FAST MOTION ESTIMATION FOR VIDEO RESOLUTION DOWN-CONVERSION USING SPATIAL-VARIANT FILTER

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 this paper, a novel fast motion estimation algorithm is proposed to predict the motion vectors of the reduced-resolution video without performing any search. The motion vectors are predicted by applying spatial-variant filters to the motion vectors of the original compressed high-resolution video. In our simulations, our method outperforms other existing fast predictive algorithms which involves no searching.

1. INTRODUCTION

With the advent of video compression technology, delivery of digital video through various kinds of channels including internet, local area network, wireless network, video-on-demand network, video compact disk (VCD), digital versatile disk (DVD), etc. is feasible and popular. Owing to the huge storage size of video, most stored video is only available in compressed format. In many applications, there is a need to reduce the resolution of the compressed video sequence in the video server and encode the reduced video into a compatible bitstream for the decoder. Such applications include the previewing of a movie from Internet, the transcoding to a lower bitrate in wireless network and down-conversion of HDTV video for low-resolution display,

In this paper, we concentrate on video compression standards such as ITU-T H.261, H.263, MPEG-1, and 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]. Some yields good prediction as the expense of relatively high computation requirement [1]. Some are fast but not very good [2].

In the next session, we propose a novel algorithm, ME-SVF, to approximate the optimal motion vector for reduced video sequence using the motion information of the original compressed high-resolution video. We will show that the proposed method is better than the existing algorithms in terms of PSNR and complexity.

2. EXISTING ALGORITHMS

In an n -to-1 down-conversion of video, each dimension of the video is reduced by $\sqrt{n}$. Let

$\{MB_{o,i}\}_{i=1}^n$ be n macroblocks of the original video with corresponding 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 to get 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 $\sqrt{n}$. We call this MEAN which can be represented by the following equation:

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

The MEAN is a good estimate if the n macroblocks have motion vectors pointing to similar directions with similar magnitude. However, if the four 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], an 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 four 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 4.

In [1], predictive motion estimation (PME) is proposed. In 4-to-1 ($n=4$) video down-conversion,

PME constructs a weighted motion vector as a candidate and also consider 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 four reference vectors V_o ($V_{o,1}, V_{o,2}, \dots, V_{o,n}$) are the same, PME uses the common value and simply reduces it by $\sqrt{n}$ to get V_r . Otherwise, it use these n motion vectors as candidate vectors and search for the best match, i.e. get the lowest mean absolute difference (MAD). By using these MAD information, it computes one more candidate for an additional search by

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

The $V_{r,i}$ with the minimum MAD value is chosen as the final V_r . The performance in terms of PSNR is good in PME, however, it involves some block matching which will increase the computations.

In this paper, we are interested in improving the algorithms that do not perform any search. This includes MEAN and AMVR, but not PME. Without the searching, MEAN and AMVR are much faster than PME, but are significantly worse than PME. The proposed algorithm follows a similar trait, but yields better peak-signal-to-noise ratio (PSNR) than MEAN and AMVR.

3. THE PROPOSED ME-SVF

The proposed algorithm, motion estimation by spatial-variant filtering or ME-SVF, predicts the motion vector for the n -to-1 reduced video by applying a spatial varying filter to the motion vector of the original high-resolution video.

In a way, ME-SVF improves on the "target", which can be the MEAN or AMVR. Let the "target" be MEAN. While the target may not be the best, it is a reasonable value. Among the n candidate motion vectors, if a vector is close to the target, it is probably quite reliable and thus should be given a larger weight. On the other hand, if a candidate vector is far away from the target, it is probably quite unreliable and thus should be given a lower weight. It is with this observation that the spatial filter is created.

The new motion vector V_r is obtained with the following spatial-variant filter equation

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

which is essentially a weighted average of the n candidate motion vectors V_i , for $i = 1, 2, \dots, n$, with spatially varying weights D_i for $i = 1, 2, \dots, n$. The weight D_i is the mapped value of the difference between the target and V_i .

We observe that the MEAN can sometimes be a very good estimate if the n motion vectors are similar. In this case, the V_r from ME-SVF will be very similar to MEAN because all candidate motion vectors will have similar weights. The MEAN can be poor and meaningless if the n motion vectors have different directions and/or magnitude. In this case, the V_r from ME-SVF will give a larger weight to any candidate vector that is close to the target vector, and smaller weight to those far away from the target. This will selectively suppress the bad outliers, and give more weight to the more reasonable candidates. It will tend to give more weights to the candidate vectors that are clustered together.

Due to limited time, we do not have time to try all possible targets and maps. In our preliminary experiment, we used MEAN as target and tried some reasonable maps for the weighting function. The weighting function used is

$$D_i = \begin{cases} 1 & \text{if } d_i < C_1, \\ 1 - (d_i - C_1) / (C_2 - C_1) & \text{if } C_1 < d_i < C_2, \\ 0 & \text{otherwise.} \end{cases}$$

where

$$d_i = (V_{i,x-dir} - V_{m,x-dir})^2 + (V_{i,y-dir} - V_{m,y-dir})^2$$

is the square Euclidean distance between V_i and V_m , and C_1 and C_2 are predefined constant. We called it ME-SVF(MEAN, $D(C_1, C_2)$)

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 structure for the 9-to-1 and 16-to-1 down-conversion. 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. 1 shows the PSNR difference between ME-SVF and AMVR 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 'football', ME-SVF is 0.2536dB higher in PSNR than AMVR. For 'salesman', ME-SVF is 1.3128dB higher than AMVR.

5. CONCLUSIONS

In this paper, we propose a modified algorithm of 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 very low and with only 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,' to appear in *Proc. 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. IEEE Int. Conf. Image Processing*, Oct. 1997.

Fig.1 the PSNR difference between ME-SVF and AMVR for the “football” sequence.

Football					Salesman

	9-to-1		16-to-1			9-to-1		16-to-1	
	Mean PSNR (dB)	Average Search points	Mean PSNR (dB)	Average Search points		Mean PSNR (dB)	Average Search points	Mean PSNR (dB)	Average Search points
Full Search	26.4692	225	26.0977	121	Full Search	40.1525	225	40.2353	121
Mean Only	25.2105	0	25.3247	0	Mean Only	39.2961	0	40.1668	0
AMVR	25.2945	0	25.4005	0	AMVR	38.7703	0	40.1340	0
ME-SVF (MEAN, D(5,195))	25.5481	0	25.5102	0	ME-SVF (MEAN, D(5,30))	40.0831	0	40.2100	0

Table 1. mean PSNR and average search points for various algorithms and various test sequences.