LETTER

Noise Robust Motion Refinement for Motion Compensated Noise Reduction

Jong-Sun KIM[†a)], ***Student Member and*** **Lee-Sup KIM**[†], ***Nonmember***

SUMMARY A motion refinement algorithm is proposed to enhance motion compensated noise reduction (MCNR) efficiency. Instead of the vector with minimum distortion, the vector with minimum distance from motion vectors of neighboring blocks is selected as the best motion vector among vectors which have distortion values within the range set by noise level. This motion refinement finds more accurate motion vectors in the noisy sequences. The MCNR with the proposed algorithm maintains the details of an image sequence very well without blurring and joggling. And it achieves 10% bit-usage reduction or 0.5 dB objective quality enhancement in subsequent video coding.

key words: *motion refinement, noise reduction, motion estimation, motion compensated noise reduction*

1. Introduction

Noise reduction not only improves the visual quality but also increases the performance of subsequent processing tasks such as coding, analysis and interpolation. A through review of noise reduction algorithms for dynamic image sequences is presented in [1]. To take full advantage of the temporal correlations that exist in an image sequence, explicit motion estimation and compensation has been used as a temporal preprocessing step in noise reduction algorithms. The performance of these motion compensated noise reduction (MCNR) algorithms clearly depends on the accuracy of the motion estimator. However, due to the presence of noise, it is difficult to find accurate motion vectors. Conventional MCNR algorithms focused on the problem of finding optimal filters and did not consider the motion estimator. Though sub-pixel motion estimation has been used for MCNR in [2] to increase motion accuracy, and noise-robust motion estimation has been proposed in [3], they still gave no attention to actual noise. The penalty of mean absolute difference has been proposed for noise robustness in [4]. However, it also lacked noise level adaptability. If there is more noise, the ambiguity of motion is increased, and this becomes severe when the motion is small. Temporal filtering under this condition causes the artifacts such as blurring and joggling. To remove these artifacts, we propose a noise robust motion refinement algorithm with noise level adaptability for MCNR.

Manuscript received April 9, 2007.
Manuscript revised August 24, 2007.

[†]The authors are with Korea Advanced Institute of Science and Technology (KAIST), Korea.
a) E-mail: jskim@mvlsi.kaist.ac.kr

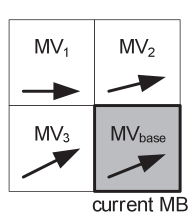
DOI: 10.1093/ietisy/e91–d.5.1581

2. Binary Motion Estimation with Hybrid Distortion Measure

Due to the presence of noise, the vector with minimum distortion such as minimum sum of absolute difference might not be optimal motion vector. As the noise level increases, the uncertainty of optimal motion vector also increases. Thus, according to the noise level of image sequence, it is necessary to adjust the distortion range within which the best motion vector can be found. Among search points which have distortion values within this range, we choose the nearest point to the base point as the best motion vector. The base point is pointed by the base vector (MV_{base}) which is computed by the weighted average of motion vectors of neighboring blocks. This is based on the fact that the motion vector of a block has a tendency to have similar values to those of neighboring blocks. For example, the base vector MV_{base} is calculated as shown Fig. 1. In Fig. 1, MV_1, MV_2 and MV_3 are motion vectors of left-upper, upper and left block respectively and a, b and c are the weight for each motion vector.

The motion refinement scheme with distortion measure based on sum of absolute difference (SAD) is shown in Fig. 2. First, the SAD for each search point is calculated and the distance from the base point to each search point is recorded. Finding the minimum SAD, the allowable distortion range for the best motion vector is calculated. Then we choose the first point which has an allowable SAD as the best motion vector in the list in which search points are placed in the order of the closeness of the base point.

In Fig. 2, the allowable distortion range for the best motion vector of a macroblock in the kth frame varies according to the noise level of the kth frame with the proportional coefficient α. For real time processing, the noise level of the kth frame NL_k is predicted by the weighted average of noise levels of m past frames as follows:



$$MV_{base,x} = \frac{a \times MV_{1,x} + b \times MV_{2,x} + c \times MV_{3,x}}{a+b+c}$$

$$MV_{base,y} = \frac{a \times MV_{1,y} + b \times MV_{2,y} + c \times MV_{3,y}}{a+b+c}$$

Fig. 1 Base vector MV_{base} computation with motion vectors of neighboring blocks.

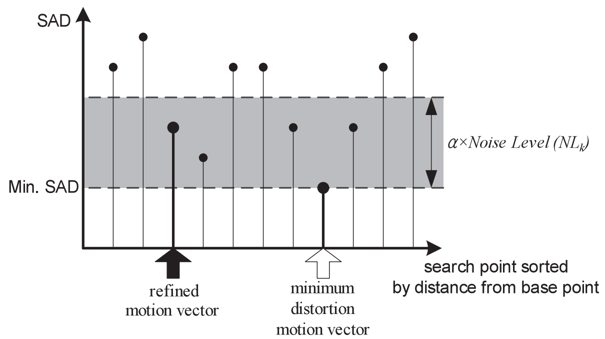


Fig. 2 Motion refinement scheme with sum of absolute difference (SAD) based distortion measure.

$$NL_k = \frac{\sum_{l=1}^{m}(W_l \times NL_{k-l})}{\sum_{l=1}^{m} W_l} \tag{1}$$

where W_l is the weight for the noise level of the $(k-l)$th frame. According to noise estimation method and block size, estimated noise level has to be properly scaled to be used as distortion. The proportional coefficient α in Fig. 2 is computed as follows:

$$\alpha = \begin{cases} 1 - \dfrac{L(MV_{base})}{MV_{thd}}, & \text{if } (L(MV_{base}) < MV_{thd}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $L(MV_{base})$, vector length of MV_{base}, is the sum of x-component and y-component of MV_{base} and MV_{thd} is a threshold value of vector length. When the motion is large, we give less importance to the motion of neighboring blocks by introducing α in order to reduce the artifact appeared in motion boundary area. It prevents the static object from following the moving object of neighboring blocks. And the human eye system is more sensitive to static or small motion area, reduction of the importance to fast motion area does not affect on visual quality. When the motion is small, the ambiguity caused by noise is increased. Thus, it is desirable to follow the motion of neighboring blocks.

3. Experimental Result

To show the performance of proposed algorithm, we simulated 1-D temporal filter for MCNR using motion estimator with and without motion refinement. We used the noise estimator based on [5]. The input sequences with CCIR601 format were corrupted with the Gaussian noise of various values of standard deviation σ = 6, 8, 10 and 12. Each sequence consisted of 80 frames. And the following parameter values were used in simulation: search range [±16, ±8], neighboring motion vector weight $a = \sqrt{2}$ and $b = c = 1$, the number of previous frames for noise level prediction $m = 3$, noise level weight $W_l = 1$ for all l, vector length threshold $MV_{thd} = 8$.

As shown in Table 1, the proposed motion refinement algorithm was very effective for MCNR and gave better performance as the noise level was increased. We achieved noticeable video quality enhancement subjectively as well as objectively. The MCNR with the proposed motion refinement kept up the details of an image sequence very well without blurring and removed the joggles which were appeared in the sequence filtered without motion refinement. Figure 3 shows the effect of the proposed algorithm on subsequent processing task, video coding [6], using fixed quantization parameter values of 10, 12, 14, 18 and 22 and constant bitrates of 1.5~3.5 Mbps for Actress sequence corrupted with σ = 6. As expected, the sequence filtered using the proposed motion refinement has a higher PSNR curve. The proposed algorithm achieved about 10% bit-usage reduction and about 0.5 dB objective quality enhancement in subsequent video coding.

Table 1 Performance comparison of 1-D temporal filtering for MCNR with motion refinement versus without motion refinement.

Sequence	σ	Y_PSNR (dB)		ΔY_PSNR (dB)
		without motion refinement	with motion refinement	
Actress	12	32.70	33.48	0.78
	10	34.29	35.02	0.73
	8	35.99	36.63	0.64
	6	38.10	38.60	0.50
Flag	12	31.93	32.52	0.59
	10	33.50	34.06	0.56
	8	35.23	35.70	0.47
	6	37.36	37.73	0.37
Bugs Life	12	31.21	31.60	0.39
	10	32.71	33.06	0.35
	8	34.32	34.62	0.30
	6	36.40	36.64	0.24
Bean	12	31.41	31.75	0.34
	10	32.84	33.14	0.30
	8	34.44	34.68	0.24
	6	36.48	36.67	0.19

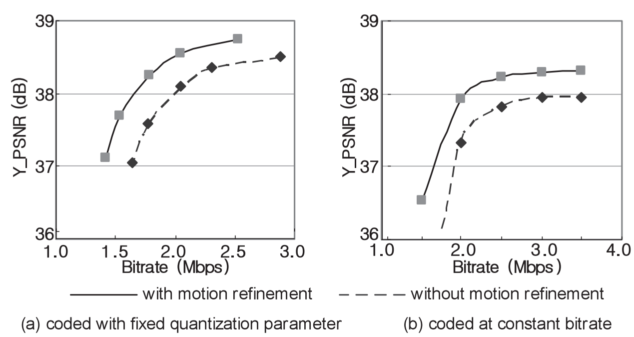


Fig. 3 Rate-distortion curve using PSNR for Actress sequence corrupted with σ = 6 and processed by MCNR.

4. Conclusion

The proposed motion refinement algorithm finds optimal motion vector in noisy sequence using spatio-temporal correlations with noise level adaptability. This noise robust motion refinement scheme is applicable to various MCNR algorithms as a temporal preprocessing and enhances noise reduction efficiency greatly because the performance of MCNR depends on the accuracy of the motion estimator. We applied the proposed algorithm to pre-processor for video encoder designed for DVD recorder SOC [7] and obtained noticeable visual quality enhancement.

References

[1] J.C. Brailean, R.P. Kleihorst, S. Efsrtatiadis, A.K. Katsaggelos, and R.L. Lagendijk, "Noise reduction filters for dynamic image sequences: A review," Proc. IEEE, vol.83, no.9, pp.1272–1292, Sept. 1995.

[2] S. Olivieri, L. Albani, and G. de Haan, "Noise-robust recursive motion estimation for H.263-based videoconferencing system," Multimedia Signal Processing, 1999 IEEE 3rd Workshop on, vol.2, pp.345–350, Sept. 1999.

[3] H.-Y. Cheong, A.M. Tourapis, J. Llach, and J. Boyce, "Adaptive spatio-temporal filtering for video denosing," Image Processing, 2004. ICIP'04. 2004 International Conference on, vol.2, pp.965–968, Oct. 2004.

[4] V. Zlokolica, A. Pizurica, and W. Philips, "Recursive temporal denoising and motion estimation of video," Image Processing, 2004. ICIP'04. 2004 International Conference on, vol.3, pp.1465–1468, Oct. 2004.

[5] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," Vision, Image and Signal Processing, IEE Proc., vol.146, no.2, pp.80–84, April 1999.

[6] "Test model 5," ISO/IEC JTC1/SC29/WG11/No400, MPEG93/457, April 2003.

[7] S5L3210 User's Manual, Samsung Electronics Co., Ltd, 2006.