Express Letter

A New Three-Step Search Algorithm for Block Motion Estimation

Renxiang Li, Bing Zeng, and Ming L. Liou, *Senior Member, IEEE*

***Abstract*—The three-step search (TSS) algorithm has been widely used as the motion estimation technique in some low bit-rate video compression applications, owing to its simplicity and effectiveness. However, TSS uses a uniformly allocated checking point pattern in its first step, which becomes inefficient for the estimation of small motions. A new three-step search (NTSS) algorithm is proposed in this paper. The features of NTSS are that it employs a center-biased checking point pattern in the first step, which is derived by making the search adaptive to the motion vector distribution, and a halfway-stop technique to reduce the computation cost. Simulation results show that, as compared to TSS, NTSS is much more robust, produces smaller motion compensation errors, and has a very compatible computational complexity.**

I. Introduction

Motion estimation plays an important role in motion compensated image sequence coding. Due to the huge computation demand of the full search method, investigation of fast motion estimation algorithms has been a focus of research in the last decade [1]–[8]. In the early 1980s, some conventional fast algorithms were proposed, such as the three-step search (TSS) [2], the 2D logarithmic search [3], the conjugate directional search [4], [5], etc. Among these algorithms, TSS becomes the most popular one for low bit-rate video application (including videophone and videoconferencing), owing to its simplicity and effectiveness. However, TSS uses a uniformly allocated search pattern in its first step, which is not very efficient to catch small motions appearing in stationary or qusi-stationary blocks. To remedy this problem, several adaptive techniques have been suggested to make the search more adaptable to motion scale and uncertainty. A dynamic search window scheme is presented in [6], where the search window size is adjusted according to the uncertainty of motion scale. The uncertainty is estimated by the difference of block distortion measure (BDM) among the checked points. A smaller difference indicates a larger uncertainty and hence the search scope will be increased in the next step. Otherwise, the normal TSS search or just the 8-neighbor search is performed. In [7], a multistage scheme was introduced. In each stage, the minimum BDM is compared with a predetermined threshold, and the search stops if the BDM value is less than the threshold. Otherwise, it proceeds to next stage.

It is clear that these two algorithms employ multiple thresholds to control the search and these thresholds play an important role on the performance of the algorithm. On the other hand, the best set of thresholds may change as the image sequence under processing changes, thus making them difficult to be used in practical applications.

This paper proposes a new three-step search (NTSS) algorithm for fast motion estimation. The search pattern in each step is fixed and no thresholding operations are involved in this algorithm. Nevertheless, it is made to better utilize the motion distribution of real world image sequences in low bit-rate video applications and is adaptive in the sense that the algorithm may stop at the second or third step. It will be shown that the NTSS algorithm retains the simplicity and regularity of the original TSS method, works better than TSS in terms of motion compensation error and robustness, and is quite compatible to TSS in terms of computational complexity.

Manuscript received April 3, 1994. This work is supported by a research grant from Hongkong Telecom Institute of Information Technology (HKTHT) and was recommended by Dr. Ming-Ting Sun.

The authors are with the Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

IEEE Log Number 9403766.

II. A Framework of Fast Motion Estimation Algorithms

Given a block of size $N \times N$, the block motion estimation searches for a motion vector (in a previous frame) that yields the minimum BDM within a neighborhood area. Suppose that the maximum motion in vertical and horizontal directions is $\pm W$. Then, there are totally $(2W+1)^2$ motion vectors to be checked if the full search method is used, each one corresponding to a point in the search window. The BDM values on all points in the search window form an error surface (assuming the absolute error as the distortion measure),

$$E(u, v) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |f_{t-\tau}(i+u, j+v) - f_t(i, j)|, \quad -W \le u, v \le W \quad (1)$$

where (u, v) is the candidate motion vector, and $f_{t-\tau}(\cdot, \cdot)$ and $f_t(\cdot, \cdot)$ refer to the blocks in a previous frame and the current frame that are to be compared. The complexity of this error surface has significant impact on the performance of the algorithm. All conventional fast algorithms [1]–[7] are explicitly or implicitly based on the following assumption [3]: *BDM increases monotonically as the checking point moves away from the global minimum.* Obviously, this assumption essentially requires that the error surface be *unimodal* over the search window. Unfortunately, this is usually not true due to many reasons such as the aperture problem, the textured (periodical) local image content, the inconsistent block segmentation of moving object and background, the luminance change between frames, and etc. As a consequence, the search would easily be trapped at a local minimum.

With the multiple stage search scheme such as TSS, the checking points in the first step are allocated uniformly in the search window. Such a configuration may not be very appropriate for some blocks in which small motions are observed, and therefore how to optimally design the checking point pattern for this step becomes the main concern. In spite of uncertainty in large spatial scale, we can reasonably assume that *the error surface is monotonic in a small neighborhood around the global minimum.* Hence, if one of the checking points is close enough to the global minimum, the chance to find the global minimum will be high. Mathematically, this can be formulated as an optimization with objective function defined as

$$\bar{d} = \sum_{x_i \in S} \|c_i - x_i\| P(x_i) \quad (2)$$

where S consists of all points in the search area, c_i is the checking point that is closest to x_i, $\|\cdot\|$ stands for the Euclidean distance

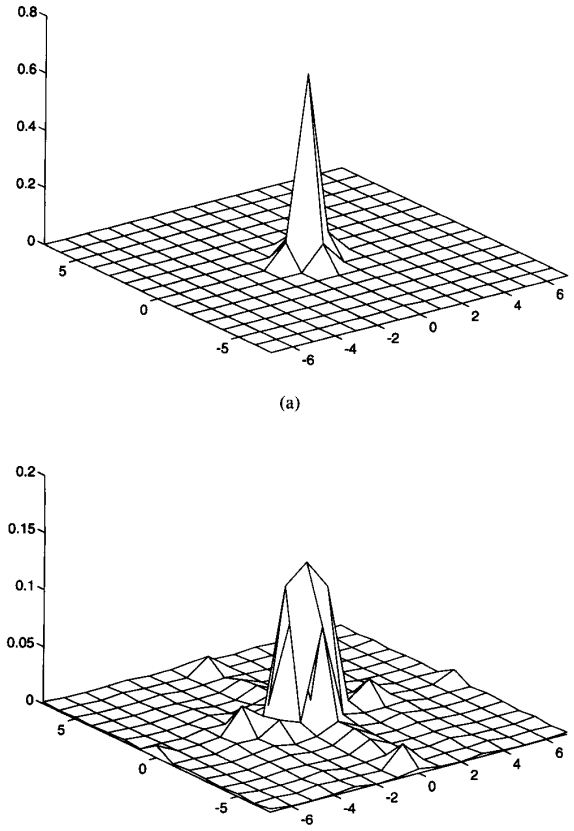


Fig. 1. The motion vector distribution derived from the full search for 100 frames of (a) Salesman sequence and (b) Miss America sequence (block size: 16 × 16).

measure, and $P(x_i)$ is the probability that global minimum will occur at x_i. Notice that the number of checking points c_i's is smaller than the cardinality of S; there would otherwise be no need for the design since the full search prevails.

When the above optimization is to be solved, one has to take into consideration some basic characteristics of the distribution of global minima. For instance, the following property is well known: *The block motion field of a real world image sequence is usually gentle, smooth, and varies slowly*, and it is particularly true in low bit-rate video applications. As a consequence, the global minimum distribution is center-biased, instead of distributed uniformly, as is demonstrated by the typical examples shown in Fig. 1. For the Salesman sequence, there are nearly 80% blocks that can be regarded as stationary or qusi-stationary blocks and most of the motion vectors are enclosed in the central 5×5 area. For the Miss America sequence, the motion vector distribution is more diverse as compared to Salesman. However, it is still highly center-biased. With such a particular form of distribution, the checking point pattern should also be center-biased so as to minimize the average distance $\overline{d}$.

III. The NTSS Algorithm

According to the discussion in the last section, we now present one feasible solution to the optimization problem, which is termed a new three-step search (NTSS) algorithm. NTSS differs from TSS by (1) assuming a center-biased checking point pattern in its first step and (2) incorporating a *halfway-stop* technique for stationary or qusi-stationary blocks. The details are given below:

1) In the first step, in addition to the original checking points used in TSS, eight extra points are added, which are the eight neighbors of the search window center, as shown in Fig. 2(a). (As such, the checking point pattern is highly center-biased.)
2) A halfway-stop technique is used for stationary and qusi-stationary block in order to fast identify and then estimate the motions for these blocks:
   a) If the minimum BDM in the first step occurs at the search window center, stop the search. (This is called the first-step-stop.)
   b) If the minimum BDM point in the first step is one of the eight neighbors of the window center, the search in the second step will be performed only for eight neighboring points of the minimum[1] and then stop the search. (This is called the second-step-stop.)

The block diagram of NTSS is shown in Fig. 2(b). Clearly, it has retained the simplicity and regularity of TSS. It is seen that the only case where a complete three-step search needs to be executed is when the minimum BDM point of the first step is not the window center nor any of its eight neighboring points. Due to the use of this new center-biased checking point pattern in the first step, the introduction of the first-step-stop technique seems quite reasonable. For such a case, the motion is really gentle and therefore the window center deems to represent the true motion vector. For the case where a second-step-stop happens, it is also easy to understand that (1) the motion is again small; (2) the execution of the second step makes the estimation more accurate (and thus is worthwhile); and (3) the possibility of finding the true motion after the second step is quite high (thus justifying the suitability of a second-step-stop).

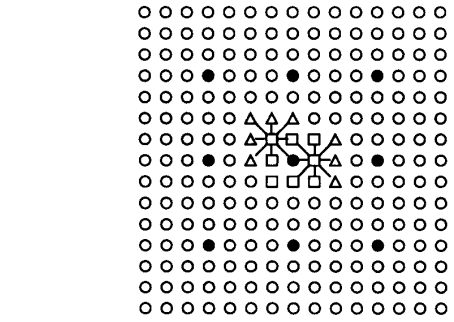
In low bit-rate video applications, the search is usually performed for an area of size 15×15, thus $W = 7$. For such a choice, the full search will check 225 points, while TSS checks 25 points, thus leading to a speed-up ratio of 9. Although NTSS uses more checking points in its first step as comapred to TSS, the first-step-stop and second-step-stop can reduce computation significantly. For example, eight block matches will be saved once a first-step-stop occurs. Depending on the position of the minimum BDM point (in the first step) on the 8 neighbors of the window center, five or three block matches will be saved once a second-step-stop occurs: (1) if the minimum is one of the four neighboring positions along the horizontal or vertical directions, five block matches will be saved; (2) if the minimum is one of the four neighbors along the two diagonal directions, three block matches will be saved.

Actually, the number of block matches needed in NTSS for estimating a block motion vector can be estimated by $17P_1 + 20P_2 + 22P_2' + 33(1 - P_1 - P_2 - P_2')$, where P_1 is the probability of occurring a first-step-stop while P_2 and P_2' are respectively the probabilities of occurring a second-step-stop in the two cases mentioned above. These probabilities are dependent on how many stationary or qusi-stationary blocks a video frame contains. In low bit-rate video applications such as videophone and videoconferencing, since the block motion is gentle and background is usually stationary, the saving is thus quite substantial. In the worst case (i.e., there is no single stationary block), the NTSS algorithm requires 33 block matches as compared to 25 matches needed in TSS. The

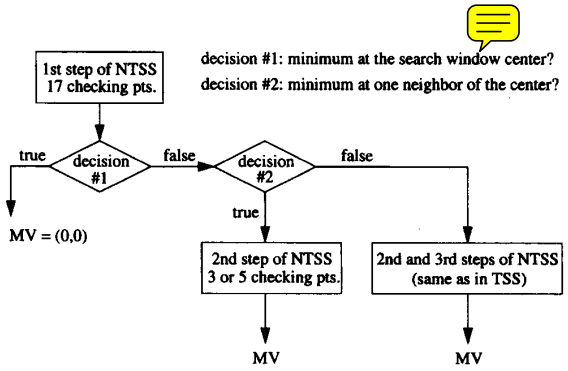
[1] Notice that some of these eight points have already been checked in the first step.

TABLE I
COMPARISON BETWEEN TSS AND NTSS (IN TERMS OF SPEED-UP RATIOS W.R.T. FULL SEARCH, PROBABILITIES OF CATCHING TRUE MOTIONS, AND AVERAGE DISTANCES

	speed-up ratios		probabilities		average distances	
	Salesman	Miss America	Salesman	Miss America	Salesman	Miss America
TSS	9.0	9.0	0.951	0.535	0.369	1.193
NTSS	10.94	7.94	0.990	0.722	0.044	0.687



(a)

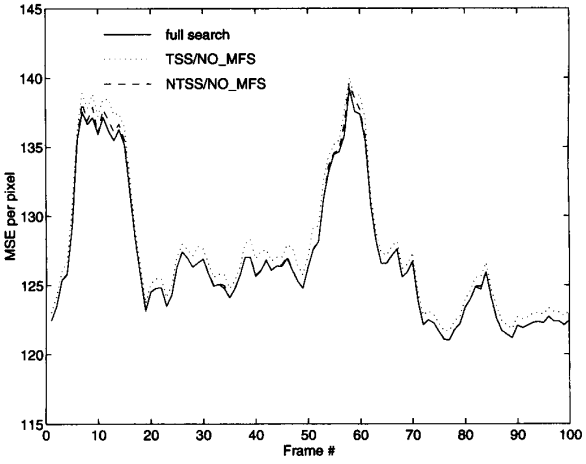


(b)

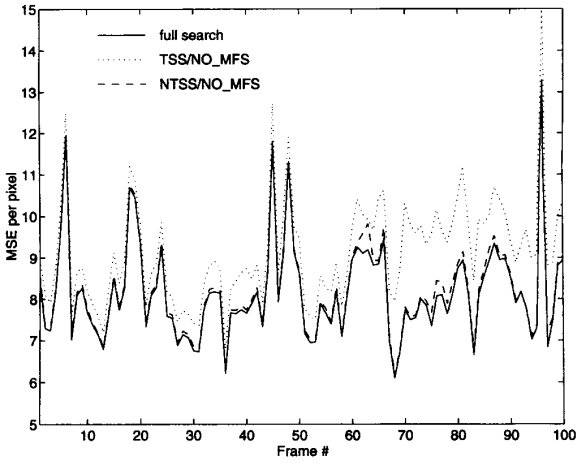
Fig. 2. (a) Filled circles are the checking points in the first step of TSS, squares are the 8 extra points added in the first step of NTSS, and triangles explain how the second step search is performed if the minimum BDM in the first step is at one of the 8 neighbors of the window center. (b) The block diagram of the NTSS algorithm.

actual speed-up ratios of NTSS with respect to full search are summarized in Table I. Comparing with the results of TSS, we see that NTSS and TSS basically possess rather comparable computational complexity.

On the other hand, we can compute, based on test image sequences, the probability that the global minimum BDM point (*i.e.*, the true motion vector) is at the center of search window if a first-step-stop occurs as well as the probability that the true motion is one of the eight neighboring points of the window center if a second-step-stop occurs at such neighbor. For the Salesman and Miss America test sequences, such probabilities are observed to be close to 1, thus proving the validity of the NTSS algorithm. Based on the test sequences, we can also compute the overall probability that the true motion is found by using NTSS or TSS. These probabilities for the two sequences are also presented in Table I, from which it is seen that NTSS provides higher probabilities than those of TSS. Finally, to further compare the effectiveness of NTSS and TSS, their average distances are calculated



(a)



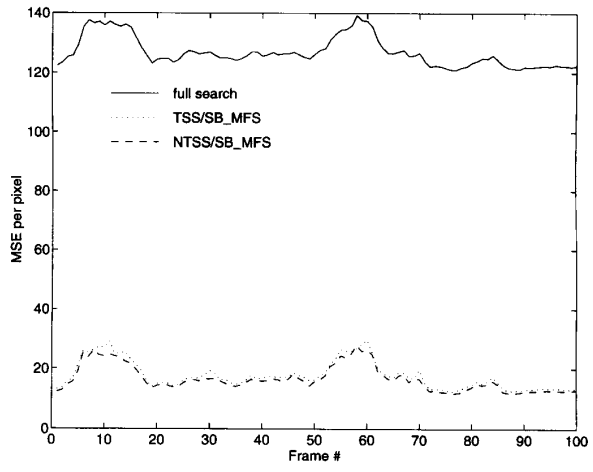
(b)

Fig. 3. Test results (in terms of MSE) of full search, TSS, and NTSS for (a) Salesman sequence and (b) Miss America sequence.

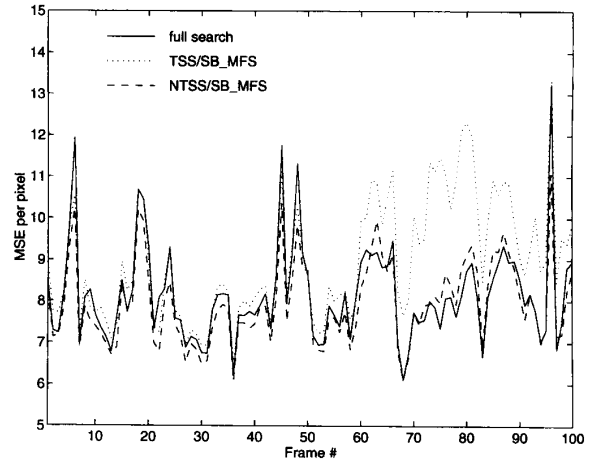
from (2) using the motion vectors obtained from full search as global minima. We see from Table I that the center-biased pattern used in NTSS results in much smaller average distance than the uniformly distributed pattern in TSS.

IV. SIMULATION

In our simulations, the BDM is defined to be the mean absolute difference (MAD). The block size is fixed at 16 × 16, and the

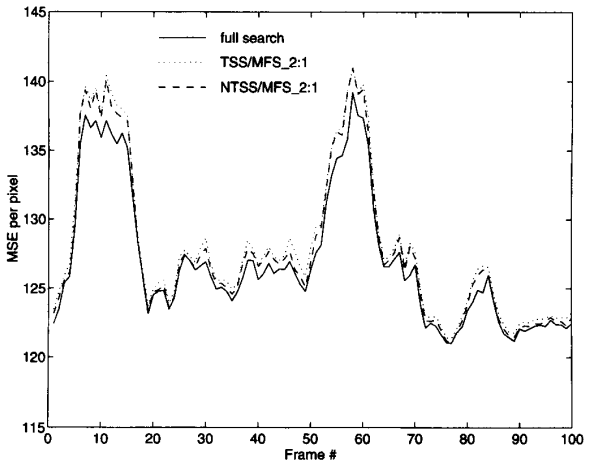


(a)

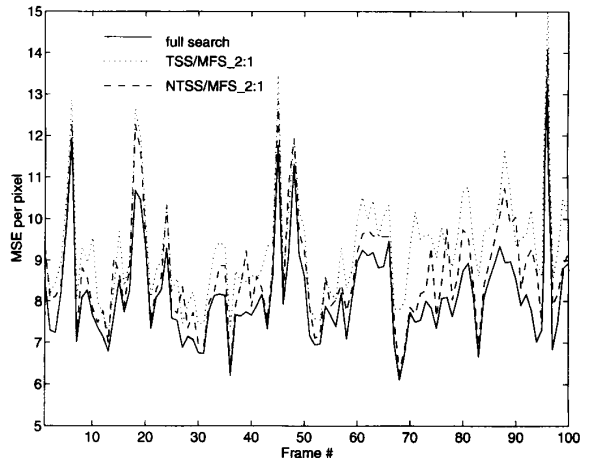


(b)

Fig. 4. Test results of TSS and NTSS with sub-block (8 × 8) subsampling and full search (16 × 16) for (a) Salesman sequence and (b) Miss America sequence.



(a)



(b)

Fig. 5. Test results of TSS and NTSS with 2:1 block motion subsampling and full search for (a) Salesman sequence and (b) Miss America sequence.

maximum motion in row and column is assumed to be ± 7. We used 100 frames of Salesman and Miss America sequences (in CIF format) to test the NTSS algorithm and compare it with TSS and full search. The results (in terms of mean-square error measure) are shown in Fig. 3. From it, we see that NTSS almost always provides the same performance as that of full search, while in many frames, the performance of TSS deteriorates badly. In addition, we also tested two motion field subsampling schemes to verify the robustness of NTSS and TSS: the first one is the sub-block subsampling and the second one is the 2:1 block subsampling [8]. They provide an additional speed-up ratio by a factor of 4 and 2, respectively. Fig. 4 presents the simulation results for the sub-block subsampling scheme. An interesting observation is that the motion compensation error for Salesman sequence is much lower than the one measured in Fig. 3 where a 16 × 16 block is used. Fig. 5 gives the results for the 2:1 block subsampling scheme. From these simulation results, it is very clear that NTSS always works better than TSS and, in particular, is much more robust than TSS when the motion field is subsampled.

V. Conclusion

In this paper, the block motion estimation (for low bit-rate video compression) was formulated as an optimization problem in which the objective function is defined by an average distance measure and a center-biased constraint has been taken into consideration. As a feasible solution, we proposed a new three-step search algorithm for fast motion estimation. We constructed a center-biased search point pattern in the first step by adding 8 extra checking points which are the neighbors of the window center. This has significantly reduced the average distance. In the mean time, we introduced the use of a halfway-stop technique to keep the NTSS algorithm compatible to TSS in terms of computational complexity. Experimental results showed that, compared to TSS, NTSS always provides smaller motion compensation errors and in particular is much more robust, no matter it is used alone or combined with motion field subsampling techniques.

REFERENCES

[1] H. G. Musmann, P. Pirsh, and H.-J. Grallert, "Advances in picture coding," *Proc. of IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. NTC 81*, pp. C9.6.1–9.6.5, New Orleans, LA, Nov./Dec. 1981.

[3] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1799–1808, Dec. 1981.

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

[5] S. Kappagantula and K. R. Rao, "Motion compensated interframe image prediction," *IEEE Trans. Commun.*, vol. COM-33, pp. 1011–1015, July 1985.

[6] L.-W. Lee, J.-F. Wang, J.-Y. Lee, and J.-D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 3, pp. 85–87, Feb. 1993.

[7] S. C. Kwatra, C-M Lin and W. A. Whyte, "An adaptive algorithm for motion compensated color image coding," *IEEE Trans. Commun.*, vol. COM-35, pp. 747–754, July 1987.

[8] B. Liu and A. Zaccarin, "New fast algorithm for estimation of block motion vectors," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 3, pp. 148–157, Apr. 1993.

The Effect of a Loop Filter on Circulating Noise in Interframe Video Coders

Christopher J. Hollier, John F. Arnold, and Michael C. Cavenor

***Abstract*— An examination of the distribution of signals within the prediction loop of an interframe video codec reveals that small random fluctuations caused by the quantizer build up over a period of time. This quantization noise circulates within the prediction loop causing a gradual decrease in the signal to noise ratio of the reconstructed sequence and an increase in the required data rate. In general, both of these effects can be nullified by the inclusion of a low pass loop filter.**

I. INTRODUCTION

Most high performance video encoding schemes employ the interframe data reduction technique known as motion compensated prediction. A prediction of the current input frame is made based upon a rearranged assembly of pixel blocks selected from the reconstruction of the previous frame. The difference image, formed by subtracting the prediction from the input frame, is then encoded and transmitted along with the motion vectors for each block in the prediction image. A reduction in the transmitted data results from the fact that the information content of the difference image is considerably less than that of the input image.

In certain implementations of this encoding technique, including the CCITT H.261 standard [1], provision is made for the optional inclusion of a low pass filter operating on the prediction image and located within the prediction feedback loop. Whilst it is agreed that the inclusion of this loop filter is generally beneficial to the operation of the encoder, it appears that there is no published evidence as to why this is so. This paper examines the role of this loop filter. While the encoder used to examine this phenomenon employs a subband filtering approach to subdivide the energy in the difference image into a range of frequency bands, it could be expected that the results obtained would equally apply to other interframe encoding techniques, including those based on the Discrete Cosine Transform (DCT).

Manuscript received March 24, 1993; revised June 30, 1994. This paper was recommended by Wen. H. Chen and was supported by the OTC Australia Limited under research contract No. 23703.

The authors are with the Department of Electrical Engineering, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, A.C.T. 2600, Australia.

IEEE Log Number 9404757.

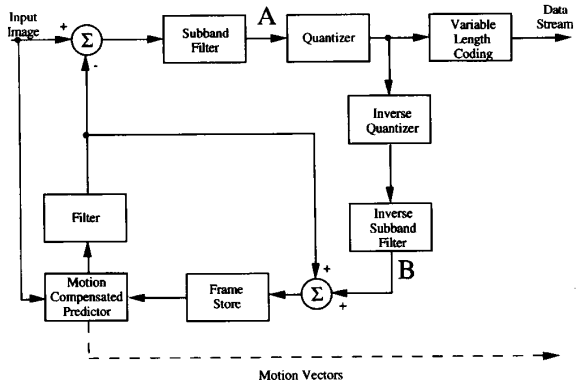


Fig. 1. Block Diagram of the Encoder.

Section II describes the basic motion compensated subband coder used in this study. Section III demonstrates the performance of this coder both with and without the inclusion of a loop filter. It is shown that employing a loop filter results in both a substantial increase in reconstructed image quality and a decrease in the transmitted data rate. Section IV describes experiments which aim to identify the reasons for this improved performance. The results obtained demonstrate that circulating noise in the feedback loop introduced by quantization and rounding in the coder without a loop filter lead directly to the reduction in performance. The major conclusions of the investigation are presented in Section V.

II. BASIC CODING TECHNIQUES

A simplified block diagram of the encoder which has been used to demonstrate the effect of the loop filter is shown in Fig. 1. The codec has deliberately been kept simple in order to demonstrate most clearly the function of the loop filter. Other more complex coders that have higher coding efficiencies have also been examined [2] and all exhibit a behaviour similar to the results described herein. In these cases, however, the influence of the loop filter is often difficult to isolate due to other artefacts introduced by the coding process.

The codec employs a standard motion compensated coding scheme which incorporates subband filtering [3] together with variable word length coding of the difference image. The subband filtering process uses the filters proposed by Le Gall and Tabatabai [4] and is used to subdivide the difference image into sixteen frequency bands as shown in Fig. 2. Each of these subbands are of differing importance in the reconstruction process and so certain bands, if desired, may be coarsely quantized without causing major image degradation. If, however, these bands are not modified in any way then the inverse subband filtering process can recover the original image exactly. The