

ROBUST BLOCK-MATCHING MOTION-ESTIMATION TECHNIQUE FOR NOISY SOURCES

Robert M. Armitano and Ronald W. Schafer

Frederick L. Kitson and Vasudev Bhaskaran

Digital Signal Processing Laboratory
Georgia Institute of Technology
Atlanta, GA 30332
armi,rws@eedsp.gatech.edu

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
kitson,bhaskara@hpl.hp.com

ABSTRACT

Video coding standards use the block-matching algorithm (BMA) and motion-compensated prediction to reduce temporal redundancies present in image sequences. Block matching is used since it is computationally efficient and produces a minimal representation of the motion field that is transmitted as side information. In order to build a robust coder the motion-estimation technique must be able to track motion in a noisy source. The approach presented in this paper uses spatio-temporal motion prediction, providing an accurate motion estimate even in the presence of noise. With this approach, noisy sources can be compressed efficiently and robustly in standard video coders (e.g., MPEG-1, MPEG-2, H.261, and H.263 [1]) with little increase in complexity.

1. INTRODUCTION

In order to build a video coder that is robust in the presence of noise, the motion-estimation process must be able to track objects within a noisy source. In a noisy source, objects appear to change from frame to frame because of the noise, not necessarily as the result of object motion. Tracking objects within a noisy environment is difficult, especially using the BMA. The motion model used in the BMA algorithm, assumes that frame-to-frame changes occur as the result of object motion. This assumption no longer holds in noisy sources. To aid the motion-estimation process in the presence of noise, motion-vector prediction has been used to improve the tracking capabilities of the video coder in this paper. The motion-vector prediction process has been implemented as the linear predictive search (LPS) as reported in [2].

The LPS uses motion-vector prediction to produce a crude estimate, $\hat{\mathbf{v}}_{\mathbf{b}}(t)$, of motion in the current block, $\mathbf{v}_{\mathbf{b}}(t)$. Once the motion-vector estimate is computed, a localized search centered on estimate is performed, that hones in on the actual motion vector, $\mathbf{v}_{\mathbf{b}}(t)$. As in the BMA, the mean absolute difference (MAD) matching criteria is used in the LPS. Previously computed motion vectors in past, present, or future frames, in a three-dimensional region of support

(ROS), are used in a predictor that determines a reliable estimate of the current motion vector,

$$\hat{\mathbf{v}}_{\mathbf{b}}(t) = \sum_{k=1}^p \alpha_k \mathbf{v}_{\mathbf{b}_k}(t_k),$$

where $\mathbf{b}$ is the current block, α_k is the estimator, p is the predictor order, and $\mathbf{v}_{\mathbf{b}_k}(t_k)$ are the previously computed motion vectors. Further references to the predicted motion vector will not be indexed by the current block, $\mathbf{b}$, the block index will be implied in the context.

In Section 2, the LPS algorithm is introduced. In Section 3, we examine the LPS algorithm with respect to its application to robust video coding and compare the behavior of the motion vectors in the full search (FS) and LPS algorithms. Simulation results using the LPS in the presence of noise are shown in Section 4. In Section 5, the conclusions are presented.

2. LINEAR-PREDICTIVE SEARCH

The LPS uses a predicted motion vector, $\hat{\mathbf{v}}$, to bias the search in the general direction of object motion, as shown in Figure 1. When the true motion vector $\mathbf{v}$, lies within the smaller search region, the LPS will result in a motion vector that is identical to the true motion. When the LPS algorithm fails, the estimate, $\hat{\mathbf{v}}$, centers the search region in an area that does not intersect the true motion vector, $\mathbf{v}$.

The computational complexity of the LPS is quite small. For each motion vector, p multiplies and $p - 1$ adds must be computed (where p is the order of the predictor). In the simulations shown here a 13th order predictor was used.

When the true motion vector $\mathbf{v}(t)$, lies within the search region defined by the LPS a motion vector that is identical to the true motion (as defined by the FS) will be found. When the LPS fails, the estimate, $\hat{\mathbf{v}}(t)$, centers the search region in an area that does not intersect the true motion vector, $\mathbf{v}(t)$. In a real-life coder the computational burden of motion estimation will require an implementation of the LPS in conjunction with a fast computation technique.

In Figure 2 the MAD surface¹ is shown for the FLOWER-GARDEN sequence. In Figure 2.a the MAD surface is shown

This work is supported in part under a grant from the Hewlett-Packard Company.

¹The MAD surface is a plot where, for each location, (d_x, d_y) , in the search region, the value of $D_{MAD}(d_x, d_y)$ is computed.

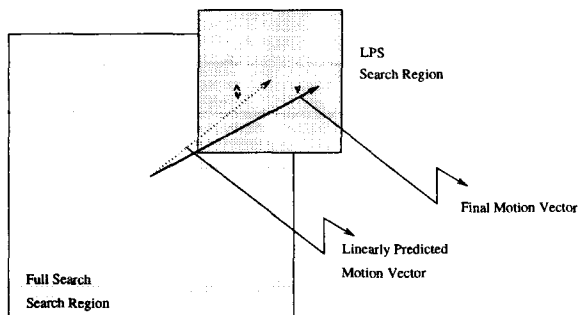


Figure 1: Linear Predictive Search Process.

for the full search BMA and the LPS MAD surface is shown in Figure 2.b. The predictive search has shifted the search region such that the minimum point in the MAD surface is centered in the search region, whereas the full search is offset by length of the motion vector. A (10,10) surface is shown in these plots. (The MAD surface is for block (9,7) in the 16th frame.)

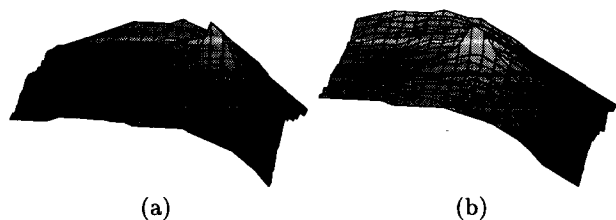


Figure 2: FLOWERGARDEN MAD surface: comparing FS to LPS.

3. LPS IN NOISY VIDEO SEQUENCES

In noisy environments, the MAD matching criteria must discriminate between the minimum value of MAD surface and a large number of false matches. By biasing the search towards the estimate, $\hat{v}(t)$, many false targets are eliminated from the computation of the MAD.

The results obtained for the LPS in noisy sources can be analyzed with respect to the motion vector field they produce. In the case of moderate noise with an SNR_p of 30 dB, the motion field obtained from the LPS is smoother than the motion field for the FS, as illustrated in Figure 3. Here we define $SNR_p = 10 \log_{10} \frac{255^2}{\sigma_n^2}$, where σ_n^2 is the variance of the noise. Having a smoother motion field not only improves the prediction error as measure by PSNR (as shown in Section 4), but also improves the efficiency with which

These values comprise a surface where the minimum distortion value corresponds to the motion vector $\mathbf{v}(t) = [d_x, d_y]^T$. It is difficult to locate the global minimum because of the ambiguity in the surface (spurious signals and noise) and due to the large number of local minima. NOTE: for display purposes, the surface plot is inverted, such that the maximum value corresponds to the displacement vector.

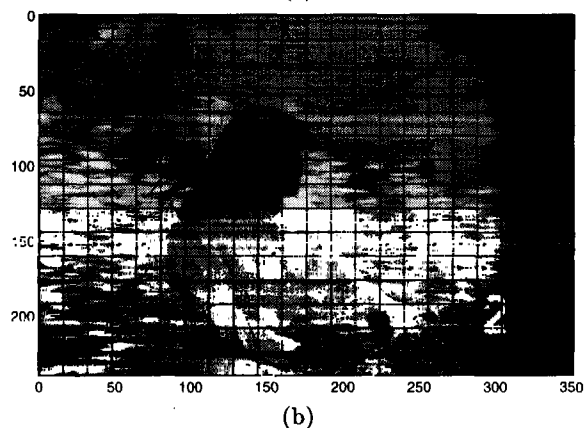
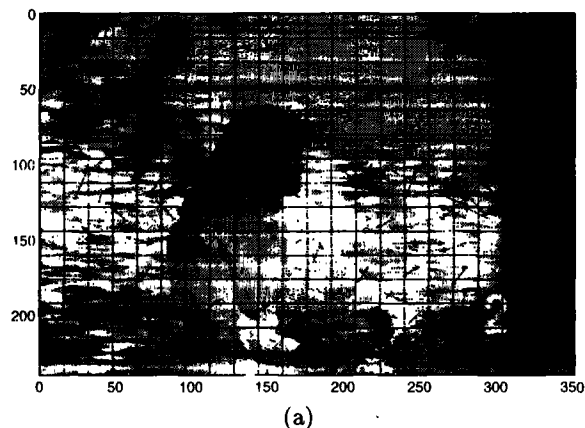


Figure 3: Motion field for the CYCLEGIRL sequence in the presence of AWGN.

the motion vectors are coded. The motion field with an input SNR_p of 30 dB is shown in Figure 3a for the FS and in Figure 3b for the LPS. In these examples, a 16×16 search region is used for the FS and LPS.

The LPS employs a 13-tap spatio-temporal filter. The ROS for this filter extends spatially along the horizontal and vertical axis and spans into the previous frame as shown in Figure 4. It was found that high-order filters perform better than low-order filters in the presence of additive white Gaussian noise (AWGN) and the 13-tap spatio-temporal filter provided sufficient robustness without adding undue coder complexity. The 13-tap filter requires a total of 13 multiplies and 12 adds per motion vector.

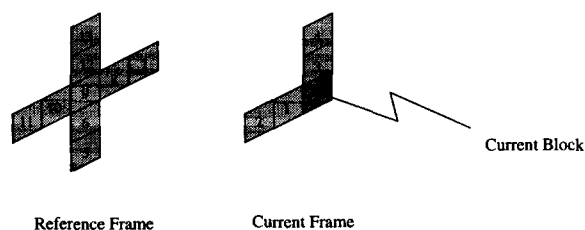


Figure 4: Region of support for the 13-tap spatio-temporal LPS prediction filter.

To simulate noisy sources, Gaussian noise was added to the image sequences. AWGN may not be the ideal noise model for video source material, but it is experimentally reproducible and indicative of the noise phenomena.

4. RESULTS

Comparisons of the full search, hierarchical search², and log search [4] indicate that the LPS is a robust technique for motion estimation. Results of these comparisons are shown for different noise levels and video sequences. Additive white Gaussian noise was added to image sequences with an input SNR_p varying from 10 dB to 50 dB. PSNR was measured between the input (noisy) frames to the coder and frames reconstructed using motion-compensated prediction.

In Figures 5 and 6, the average PSNR (across all input frames) is plotted against input noise level. The average PSNR, $PSNR_{avg}$, is given as

$$PSNR_{avg} = \frac{1}{F} \sum_{i=1}^F PSNR_i,$$

where $PSNR_i$ is the measured PSNR for frame i and F is the total number of frames. Here we compare the LPS scheme against conventional motion-vector search schemes. Under normal operating conditions, e.g., input SNR_p between 30 to 50 dB, the performance of the LPS is as much as 1 dB better than the performance of the FS algorithm for the CYCLEGIRL sequence, as seen in Figures 5 and 7. In Figure 6, the results for the CHEERLEADER sequence are plotted. In CHEERLEADER examples, the LPS is always better than the FS, log search, and hierarchical search for every noise conditions, but the improvement using the LPS is not as great as in the CYCLEGIRL case. In Figure 5, a 16×16 search region was used with a 13-tap spatio-temporal filter. In Figure 6, an 8×8 search region is used with a 13-tap spatio-temporal filter.

All BMA motion estimation techniques fail for extremely noisy sequences, e.g., for input SNR_p of 10 dB. At low noise levels (input SNR_p from 40 to 50 dB) the coder's performance is comparable to its performance in a noise-free environment. For high-noise environments, little can be done to improve motion-estimation efficiency. Figures 5 and 6 show all block-matching motion-estimation algorithms failing at high noise levels,³ (10 to 20 dB); however, even at these high levels, LPS yields a performance gain of 0.1 to 0.05 dB over the FS algorithm.

In Figure 7, the performance of the LPS is shown at a fixed input noise level of 30 dB across all input frames of the CYCLEGIRL sequence. In this example, a 16×16 search region was used with a 13-tap spatio-temporal filter. The LPS outperforms the FS and other fast search algorithms at this moderate noise level.

In Figure 8, the performance for the CHEERLEADER is shown under a high-noise condition. Here, an extremely

²The hierarchical search used in these experiments use down-sampling only, without motion-vector filtering [3].

³In typical video sequences, realistic noise levels occur below 30 dB, i.e., $SNR_p > 30$ dB.

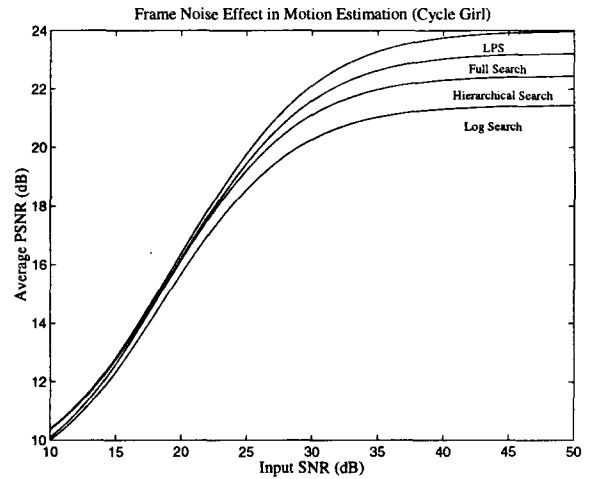


Figure 5: PSNR (average) vs source PSNR for CYCLEGIRL sequence.

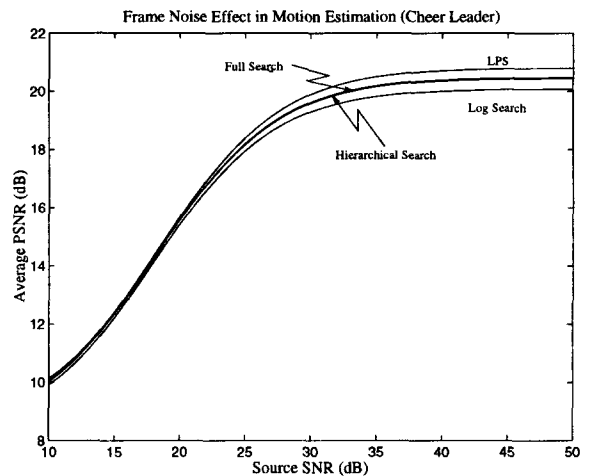


Figure 6: PSNR (average) vs source PSNR for CHEERLEADER sequence.

high input SNR_p of 10 dB is used for the input to the motion estimation algorithm. Although the LPS performs better than the other algorithms, the difference in performance between the different methods is negligible. However, since the LPS adds little complexity to the video coder, it should be used in all cases.

Performance of the LPS is dependent on the amount of motion present in the video sequence and on the size of the search window. In Figure 9, the improvement of the LPS over the FS algorithm is shown. For very large search regions, i.e., search regions greater than 16×16 , there is negligible improvement when using the LPS, since the algorithm cannot benefit from the localized search. For smaller search windows, i.e., search windows smaller than 4×4 , the search region is unable to encompass the minimum MAD. At intermediate search region sizes, the LPS benefits from the localized-search process and for these cases the

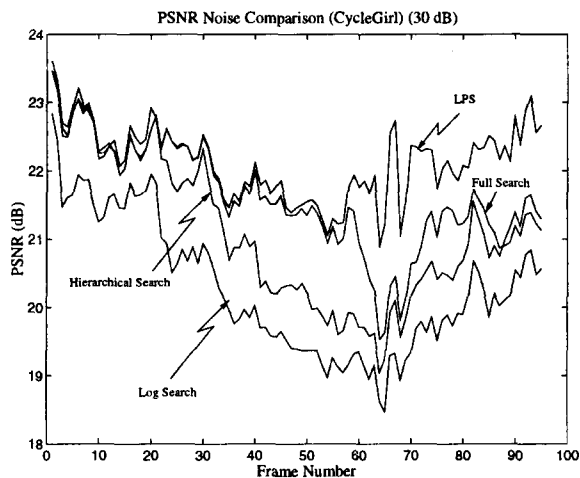


Figure 7: PSNR vs frame number of fixed input PSNR, CYCLEGIRL sequence.

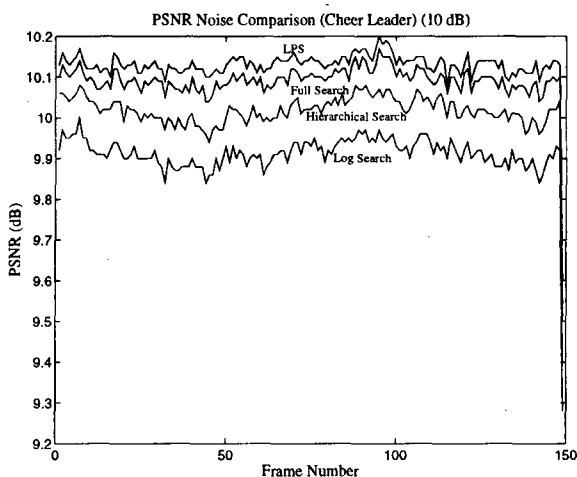


Figure 8: PSNR vs frame number of fixed input PSNR, CHEERLEADER sequence.

LPS's performance is optimal as evident in Figure 9.

5. CONCLUSIONS

The LPS provides an advantage over the FS algorithm in the presence of AWGN. With the LPS, the motion field is smoother, providing a more accurate measure of object motion. This characteristic of the LPS provides performance gains over the FS algorithm with AWGN. At relatively low-noise levels, the LPS's performance is comparable to its performance in the noise-free environment. At intermediate noise levels, SNR_p around 30 dB, the gain provided by the LPS with the CYCLEGIRL video sequence is more than 1.5 dB with a 16×16 search region to as much as 3 dB with an 8×8 search region, as shown in Figure 7. At high noise levels SNR_p around 10 dB the BMA model fails, yet even under these extreme conditions, the LPS provides

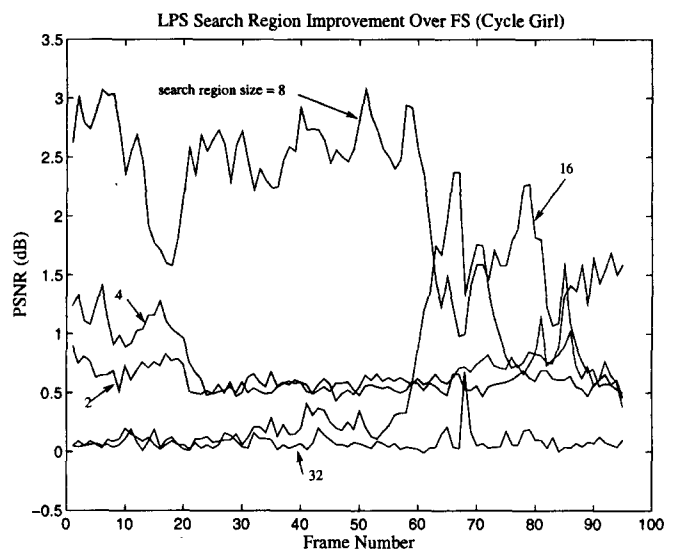


Figure 9: Improvement over FS for the LPS for different search regions in the presence of 30 dB AWGN for the CYCLEGIRL sequence.

improvement in performance over the FS algorithm.

Since the computational burden of the LPS is negligible when compared with the computational complexity of the FS BMA, its use in motion-compensated prediction is recommended. In addition to its PSNR performance, the LPS also yields smooth motion fields, as seen in Figure 3. This characteristic makes the motion vectors easier to code, lowering the overhead associated with motion parameters.

6. REFERENCES

- [1] V. Bhaskaran and K. Konstantinides, eds., *Image and Video Compression Standards*. Kluwer Academic Publishers, 1995.
- [2] R. M. Armitano, R. W. Schafer, F. L. Kitson, and V. Bhaskaran, "Linear predictive coding of motion vectors," *Proceedings of the Internal Picture Coding Symposium*, pp. 223-236, September 1994.
- [3] M. Bierling and R. Thoma, "Motion compensating field interpolation using a hierarchical structured displacement estimator," *Signal Processing*, vol. 11, pp. 387-404, December 1986.
- [4] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, vol. 29, pp. 1799-1808, December 1981.