A NEW BLOCK MOTION VECTOR ESTIMATION USING ADAPTIVE PIXEL DECIMATION

Yui-Lam CHAN and Wan-Chi SIU

Department of Electronic Engineering,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
Tel : (852)-27666229 Fax : (852)-23628439

ABSTRACT

A new adaptive technique based on pixel decimation for estimating motion vector is presented. In traditional approach, a uniform pixel decimation is used. Since some pixels in each block do not enter into the matching criterion, this approach might limit the accuracy of the motion vector. In this paper, we select the most representative pixels based on image content in each block for the matching criterion. This is due to the fact that high activity in the luminance signal such as edges and texture contributes mainly to the matching criterion. Our approach can compensate the drawback in standard pixel decimation techniques. Computer simulations show that this technique is close to the performance of the exhaustive search with a significant reduction on computational complexity.

1. INTRODUCTION

Block motion estimation is being widely used[1-5] in video coding. Many fast and efficient algorithms[2-7] have been developed to reduce the computational complexity. But these methods have the undesirable problem of local minimum. Instead of limiting the number of locations to be searched, Koga et al. [8] subsampled the pixel block so as to reduce the computational complexity. Bierling [9] introduced a hierarchical motion vector estimation in which a first approximation of motion vector is obtained from a low-pass-filtered and subsampled image. However, since only a uniform fraction of the pixels enters into the matching computation, the use of these standard subsampling techniques can seriously affect the accuracy of motion vector detection. Thus, Liu and Zaccarin [10] have successfully used an alternating pixel decimation pattern. The subsampling patterns are alternated over the locations searched so that all pixels of a block contribute to the computation of the motion vector. In fact, high activity in the luminance signal such as edges and texture mainly contributes to the matching criterion. In this paper, some most representative pixels are selected instead of uniform subsampling. We use the relationships between a pixel and it neighbors, thus we can just employ those 'main pixels' to represent all others. Furthermore the prediction error compared with the uniform subsampling is significantly reduced by a proper selection of a subset of the pixels. The result is very close to the exhaustive search without pixel decimation. Furthermore, the performance of our algorithm is better than that of Liu and Zaccarin's [10] algorithm.

This paper describes the new block motion vector estimation and also presents the results obtained by our proposed adaptive pixel decimation. Section 2 describes the proposed block motion estimation. The computational complexity of our proposed algorithm is presented in section 3. Experimental results obtained by computer simulations are discussed in section 4. Finally, a conclusion is given in section 5.

Acknowledgement : This work was supported by The Croucher Foundation PolyU340/055.

2. THE NEW PROPOSED ADAPTIVE PIXEL DECIMATION

In uniform pixel decimation technique as depicted in figure 1, since 3/4 of the pixels in each block do not enter into matching computation regularly, it will limit the accuracy of the motion vector. This approach could possibly be able to obtain a good estimation of motion vector when the intensity of the block is nearly uniform. However, in the case of high activity blocks, some details may be neglected. Thus, it probably would introduce excessive prediction error. This paper is based on the fact that high activity in the spatial domain such as edges and texture contributes mainly to the mean absolute difference (MAD) matching criterion. We can vary the number of selected pixels based on the image

0-7803-2431-5/95 $4.00 © 1995 IEEE

Table 1: Definition of each region

K	I_K	h	k	Remark
1	$I(0,0)$	1	$0,1$	
$2,3$	$I(0,3), I(0,6)$	$0,1$	$0,\pm1$	$(h,k) \neq (0,0)$
$4,7$	$I(3,0), I(6,0)$	$0,\pm1$	$0,1$	$(h,k) \neq (0,0)$
$5,6$	$I(3,3), I(3,6)$	$0,\pm1$	$0,\pm1$	$(h,k) \neq (0,0)$
$8,9$	$I(6,3), I(6,6)$			

details. In other words, we can use less pixels when the block has uniform intensity. But in the high activity block, more pixels can be employed for the MAD matching criterion. This adaptive approach can reduce the prediction error compared with the standard pixel decimation. In our algorithm, we use the relationship between a pixel and its neighbors to select the most representative pixels. The procedure for selecting pixels can be described as follows:

1. Initially, nine pixels are selected as shown in figure 2a. The intensities of the selected pixels are $I(3i, 3j)$, for $i, j = 0, 1, 2$.

2. The 8×8 pixel block is divided into 9 regions, depicted in figure 2b, and each region has its corresponding central pixel, I_K, the selected pixels in step(1), where K is the region number.

3. In each region, the difference between the central pixel, I_K , and one of its neighbor pixels (the dotted line pixel as shown in figure 2b) is defined as
$$D_K(h,k) = |I_K(h,k) - I_K|$$
where (h,k) is the location of the neighbor pixel in region K, with (h,k) as the displacements from the central pixel. Table 1 gives the definition for each region.

4. In each region, $D_K(h,k)'s$ are arranged in descending order.

5. If the maximum value of $D_K(h,k)$ is greater than the threshold value, T, this pixel is selected and the pixel selection proceeds to next step. Otherwise, the pixel selection is stopped.

6. If, the next maximum value of $D_K(h,k)$ is greater than T, the pixel selection proceeds to step(7). Otherwise, the pixel selection is stopped.

7. If, its neighbor pixels, except central pixel, have not already been selected within each region, this pixel is selected. Otherwise, the difference of its intensity between this pixel and the already selected neighbor pixel is checked. If the difference is greater than T, this pixel is selected. The pixel selection can then proceed back to step(6).

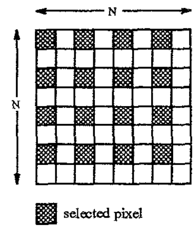

Figure 1: The selected pixels of 4 to 1 subsampling

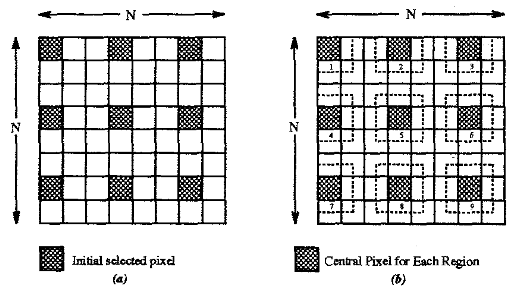

Figure 2: Adaptive pixel selection (a) 9 selected pixels. (b) the selected pixels in (a) are considered as the central pixel for each region, the dotted lines indicate the neighbor pixels for respective central pixels in each region.

The n motion vectors which have the minimum MAD obtained from above are selected. Then, we compute the MAD matching criterion for each of the n motion vectors using all pixels. The one that has the minimum MAD among n motion vectors is selected as the final motion vector. This method can reduce the possibility of local minimum occured.

3. COMPUTATIONAL COMPLEXITY

The computational complexity of the pixel decimation algorithm is a function of the number of locations searched (S), the number of operations per pixel needed to compute the MAD matching criterion (K), and the number of selected pixels. If the pixel decimation technique is not used, a total of SKN^2 operations is required to estimate the motion vector of a block size of $N \times N$. If the standard 4 to 1 pixel decimation is employed, the number of operations will be $SK(N^2/4)$.

In our proposed algorithm, we first select the pixel pattern, but this operation is negligible as compared with the computing time of the MAD matching criterion. Different blocks have different numbers of selected pixels in the proposed algorithm. For the uniform intensity block, the number of selected pixels may be 9 (less than $N^2/4$). But, for very high detail blocks, the number of selected pixels may be greater than $N^2/4$. So, the number of operations is different for a different block. Let $H(i,j)$ be the number of selected pixels at block location (i,j) in the frame. The average number of operations of the adaptive pixel decimation per block (F) is obtained as:

$$F = \frac{SKN^2}{PL} \sum_{i=0}^{\frac{P}{N}-1} \sum_{j=0}^{\frac{L}{N}-1} H(i,j)$$

where P is the number of pixels per line and L is the number of lines in the each frame.

In this approach, the number of operations(F) also depends on the Threshold(T) selection. If T is decreased, the number of operations will be increased and the performance is also better. If the refinement of motion vector is used, an extra number, $nK(N^2 - H(i,j))$, of operations per block is needed. Since $nK(N^2 - H(i,j)) \ll F$, this overhead is negligible.

4. RESULTS

A series of computer simulations have been conducted to evaluate the performance of the adaptive pixel decimation technique. The image sequences " football" and "tennis" have been used. A maximum allowable displacement in the x and y directions is 8 with block size of 8x8. We compare the algorithms using the prediction error(MSE) of the motion-compensated frames.

In the following comparison, the threshold value (T) is chosen such that the computational complexity of our proposed algorithm is the same as standard 4 to 1 pixel decimation [8]. In other words, the computation is reduced by a factor of 4 as compared with the exhaustive search without pixel decimation. The number of motion vectors(n) selected in the first stage is chosen as 4. The prediction errors (MSE) of a standard 4 to 1 pixel decimation [8] and that of the proposed algorithm are shown in figures 3 and 4. It is seen that the performance of our proposed algorithm is significantly better than that of the standard 4 to 1 pixel decimation.

Figures 5 and 6 show the MSE difference of the exhaustive search without pixel decimation and our proposed adaptive pixel decimation algorithm. The MSE difference is very small especially before the 70th frame in "football" sequence. The proposed algorithm is even better in some of frames(the negative value of MSE difference). It shows that the proposed algorithm can get the true motion vectors, so the predicted frame is more smooth and this predicted frame produces much better quality output for using it as the reference frame for the next frame. The MSE difference of the exhaustive search without pixel decimation and the Liu and Zaccarin [10]'s pixel decimation using alternating subsampling patterns is also shown in figures 5 and 6, all the frames from proposed algorithm are better than that of Liu and Zaccarin [10]'s approach. The results show that the proposed adaptive pixel decimation algorithm is very effective. It is very suitable for image sequences which contain edge objects moving in still and smooth background. It is because the number of pixels used in the MAD criterion depends on image content. The block which contains edge and texture often leads to large MSE. But, our adaptive approach uses more pixels to reduce the MSE in these block and employs less pixels in the smooth block to reduce the computation burden.

5. CONCLUSION

A new fast block matching algorithm is proposed to compensate the drawback in uniform pixel decimation technique. This proposed algorithm uses the relationships between a pixel and its neighbors, the most representative pixels are used as the matching criterion. This proposed algorithm can reduce the heavy computational burden of the exhaustive search without significantly increasing the prediction error of the motion-compensated frames. This new fast block matching algorithm is significantly better than that of the standard pixel decimation, and shows improvement compared to the famous approach given by Liu and Zaccarin [10]. This approach is certainly an efficient technique for block matching motion vector estimation.

6. REFERENCES

[1] N.G.Musmann, P. Pirch, and H.J. Grallat, "Advances in Picture Coding", Proc. of the IEEE, vol.73, No.4, pp.523-548,Apr.1985.

[2] M. Ghanbari, "The Cross-search Algorithm for Motion Estimation", IEEE Trans. on Commun., vol.33, No.9, pp.1011-1015, Sept. 1985.

[3] J.R.Jain and A.K.Jain, "Displacement Measurement and It's Applications in Interframe Coding", IEEE Trans. on Commun., vol.29, pp.1799-1808, July 1981.

[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing", in Proc. Nat. Telecommun. Conf., pp.G.5.3.1-5.3.5, Nov.29-Dec.3, 1981.

[5] R. Srinivasan and K.R.Rao, "Predictive Coding Based on Efficient Motion Estimation", IEEE Trans. Commun., vol.33, pp. 1011-1014, Sept. 1985.

[6] K.R.Rao and P. Yip, Discrete Cosine Transform - Algorithms, Advantages and Applications. New York : Academic Press, 1990.

[7] G. Kummerfeldt, F. May and W. Wolf, "Coding televisioin signals at 320 and 64kbit/s", 2nd Internat. Tech. Symp. on Optical and Electro-Optical Applied Science and Engineering, Cannes, December 1985.

[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conferencing", in Proc. Nat. Telecommun. Conf., Nov.29-Dec.3, 1981, pp.G.5.3.1-5.3.5

[9] M. Bierling, "Displacement Estimation by Hierarchical Block Matching", in SPIE, Visual Commun. Image Processing'88, vol.1001, pp. 942-951, 1988.

[10] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors", IEEE trans. on Circuits and Systems for Video Technology, Vol.3, No.2, April 1993.

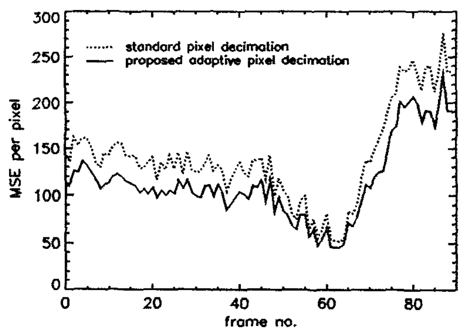

Figure 3: MSE produced by algorithm with standard 4 to 1 pixel decimation and the proposed algorithm in "football" sequence

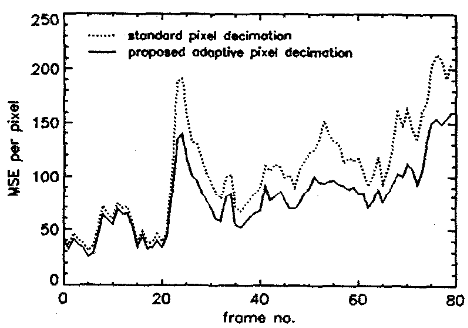

Figure 4: MSE produced by algorithm with standard 4 to 1 pixel decimation and the proposed algorithm in "tennis" sequence

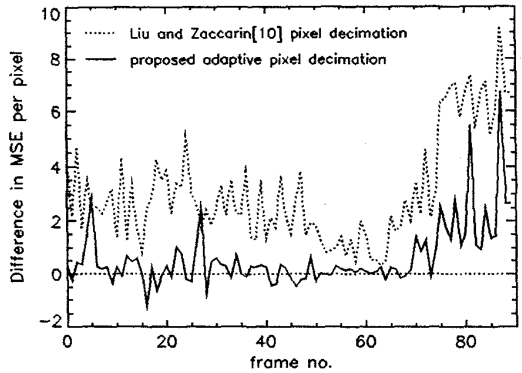

Figure 5: MSE difference of the exhaustive search without pixel decimation and the proposed adaptive pixel decimation, and that of Liu and Zaccarin[10]'s method in "football" sequence

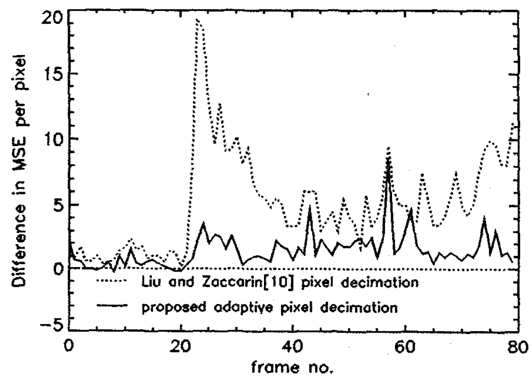

Figure 6: MSE difference of the exhaustive search without pixel decimation and the proposed adaptive pixel decimation, and that of Liu and Zaccarin[10]'s method in "tennis" sequence