those of MAP detection, as shown in Fig. 8. We thus conclude that BMDA works quite well for different codebook sizes, as long as the image is not too coarsely quantized.

3) Dependence on Image Characteristics: To examine the dependency of BMDA on image characteristics, the 512 × 512 "peppers" image was selected. For the sake of simplicity, and considering that only a rough illustration is needed, the "peppers" image is used as both the training image and the test image, although it is not realistic to do so in practical vector quantizer design. The block size is 4 × 4 and the codebook size is 512. Fig. 9 shows that BMDA also performs quite well on this image.

Note that the same threshold parameters have been used for different channel error rates, different block sizes, different codebook sizes and different test images. The results are all quite good. This suggests that the BMDA detection scheme is very robust and easy to implement.

V. Conclusion

A boundary-matching-based detection and correction scheme, referred to as the BMDA, is proposed here for the recovery of erroneously received image vectors over noisy channels. A special way to organize a full search VQ codebook for progressive transmission is also introduced to facilitate the implementation of the proposed BMDA. Simulation results show that the proposed scheme provides significantly better performance than an MAP-based scheme over all channel error rates. It is also shown that this detection scheme is quite robust with respect to the variations of channel error rates, block size, and codebook size. Future work will include the addition of channel coding (for example, unequal error protection) to the system.

References

[1] D. Comstock and J. D. Gibson, "Hamming coding of DCT compressed images over noisy channels," *IEEE Trans. Commun.*, vol. COM-32, pp. 856–861, July 1984.
[2] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 6, pp. 827–838, Nov. 1987.
[3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Amsterdam, The Netherlands: Kluwer, 1992.
[4] S. C. Huang and Y. F. Huang, "Principal component vector quantization," *J. Visual Commun. Image Representation*, vol. 4, no. 2, pp. 112–120, June 1993.
[5] T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 170–185, Apr. 1992.
[6] W. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 5, pp. 417–420, 1993.
[7] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[8] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov source over discrete memoryless channels - apps. to combined source-channel coding," Tech. Rep. 92-25, Univ. of Maryland, College Park, MD, 1992.
[9] E. A. Riskin, R. Ladner, R.-Y. Wang, and L. E. Atlas, "Index assignment for progressive transmission of full search vector quantization," *IEEE Trans. Image Processing*, pp. 307–312, May 1994.
[10] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 838–846, June 1991.
[11] N. Tanabe and N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," *IEEE J. Select. Areas Commun.*, vol. 10, no. 5, pp. 926–943, June 1992.
[12] R. Wang, E. Riskin, and R. Ladner, "Codebook organization to enhance maximum a posteriori detection of progressive transmission of vector quantized images over noisy channels," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 233–236, vol. 5, 1993, also to appear in *IEEE Trans. Image Processing*.
[13] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Trans. Commun.*, vol. 38, no. 12, pp. 2147–2158, Dec. 1990.
[14] W. J. Zeng, "Vector quantization for image coding," Master thesis, Dept. of Elec. Eng., Univ. of Notre Dame, Notre Dame, IN 46556, Aug. 1993.

New Adaptive Pixel Decimation for Block Motion Vector Estimation

Yui-Lam Chan and Wan-Chi Siu

***Abstract*— A new adaptive technique based on pixel decimation for the estimation of motion vector is presented. In a traditional approach, a uniform pixel decimation is used. Since part of the pixels in each block do not enter into the matching criterion, this approach limits 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 mainly contributes 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 significant computational reduction.**

I. Introduction

In video coding applications such as high-definition television (HDTV), video conferencing, etc., block motion estimation is being widely used [1]–[5]. This approach assumes that the motion field over blocks of pixels is treated as a unit and moved as a group. Besides, the block motion can be modeled as purely translation.

For block matching, the present frame is divided into two-dimensional small blocks of $N \times N$ pixels. Each block in the current frame estimates its motion vector by evaluating some matching criteria over the blocks in the previous frame and selecting the block which yields the closest matching. There are many choices [6] for the matching criterion, e.g., mean square error (MSE), mean absolute difference (MAD), etc. Among these criteria, the MAD is the most popular one because it does not require any multiplication and produces similar performance as the MSE. The MAD matching criterion for $N \times N$ block is given by

$$\mathrm{MAD}(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_n(i, j) - I_{n-1}(i+x, j+y)|$$
$$-D \le x, y \le D$$

where $I_n(i, j)$ is the intensity of the pixel at location (i, j) within the block in the nth frame and the displacement is (x, y). The motion vector is $\arg_{(x, y)} \min \mathrm{MAD}(x, y)$. D is the maximum possible displacement in the motion vector (x, y).

Exhaustive search block matching evaluates the MAD at all possible displacements, $[(2D+1)^2]$, in the search window to find the optimal motion vector. This method is computationally very

Manuscript received August 10, 1994; revised March 13, 1995. This paper was recommended by Associate Editor Kou-Hu Tzou. This work was supported by the Croucher Foundation PolyU340/055.
The authors are with the Department of Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.
Publisher Item Identifier S 1051-8215(96)01321-3.

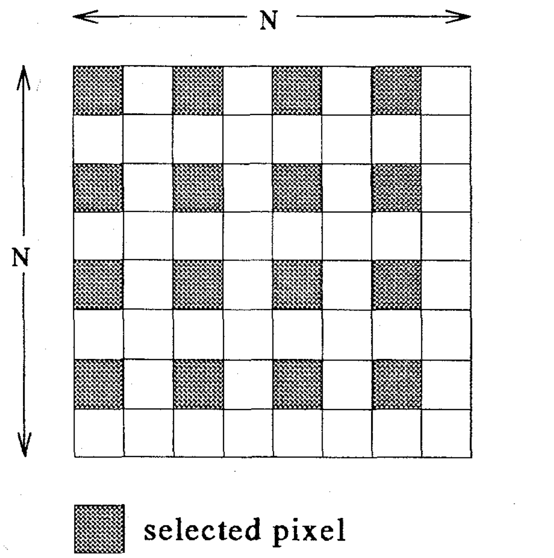

Fig. 1. The selected pixels of 4 to 1 subsampling.

TABLE I
DEFINITIONS OF EACH REGION

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,±1	(h,k)≠(0,0)
4,7	I(3,0), I(6,0)	0,±1	0,1	(h,k)≠(0,0)
5,6,8,9	I(3,3), I(3,6) I(6,3), I(6,6)	0,±1	0,±1	(h,k)≠(0,0)

expensive and thus efficient algorithms such as the 2-D-logarithmic search [3], three step search [4], conjugate directional search [5], etc. [6], 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.* [4] subsamples the pixel block so as to reduce the computational complexity. Kummerfeldt *et al.* [7] uses a field subsampling by a factor of 4 : 1 combined with hybrid coding algorithm. Bierling [8], [9] introduced a hierarchical motion vector estimation in which a first approximation of motion vector is obtained from 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] uses the 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, most representative pixels are selected instead of uniform subsampling. We use the relationships between a pixel and its 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 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.

II. THE NEW PROPOSED ADAPTIVE PIXEL DECIMATION

In the traditional approach, when matching a block in the current frame to a block in a previous frame is considered, every pixel of

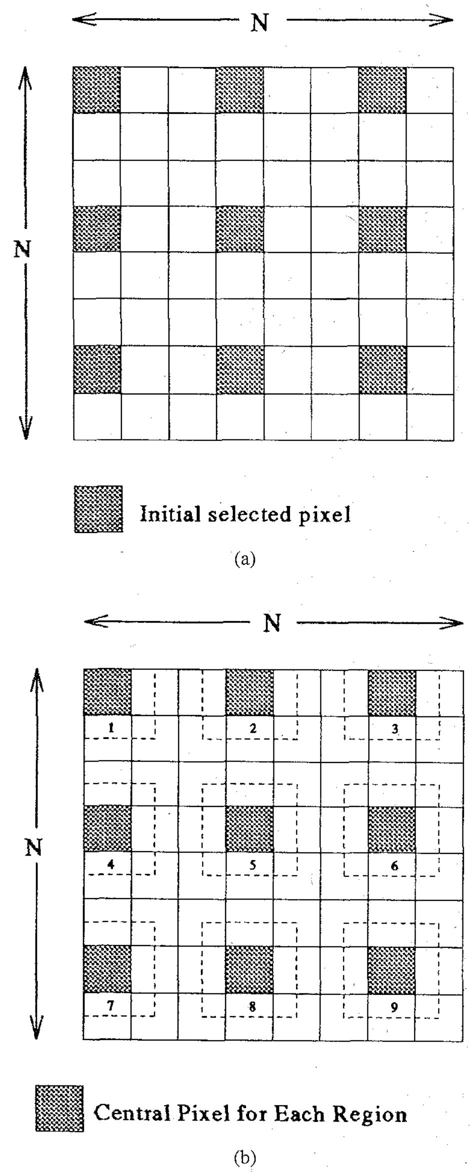

Fig. 2. Adaptive pixel selection (a) nine selected pixels. (b) The selected pixels in (a) are considered as the central pixel for each region, the dotted lines indicate the neighbor pixels of respective central pixels in each region.

the block is employed as the matching criterion. If the block is selected as 8 × 8, 64 points need to be compared in the MAD matching criterion. This seems to be too heavy and unnecessary. Based on the assumption that all pixels within each block move by the same amount, there is still a chance that it will not be the true motion vector when the MAD gets its minimum. In other words, if $\text{MAD}_{\min} = \text{MAD}(x_0, y_0) \neq 0$, there is a chance that although $\text{MAD}(x_1, y_1) > \text{MAD}(x_0, y_0)$, but (x_1, y_1) might still be the true motion vector. The larger the $\text{MAD}_{\min}$ is, the bigger the chance of this situation happening. That is, when more pixels are employed, the greater error might possibly occur because of the effect of distortion accumulation. If less pixels are used, the value of $\text{MAD}_{\min}$ will get smaller in the same case, hence the chance just mentioned will also be smaller. Koga *et al.* [4] proposed subsampling the original image sequence by a factor of two both horizontally and vertically as shown in Fig. 1. This approach can reduce the computation by a factor of

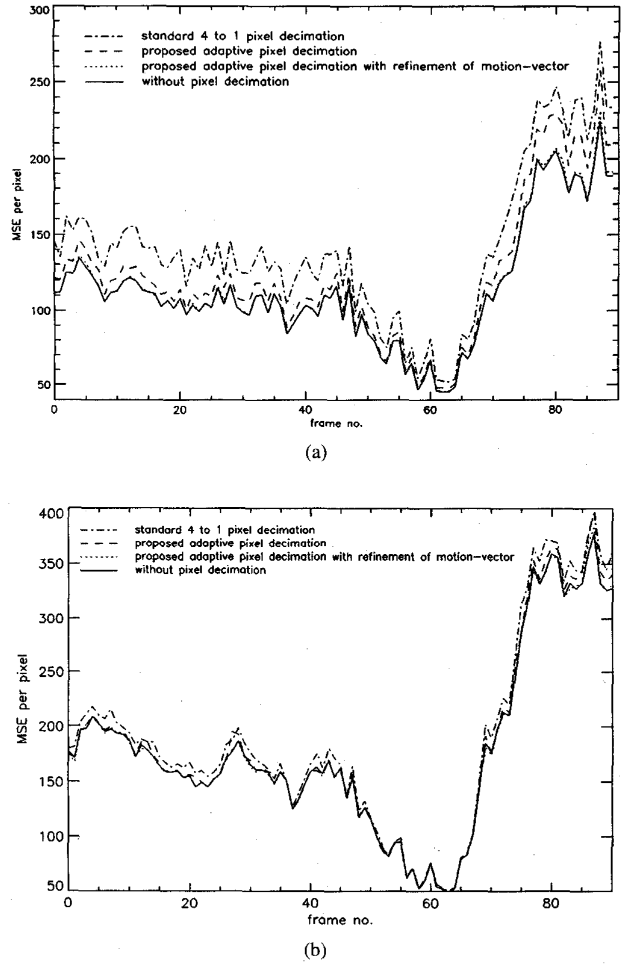

Fig. 3. MSE produced by algorithm with no pixel decimation, standard 4 to 1 pixel decimation and the proposed algorithm for "football" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

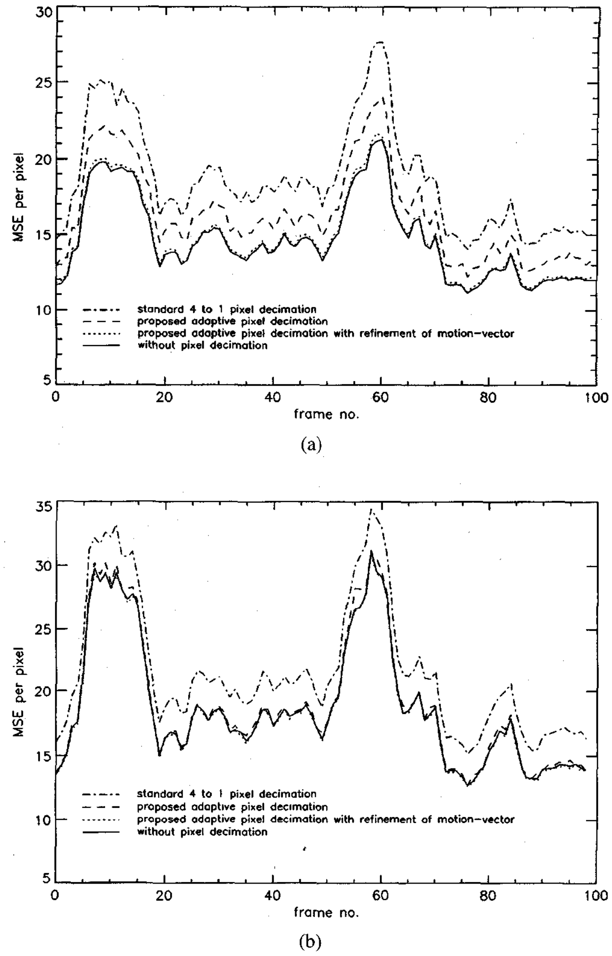

Fig. 4. MSE produced by algorithm with no pixel decimation, standard 4 to 1 pixel decimation and the proposed algorithm for "salesman" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

four. But, in this uniform pixel decimation technique, 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 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 spatial domain such as edges and texture mainly contributes to the MAD criterion. We can vary the number of selected pixels based on the image details. In other words, we can use fewer 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 standard pixel decimation.

A. Description of the Proposed Algorithm

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 Fig. 2(a). The selected pixels are $I(3i, 3j)$, for $i, j = 0, 1, 2$.
2) The 8 × 8 pixel block is divided into nine regions, depicted in Fig. 2(b), 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 central pixel, I_K, and one of its neighbor pixels [the dotted line pixel as shown in Fig. 2(b)] 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 I gives the definitions for each region.
4) In each region, $D_K(h, k)$ 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 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).

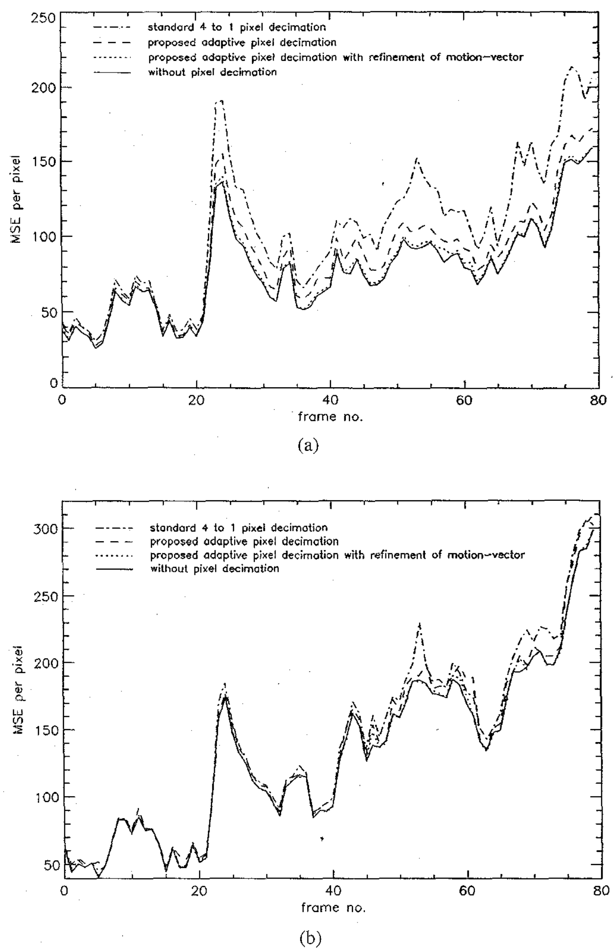

Fig. 5. MSE produced by algorithm with no pixel decimation, standard 4 to 1 pixel decimation and the proposed algorithm for "tennis" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

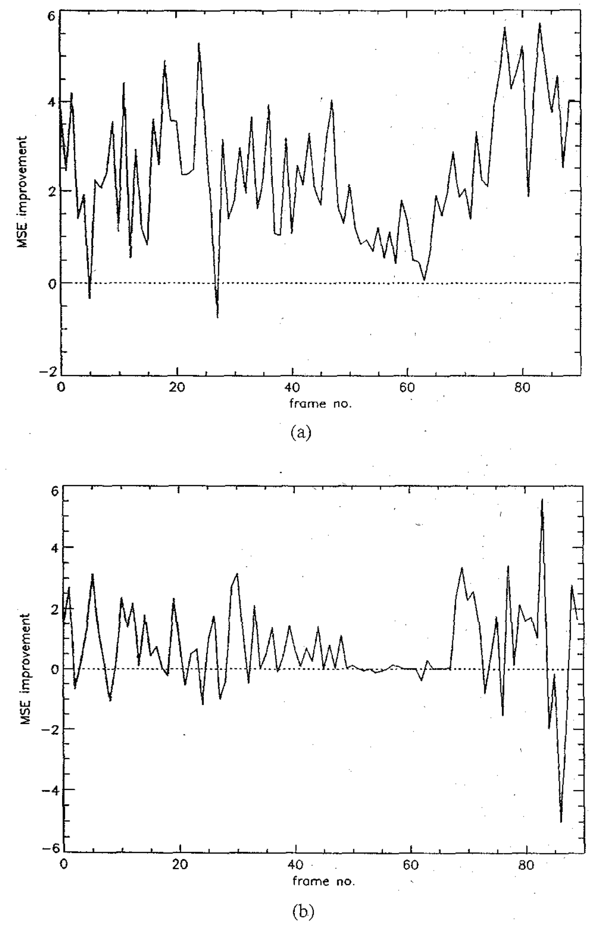

Fig. 6. Improvement of the proposed adaptive pixel decimation as compared with Liu and Zaccarin's [10] method for "football" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

We have used block size of 8 × 8 as an example for the description of the proposed algorithm, however, the extension of the proposed scheme to a large block size, say 16 × 16, is straightforward.

B. Refinement of the Motion Vector

The motion vector which is obtained from Section II-A using exhaustive search prevents fault estimation due to local minimum problem. Though the selected pixels are the most representative in the block, not all pixels enter into the matching criterion computation. The motion vector obtained can be refined as follows.

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 approach can significantly reduce the possibility of forming local minima.

III. 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 are 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. There are about 52–105 subtractions for pixel pattern selection of each 8 × 8 block, but these numbers of operations are negligible as compared to the computation required for the MAD matching criterion (for which 127 additions/subtractions are required for each MAD computation).

Different blocks have different numbers of selected pixels in the proposed algorithm. For the uniform intensity block, the number of selected pixels may be nine (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 different blocks. 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

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

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

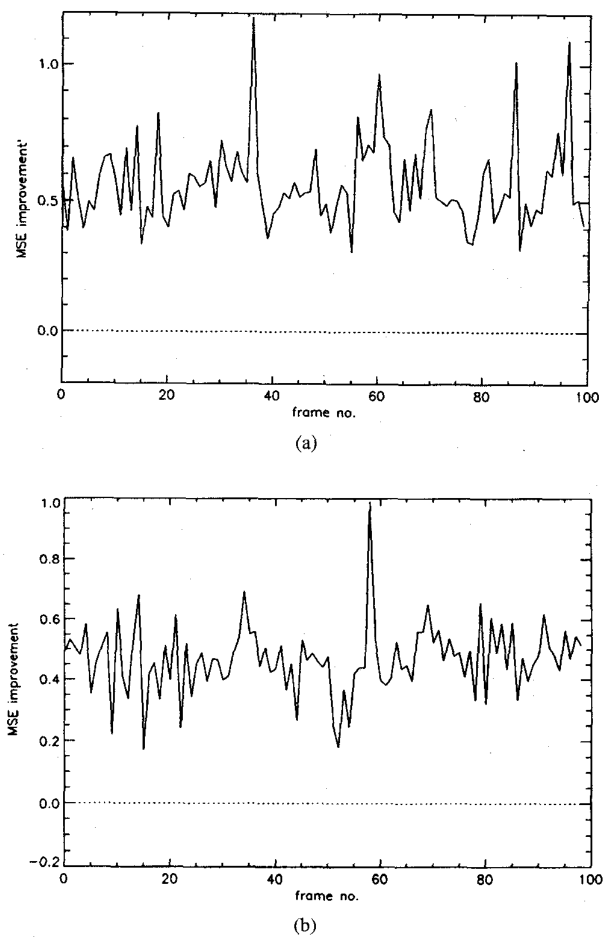

Fig. 7. Improvement of the proposed adaptive pixel decimation as compared with Liu and Zaccarin's [10] method for "salesman" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

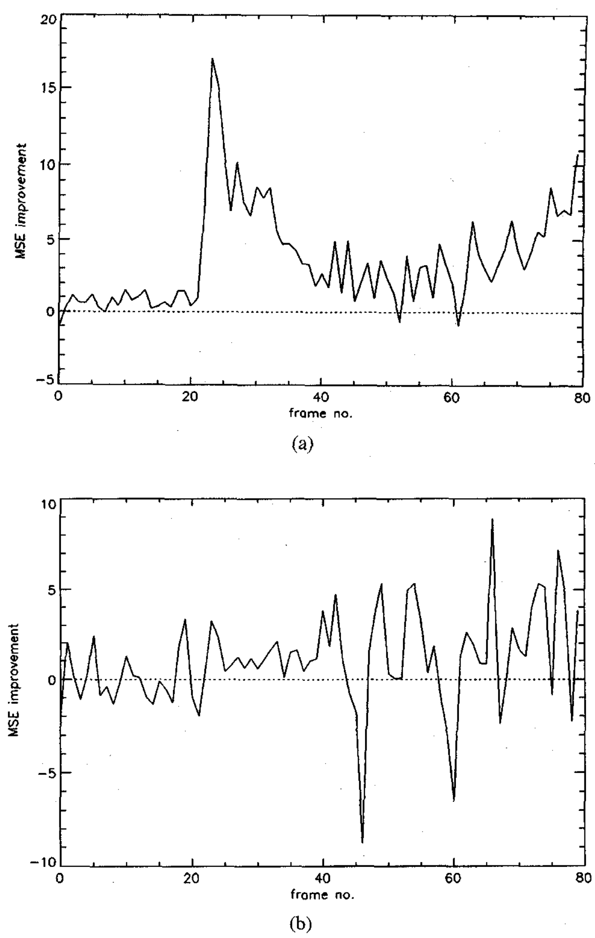

Fig. 8. Improvement of the proposed adaptive pixel decimation as compared with Liu and Zaccarin's [10] method for "tennis" sequence with (a) 8 × 8 block size and (b) 16 × 16 block size.

TABLE II
MSE PRODUCED BY DIFFERENT ALGORITHMS AT FRAME NUMBER WITH DIFFERENT BLOCK SIZE

	Frame no.	Exhaustive search without pixel decimation	Standard 4 to 1 pixel decimation	Proposed adaptive pixel decimation	Proposed adaptive pixel decimation with refinement of motion vector	Liu and Zaccarin [10]
Football	2	112.20	134.95	120.37	111.94	114.38
	80	199.30	235.82	227.86	200.37	205.07
salesman	10	19.83	24.88	22.19	20.05	20.72
	37	13.78	17.13	15.15	13.95	15.13
tennis	25	135.95	191.21	155.27	139.46	154.47
	67	82.63	108.88	91.76	83.98	86.14

	Frame no.	Exhaustive search without pixel decimation	Standard 4 to 1 pixel decimation	Proposed adaptive pixel decimation	Proposed adaptive pixel decimation with refinement of motion vector	Liu and Zaccarin [10]
Football	2	171.02	181.54	171.36	167.71	170.41
	80	343.64	371.87	353.33	343.30	345.45
salesman	10	29.41	32.59	30.22	29.47	29.69
	37	16.62	19.25	17.07	16.62	17.18
tennis	25	174.13	184.54	177.71	175.19	177.52
	67	173.89	188.02	184.39	171.81	180.68

of motion vector is used, an extra $nK[N^2 - H(i, j)]$ operations per block is needed. Since $nK[N^2 - H(i, j)] \ll F$, this overhead is negligible.

IV. RESULTS

A series of computer simulations have been conducted to evaluate the performance of the proposed adaptive pixel decimation technique. The image sequences "football," "salesman," and "tennis" have been used. The sequences contain translation, zooming-out and both slow and fast panning. A maximum allowable displacement (D) in the x and y directions is eight with block size of 8 × 8 and 16 × 16. 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 the standard 4 to 1 pixel decimation [4]. In other words, the computation is reduced by a factor of four as compared with the exhaustive search without pixel decimation. The number of motion vectors ($\boldsymbol{n}$) selected in the first stage is chosen as four. The predictions errors (MSE) of a standard 4 to 1 pixel decimation [4] and that of proposed algorithm with and without refinement of motion vector using different block sizes are shown in Figs. 3–5. It is seen that our proposed adaptive algorithm is significantly better than the standard 4 to 1 pixel decimation using different block sizes. In Figs. 3–5, the performance of the exhaustive search without pixel decimation for different block sizes are also shown. It is seen that the proposed algorithm without refinement of motion vector has an MSE very close to the exhaustive search without pixel decimation except after the 70th frame in "football" sequence for the block size of 8 × 8. But, the proposed algorithm with refinement of motion vector can reduce this error significantly. The proposed algorithm is even slightly better in some frames. It shows that our proposed algorithm can obtain better MSE performance when using MAD matching criterion.

The positive values in y-axis of Figs. 6–8 indicate the improvement in MSE of our proposed adaptive pixel decimation algorithm as compared with Liu and Zaccarin's [10] pixel decimation using alternating subsampling patterns in different block size. With an 8×8 block size, nearly all the frames from the proposed adaptive pixel decimation are better than that of Liu and Zaccarin's [10] approach. While, by using a block size of 16×16, about 72% of frames from proposed algorithm are better than that of Liu and Zaccarin's [10] approach in the "football" and "tennis" sequences. And, in the "salesman" sequence, all the frames of the proposed adaptive pixel decimation have improvement as compared with Liu and Zaccarin's [10] approach. Table II shows the MSE of different algorithms with different block sizes. 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 because the number of pixels used in the MAD criterion depends on the 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 blocks and employs less pixel in the smooth block to reduce the computation burden.

V. Conclusion

A new two-step 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 standard pixel decimation, and shows improvement compared to the famous approach given by Liu and Zaccarin [10]. This approach can certainly be used as an efficient technique for block matching motion vector estimation.

References

[1] N. G. Musmann, P. Pirch, and H. J. Grallat, "Advances in picture coding," *Proc. IEEE*, vol. 73, no. 4, pp. 523–548, Apr. 1985.

[2] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. COM-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. Commun.*, vol. COM-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.*, New Orleans, LA, Nov. 29–Dec. 3, 1981, pp. G.5.3.1–5.3.5.

[5] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COM-33, pp. 1011–1014, Sept. 1985.

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

[7] G. Kummerfeldt, F. May, and W. Wolf, "Coding television signals at 320 and 64 kbit/s," in *2nd Int. Tech. Symp. on Optical and Electro-Optical Applied Science and Eng.*, Cannes, Dec. 1985.

[8] M. Bierling and R. Thoma, "Motion compensating field interpolation using a hierarchically structured displacement estimator," *Signal Processing (EURASIP)*, vol. 11, no. 4, pp. 387–404, Dec. 1986.

[9] M. Bierling, "Displacement estimation by hierarchical block matching," in *SPIE, Visual Commun. Image Processing '88*, vol. 1001, pp. 942–951.

[10] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, Apr. 1993.

A Locally Quadratic Model of the Motion Estimation Error Criterion Function and Its Application to Subpixel Interpolations

Xiaoming Li and Cesar Gonzales

***Abstract*—It is observed that around the optimum point of the motion estimation process the error criterion function is well modeled as a quadratic function with respect to the motion vector offsets. This locally quadratic functional model decomposes the motion estimation optimization at subpixel resolutions into a two-stage pipelinable processes: full-search at full-pixel resolution and interpolation at any subpixel resolution. Practical approximation formulas lead to the explicit computations of both motion vectors and error criterion functional values at subpixel resolutions.**

I. Introduction

Accurate motion estimation is essential to effective motion-compensated video signal processing, and subpixel resolutions are required for high quality applications.

For the convenience of presentation, the present object is assumed to be an $M \times N$ macroblock, the search window is assumed to be an $X \times Y$ rectangular region, and the error criterion is the sum of the squares of pixel differentials. The principles and algorithms apply equally well to other object models, other search window shapes, and other matching criteria.

The motion estimation is described as an optimization problem

$$\|e^*\|_2^2 = \underset{x,y,w_x,w_y}{\text{minimize}} \sum_{m,n} |C[m][n] - \bar{P}[m+x+w_x][n+y+w_y]|^2$$

where the interpolated reference window $\bar{P}$, is defined as

$$\begin{aligned}\bar{P}[m+x+w_x][n+y+w_y] &= w_x \times w_y \times P[m+x][n+y] \\ &+ (1-w_x) \times w_y \times P[m+x+1][n+y] \\ &+ w_x \times (1-w_y) \times P[m+x][n+y+1] \\ &+ (1-w_x) \times (1-w_y) \times P[m+x+1][n+y+1].\end{aligned}$$

Optimization of this motion estimation at subpixel resolutions of W_x, W_y (representing the number of subpixel levels) requires computations of

$$M \times N \times (X-M+1) \times (Y-N+1) \times W_x \times W_y \text{ subtractions}$$

$$M \times N \times (X-M+1) \times (Y-N+1) \times W_x \times W_y \text{ multiplications}$$

Manuscript received March 31, 1995. This paper was recommended by Associate Editor L. Wu.

The authors are with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Publisher Item Identifier S 1051-8215(96)01670-9.