appears at the corner blocks of these n macroblocks.

We follow the same steps of PME but, in step 2, search only the candidate motion vectors of the shaded macroblocks shown in Fig.1.



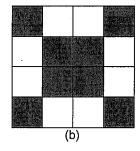


Fig.1 (a) 9-to-1 conversion and (b) 16-to-1 conversion.

For the 16-to-1 conversion, we only use the mean of the four motion vectors as 1 candidate for the four macroblocks in the center instead of searching all four candidates separately. Computation is reduced and we only have 5 candidates for both cases.

4. RESULTS

We tested our algorithms on several MPEG I video sequence with SIF resolution (352×240) and a GOP of 15 frames with IPPP frame structure in each GOP for the 9-to-1 and 16-to-1 down-conversion. In order to compare the performance of different algorithms, we calculate the peak signal-to-noise ratio (PSNR) between the predicted frames of the reduced video using different algorithms and the original reduced frames.

Fig. 2 shows the PSNR difference from full search of MEAN and MPME-LS for the 'football' sequence. The search area for full search is ± 7 for 9-to-1 conversion case.

Table 1 shows the mean PSNR values of different algorithms, the difference in mean PSNR values between full search and different algorithms and the average search point for each macroblock for the 'football' and 'salesman' test sequences.

For the case of 9-to-1 conversion, we calculate the MAD for at most 6 points for each macroblock in MPME instead of 11 points in PME and do not need to calculate for those blocks with identical candidate motion vectors. In the case of the fast moving scenes in football sequence, the average search points for each motion vector is about 2.9 for MPME and 4.2 for PME which is about 1.45 times speed up. In the case of the head-and-shoulder scene like salesman, the average search point is about 0.8 for MPME and 1.2 for PME which is 1.5 times speed up. We can achieve a even better quality with the additional halfpixel local search (MPME-LS) at the price of more search points. Note that the half-pixel search is needed anyway if half pixel accuracy is needed in the full search.

5. CONCLUSIONS

In this paper, we propose an algorithm called modified predictive motion estimation to estimate the best motion vector for reduced resolution video sequences using the motion information of the original compressed video sequence. The computational need of the modified algorithm is reduced with only very minor degradation in terms of PSNR.

6. ACKNOWLEDGEMENT

This work was funded by RGC CERG grant HKUST6009/97E.

7. REFERENCES

- [1] J.W.C. Wong, O. C. Au, P.H.W. Wong and A. Tourapis, 'Predictive Motion Estimation for Reduced-Resolution Video from High-resolution Compressed Video,' *Proc. Of IEEE Int. Conf. On Image Processing*, Oct. 1998.
- [2] B. Shen, I. K. Sethi and V. Bhaskaran, "Adaptive motion vector resampling for compressed video down-scaling," Proc. Of IEEE Int. Conf. On Image Processing, Oct. 1997.

Sequence	Football				Salesman			
	9-to-1		16-to-1		9-to-1		16-to-1	
	Mean PSNR	Average Search points						
Full Search	26.47	225	26.10	121	40.15	225	40.24	121
Mean Only	25.21	*	25.32	* '	39.30	*	40.17	*
AMVR	25.29	*	25.40	*	38.77	*	40.13	*
PME	26.24	4.1	25.97	6.0	40.12	1.2	40.23	1.2
PME-LS	26.39	6.9	26.07	9.3	40.15	2.8	40.24	2.9
МРМЕ	26.16	2.9	25.86	3.2	40.11	0.8	40.23	0.8
MPME-LS	26.35	5.5	26.01	6.3	40.15	2.1	40.25	2.1

Table 1. mean PSNR and average search points for various algorithms and various test sequences. * These two algorithms have negligible computation compared with full search.

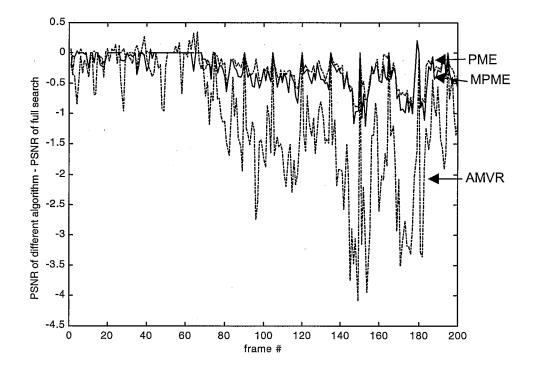


Fig.2 the PSNR difference from full search of MPME, PME, AMVR for the "football" sequence.