A NEW FAST MOTION ESTIMATION METHOD BASED ON TOTAL LEAST SQUARES FOR VIDEO ENCODING

Sachin G. Deshpande

Information Processing Laboratory
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

Jenq-Neng Hwang

Information Processing Laboratory
Department of Electrical Engineering
University of Washington
Seattle, WA 98195

ABSTRACT

We present a new fast motion estimation method useful for high speed video encoding. Most of the motion estimation methods for video coding can be classified as Block Matching (BM) methods or Pel Recursive (PR) methods . Majority of the current fast motion estimation methods belong to block matching category. These methods try to reduce the number of search locations. Our proposed method is based on the pel recursive formulation. However, in order to achieve fast estimation, we operate on a block of pixels using a Total Least Squares (TLS) based estimation scheme which tries to estimate the true motion vector for each block. The major advantages of the proposed method include very fast estimation, almost constant time for motion estimation for all the video sequences, fractional pel accuracy, and better performance for noisy sequences. We present extensive simulation results to illustrate the performance of the proposed method.

1. INTRODUCTION

Motion estimation is the most time consuming operation in a typical video encoder. In video encoding standards like MPEG1/MPEG2 and H.261/H.263, which use block based motion estimation, a Full Search (FS) motion estimation takes up to 70-80 % of the encoding time. Most of the motion estimation algorithms used in video encoding belong to either Block Matching Algorithms (BMAs) or Pel Recursive Algorithms (PRAs). Majority of the current fast motion estimation methods [13],[5] employ a block matching algorithm and try to reduce the number of search locations in the search range. These algorithms are either ad hoc or make an assumption that the error increases monotonically from the best match location and thus they often end up in finding a local optimum. Another drawback of the matching algorithms is that they try to minimize the selected error measure and hence result in better coding efficiency even though the best match may not represent the actual motion for the particular block. In certain applications like motion compensated interpolation [3], it is more important to obtain the true motion information instead of the best match. PRAs try to estimate the true motion at each pixel and usually result in lesser coding efficiency than the BMAs.

Various fast block matching motion estimation algorithms such as 2-D Logarithmic search, Three Step Search (TSS), Conjugate Direction Search (CDS), One-at-a-time Search (OTS), New three Step Search (NTSS), Block Based Gradient Descent Search (BBGDS) [13] exist. Each of these algorithms can be represented as a point on the speedup versus PSNR plane. To compare our proposed fast motion estimation method, we chose BBGDS algorithm from this category of algorithms because of its superior performance over others as reported in [13].

The Pel Recursive Algorithms use the temporal and spatial difference between the pixel intensity in the previous and the current frame to estimate the interframe translation for each pixel. Previous approaches [14],[1] have mainly used a least squares, Wiener-based, or gradient descent recursive method. The Wiener-based and gradient descent methods operate on each pixel and use a causal neighborhood around it.

In our proposed method for fast motion estimation we start with the assumptions and methodology similar to the PRAs, but we operate on a block of pixels and find a single motion vector for each block. In order to achieve a fast motion estimation our proposed method operates on a block basis and is not recursive. We also propose to use the Total Least Squares (TLS) [8] estimation scheme which is motivated by the use of the previously reconstructed frame during the motion estimation. The TLS formulation will also result in a robust motion estimation when the video sequences are corrupted by noise. The motion estimation algorithms use previously reconstructed reference frame instead of the actual previous reference frame because the decoder only has the previously decoded frame to do the motion compensation. The previously reconstructed image can be considered as a “noisy" estimate of the actual previous frame. Furthermore, neglecting the higher order terms in the Taylor series expansion results in a truncation error. The TLS formulation tries to find the “true” motion in this case. It is known that if the errors are independent and Gaussian distributed in both the observation and the measurement data, then TLS solution is equivalent to the maximum likelihood estimator [2]. With the use of TLS, the motion estimation tries to find the true motion information and hence is expected to result in a better performance for the motion compensated interpolation type of applications. Also the obtained motion vector field from the proposed method will be smoother than that from the matching algorithms. Another motivation for using a pel recursive type

A PART OF THIS RESEARCH WAS PARTIALLY SUPPORTED BY A GRANT FROM INDUSTRIAL TECHNOLOGY RESEARCH INSTITUTE, TAIWAN, UNDER GRANT # 63-8550 AND ANOTHER GRANT FROM WASHINGTON TECHNOLOGY CENTER, UNDER GRANT # 63-8809.

of estimation scheme is that the estimated motion vectors have a fractional pel accuracy which can improve the performance of the video encoding schemes significantly when combined with pixel interpolation [6]. Half-pel accurate motion estimation for H.263 [9] has been the main reason for approximately 2 dB improvement in performance as compared to H.261 encoding as reported in [7].

Amongst the main contribution of this paper is an attempt to combine the best features of the block based methods, e.g. use of a single motion vector for each block to reduce the coding bits; without performing the actual block matching, and the best features of pel recursive methods, e.g. fractional pel accuracy; without performing any recursion in order to achieve a very fast motion estimation. Majority of the current video coding standards only support a block (macroblock) based motion estimation, hence the proposed method is suitable for use with any of these standards. The proposed fast motion estimation method results in an almost constant time for motion estimation for videos with different type of motion. To test the feasibility of this proposed approach we have done extensive simulations.

2. FAST MOTION ESTIMATION USING TOTAL LEAST SQUARES

The Pel Recursive Algorithms assume that image intensity in the current frame at a pel location (x, y) is related to the intensity from the previous frame by a displacement vector $(d1, d2)$, i.e. $f(x, y, t) = f(x - d_1, y - d_2, t - \Delta t)$, where t and $t - \Delta t$ are the time instants corresponding to the current and the previous frame respectively. Hence the frame difference $FD(x, y)$ at a pel (x, y) can be written as

$$\begin{aligned} FD(x,y) &= f(x,y,t) - f(x,y,t-\Delta t) \\ &= f(x,y,t) - f(x+d_1,y+d_2,t) \\ &= f(x-d_1,y-d_2,t-\Delta t) - f(x,y,t-\Delta t) \quad (1) \end{aligned}$$

We do a Taylor series expansion for $f(x - d_1, y - d_2, t - \Delta t)$ and obtain

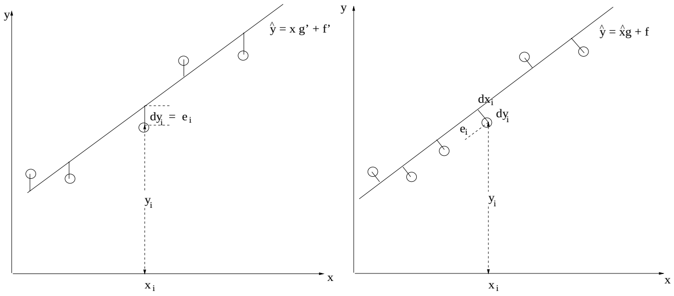
$$\begin{aligned} FD(x,y) &= -\nabla_x f(x,y,t-\Delta t)d_1 - \nabla_y f(x,y,t-\Delta t)d_2 \\ &\quad +HOT, \quad (2) \end{aligned}$$

where HOT denotes the higher order terms of the Taylor series expansion. Alternatively it is also possible to do a Taylor series expansion for $f(x + d_1, y + d_2, t)$. To achieve fast motion estimation and to make the algorithm compatible with the current block based motion vector supported video coding standards, we assume that all the pixels in a block in the current frame have the same displacement with respect to the previous frame. Thus the above equation can be written together for N (commonly $16 \times 16 = 256$) pixels in a block as

$$\begin{bmatrix} FD(x_1, y_1) \\ FD(x_2, y_2) \\ \cdot \\ \cdot \\ FD(x_N, y_N) \end{bmatrix} = - \begin{bmatrix} \nabla_x f(x_1, y_1, t - \Delta t) & \nabla_y f(x_1, y_1, t - \Delta t) \\ \nabla_x f(x_2, y_2, t - \Delta t) & \nabla_y f(x_2, y_2, t - \Delta t) \\ \cdot & \cdot \\ \cdot & \cdot \\ \nabla_x f(x_N, y_N, t - \Delta t) & \nabla_y f(x_N, y_N, t - \Delta t) \end{bmatrix} \times \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \mathbf{HOT}, \quad (3)$$

with **HOT** being an $N \times 1$ vector.

To compute the gradients in $N \times 2$ data matrix on the right hand side of Eq. (3) we use the previously reconstructed frame, which is available at the decoder. Hence it can be considered as a noisy measurement when we are interested in the actual motion between the current and the previous frame. Also since we ignore the **HOT**, the measurements are noisy. Similarly, the frame difference values on the left hand side of the same equation may be corrupted by sensor noise and also use the previously reconstructed frame. Eq. (3) can thus be formulated as $(\mathbf{b} + \partial\mathbf{b}) = (\mathbf{A} + \partial\mathbf{A}) \cdot \mathbf{d}$. To solve this equation for the motion vector $\mathbf{d}$ using the Total Least Squares (TLS) method [8], we have to minimize the cost $C = ||\partial\mathbf{A}||^2 + ||\partial\mathbf{b}||^2$, where $|| \cdot ||$ corresponds to the Frobenius norm. Figure 1 illustrates the difference between the use of the Total Least Squares (TLS) cost function and the usual ordinary Least Squares (LS) cost function. The problem is posed as the familiar straight line fitting for the given set of data points. As illustrated in Figure 1 (a) the LS cost function minimizes the cost $C_1 = ||\partial\mathbf{y}||^2$. From the figure it is clear that the LS cost function minimizes the sum of the squares of distances parallel to the y axis from each point to the best fitting line. This assumes that the error (noise) is confined to the y_i observations and the data points x_i are error free (noiseless). The TLS formulation for line fitting is more appropriate when both the y_i and x_i are erroneous (noisy). From the Figure 1 (b), the cost minimized in this case can be seen to be the sum of the squares of perpendicular distances from each point to the best fitting line. Thus the cost function minimized is $C_2 = ||\partial\mathbf{x}||^2 + ||\partial\mathbf{y}||^2$, which for the motion vector estimation problem becomes equal to the cost C given above. It should be noted that we use the term "noisy" while modeling the measurements and the observations in a rather broader manner similar to [4]. Because of this formulation the proposed method results in a robust estimation even when the video sequences are actually corrupted by noise (sensor noise).



(a) (b)

Figure 1: (a) The Least Squares (LS) and (b) the Total Least Squares (TLS) solution for the straight line fitting problem for data points.

The above cost minimization leads to the TLS solution obtained by taking the SVD of the augmented data matrix

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{b} \end{array} \right] = \mathbf{U\Sigma V}^T \quad (4)$$

where $\mathbf{U}$ is an $N \times 3$, $\mathbf{V}$ is a 3×3 orthonormal matrix and $\mathbf{\Sigma}$ is a 3×3 diagonal matrix. The TLS solution is obtained by using the last column of $\mathbf{V}$ matrix, which spans the null space of augmented data matrix. Writing the matrix $\mathbf{V}$ in partitioned form as

$$\mathbf{V} = \begin{bmatrix} \mathbf{V_{11}} & \mathbf{V_{12}} \\ \mathbf{V_{21}} & V_{22} \end{bmatrix}, \quad (5)$$

where $\mathbf{V_{11}}$ is a 2×2 sub matrix and V_{22} is a scalar. The TLS solution for the motion vector $\mathbf{d}$ is given by $\mathbf{d} = -\mathbf{V_{12}}V_{22}^{-1}$.

Since the Taylor series expansion is used in our proposed method, it is expected to be more accurate for small displacements (motion vectors). But if the scene contains large motion activity, a displaced frame difference (DFD) can be used instead of using the frame difference (i.e. Taylor series expansion about zero motion vector). With this modification, although the motion vectors may be large, the differential motion vectors (actual motion vectors minus the predicted motion vectors) will be small and can be estimated using Taylor series expansion.

3. SIMULATION RESULTS

All the simulations were carried out on the standard H.263 test sequences. We used the QCIF (176 × 144) video sequences “Miss America", “Trevor", “Suzie", and “Claire" at 30 fps. “Claire” test sequence had 490 frames where as the other sequences had 150 frames each. The first set of simulations compared the performance of the proposed motion estimation with the Full Search (FS), Block Based Gradient Descent Search (BBGDS) fast motion estimation and Wiener-based motion estimation. BBGDS was chosen for the comparison because it is one of the most well known fast motion estimation algorithms and also a comparison of its performance with respect to the other fast algorithms shows it superiority over the others [13]. Table 1 shows the performance comparison for various algorithms with respect to the average PSNR. The search range used was [-15,15]. It can be seen that TLS motion estimation has a comparable or better (due to fractional pel accuracy) performance with respect to the other motion estimation methods. The proposed method results in an almost constant time for motion estimation for any search range, where as the time for the search type of methods increases with the search range. All the algorithms perform better than Wiener-based method in terms of the average PSNR value. Table 2 shows the speed up ratio of various motion estimation methods compared with the FS method. The proposed TLS method can be seen to achieve a very high speed up in comparison with the other methods.

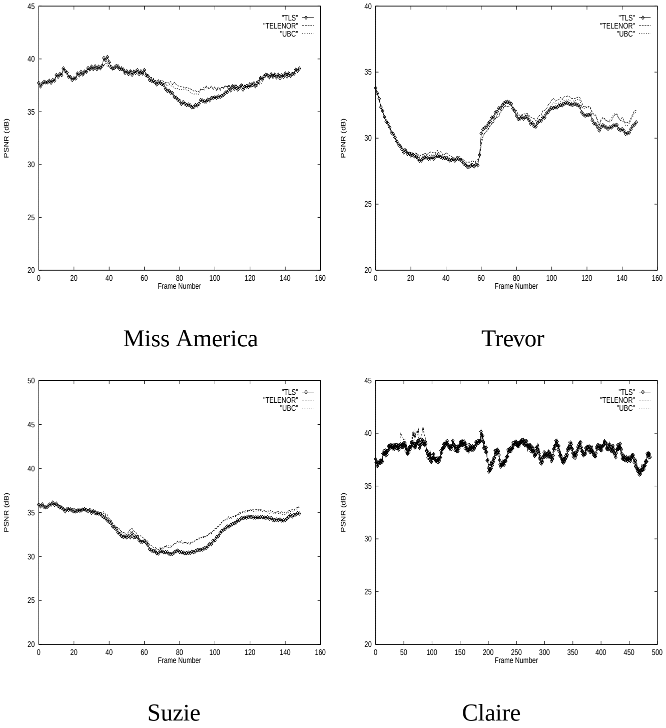
	BMA	BBGDS	Wiener	TLS
Miss America	41.38	41.31	40.17	41.60
Trevor	34.63	34.36	32.93	33.59
Suzie	36.34	35.70	33.94	35.50
Claire	42.87	42.82	42.16	43.12

Table 1: Performance comparison of various algorithms with respect to the average PSNR.

	BMA	BBGDS	Wiener	TLS
Miss America	1	56.46	60.54	99.00
Trevor	1	45.24	76.40	133.33
Suzie	1	44.44	62.34	89.28
Claire	1	68.02	69.70	75.75

Table 2: Speedup factors comparison of various algorithms with respect to the Full Search BMA.

We also implemented the proposed motion estimation method in a H.263 [9] encoder with half pel accurate motion estimation. This is a modified version of Telenor’s H.263 encoder [15] with the original motion estimation replaced by the proposed scheme. Telenor’s original encoder implements the motion estimation as per ITU TMN 5 model [10]. A constant bitrate of 36 kb/s was obtained using a rate control algorithm. We compare the speedup (Table 4) and average PSNR (Table 3) of our H.263 encoder with that of the original Telenor encoder and University of British Columbia’s (UBC) fast motion estimation encoder [5] (also a modified version of the original Telenor encoder) which uses motion estimation method from ITU TMN 8 model [11]. Figure 2 shows the average PSNR plots for all three encoders for all four sequences. Figure 3 shows the original and reconstructed frame number 100 for “Trevor" sequence using different encoders. From all the simulation results we observe that the proposed method is able to do very fast motion estimation and overall encoding at the expense of a slight degradation in PSNR value and with a comparable subjective quality.



Miss America

Trevor

Suzie

Claire

Figure 2 : Performance comparison plots for various existing H.263 video standard encoders, with different motion estimation algorithms.

	Telenor	UBC	TLS
Miss America	38.17	38.07	37.86
Trevor	30.88	30.75	30.58
Suzie	33.90	33.82	33.28
Claire	38.42	38.36	38.27

Table 3: Performance comparison of various existing H.263 encoders with respect to the average PSNR.

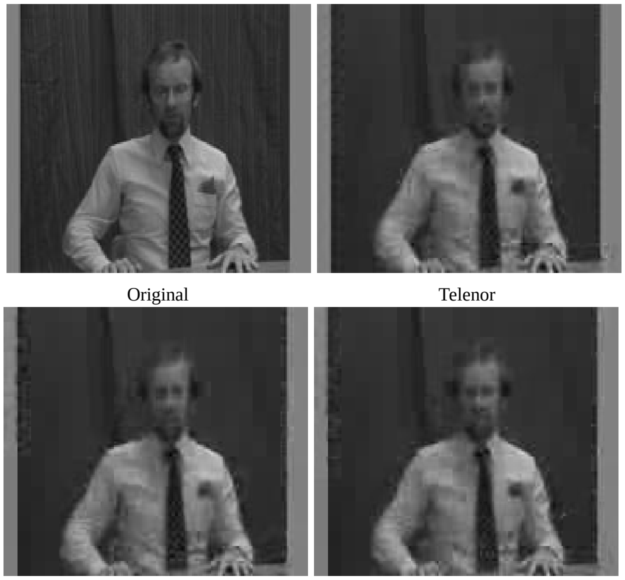
	Telenor	UBC	TLS
Miss America	1	35.05(5.21)	63.62(7.75)
Trevor	1	32.48(5.51)	71.22(10.41)
Suzie	1	30.55(5.51)	70.23(9.26)
Claire	1	26.86(4.16)	45.68(8.73)

Table 4: The speedup factors for the motion estimation part (and overall encoding) compared with the original Telenor’s H.263 encoder.

In a real application the video sequence may be corrupted by camera noise or some other noise sources (in the remote surveillance kind of application). Considerable research [12] has been done in processing of noisy image sequences. However most of the video compression motion estimation algorithms have not focussed on this aspect. The final set of simulations were done on the four test sequences with additive white Gaussian noise. A Gaussian noise with mean 0 and sigma 12 was added to all Y, Cb, and Cr components. The noise level was chosen as the maximum value which gives a reasonable decoded video quality for all four test sequences after encoding with H.263 encoders at 64 kb/s. To increase the motion activity between the successive frames, a temporal sampling factor of 3 was used for all the sequences. This is in order to illustrate the performance of the proposed method even when the amount of motion activity between the successive frames is rapid. We calculated two PSNR values for these simulations. The first values are between the noisy sequence and the decoded (reconstructed after encoding) sequence. Average PSNR degradation for TLS method for this case was 0.1 and 0.04 dB with respect to Telenor and UBC encoders. The second set of PSNR values are between the original (noiseless) sequence and the decoded sequence. These are given in Table 5. The speedup factors for the motion estimation and overall encoding are similar to those reported above for noiseless case and are not included again. As can be observed from Table 5 the proposed method achieves a slightly higher average PSNR than both Telenor's and UBC's encoder when the PSNR values are with respect to the original (noiseless) sequence. This is because our proposed TLS method is more robust in noisy environments. The average PSNR improvement is small, however this is obtained at the same time achieving an average speedup of 64 times and 9 times respectively for the motion estimation and overall encoding as compared to Telenor's encoder.

	Telenor	UBC	TLS
Miss America	28.97	28.90	29.01
Trevor	24.59	24.56	24.65
Suzie	26.34	26.36	26.51
Claire	25.05	25.10	25.12

Table 5: Performance comparison of various existing H.263 encoders with respect to the average PSNR for noisy video sequences. PSNR values are between the original (noiseless) and the reconstructed sequence.



Original

Telenor

UBC

TLS

Figure 3 : Original and reconstructed frame number 100 for "Trevor" sequence from different H.263 encoders.

4. CONCLUSIONS

We have proposed a Total Least Squares (TLS) based fast motion estimation method which is suitable for very fast video encoding in noisy environments. The method attempts to combine the advantages of block based algorithms and pel recursive algorithms for motion estimation. Extensive simulation results illustrate the very high speed of the proposed method over other existing methods. The proposed method was used to build a fast H.263 video encoder and its performance was compared with the currently available encoders. The TLS method is robust and results in overall superior performance in speed and PSNR compared to the other methods in a noisy environment.

References

[1] J.Biemond, L.Looijenga, D.E.Boekee, and R.H.J.M.Plompen, "A Pel-Recursive Wiener-Based Displacement Estimation Algorithm," *Signal Processing*, Vol. 13, No. 4, pp. 399-412, Dec. 1987.

[2] A.M.Bloch, "A Geometric Approach to Errors-in-Variables Models," in B.N.Datta *et al*, Eds., *Linear Algebra in Signals, Systems & Control*, Philadelphia, PA:SIAM, pp. 481-492, 1988.

[3] C.Cafforio, F.Rocca, and S.Tubaro, "Motion Compensated Image Interpolation," *IEEE Trans. on Communications*, Vol. 38, No. 2, pp. 215-222, Feb. 1990.

[4] Y.-M.Chou, and H.-M. Hang, "A New Motion Estimation Method Using Frequency Components," *Journal of Visual Communication and Image Representation*, Vol. 8, No. 1, pp. 83-96, Mar. 1997.

[5] G.Cote, M.Gallant, and F.Kossentini, "Efficient Motion Vector Estimation and Coding for H.263-Based Very Low Bit Rate Video Compression," `http://www.ee.ubc.ca:80/image`.

[6] B.Girod, "Motion Compensating Prediction with Fractional-Pel Accuracy," *IEEE Trans. on Communications*, Vol. 41, No. 4, pp. 604-612, Apr. 1993.

[7] B.Girod, N.Farber, and E.Steinbach, "Performance of the H.263 Video Compression Standard," Invited Paper, *J. VLSI Signal Processing*, (to appear).

[8] G.H.Golub and C.F.Van Loan, "An Analysis of the Total Least Squares Problem," *SIAM J. Numerical Analysis*, Vol. 17, No. 6, pp. 883-893, Dec. 1980.

[9] International Telecommuncation Union (ITU), *H.263 : Video Coding for Low Bitrate Communication*, Draft, May 1996-1997.

[10] International Telecommuncation Union (ITU) Standardization Sector, "Video Codec Test Model, TMN5," Jan. 1995.

[11] International Telecommuncation Union (ITU) Standardization Sector, "Video Codec Test Model, TMN8," June 1997.

[12] R.P.Kleihorst, R.L.Lagendijk, and J.Biemond, "Noise Reduction of Image Sequences Using Motion Compensation and Signal Decomposition," *IEEE Trans. on Image Processing*, Vol. 4, No. 3, pp. 274-284, Mar. 1995.

[13] L.-K.Liu and E.Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 4, pp. 419-422, Aug. 1996.

[14] A.N.Netravali and J.D.Robbins, "Motion-Compensated Television Coding : Part I," *Bell System Tech. Jnl.*, Vol. 58, No.3, pp. 631-670, Mar. 1979.

[15] Telenor Res. & Dev., "H.263 Encoder, Decoder Version 2.0," `http://www.nta.no/brukere/DVC/`, June 1996.