A fast motion estimation method using an enhanced motion vector and DC matching methodology

FARZIN AHMADIANPOUR AND M. OMAIR AHMAD
Department of Electrical and Computer Engineering
Concordia University
1455 De Maisonneuve Blvd. W.,
Montreal, Qc., CANADA H3G 1M8

Abstract: - A new fast block matching algorithm for motion estimation is introduced. This algorithm uses a new enhanced motion vector system combined with an effective DC matching technique. The proposed method transforms the expensive 2-D block-matching problem into a simpler 1-D matching by choosing some blocks as eligible candidates and eliminating the others. Test results of applying the proposed method on a number of MPEG video sequences are included. These results indicate that the proposed method not only can reduce the computational complexity by a factor ranging from 1 to 7, but also can reduce the prediction error by 2 to 20 percent as compared to the full search method. There are some cases that existing motion estimation techniques obtain a false motion vector with a huge prediction error, while the proposed method can find an enhanced motion vector with an excellent accuracy.

Key-Words: - Enhanced motion vector, Fast motion estimation, DC matching, Video compression, Video coding.

1 Introduction
The full search block-based algorithm, even though computationally intensive, is still considered to be the best in terms of quality and prediction performance [1]. Some of the previously proposed fast block-based methods are based on the notion that “very good” matches are likely to be found in the vicinity of reasonably good matches. There are a large number of algorithms that make this assumption and they may be classified as algorithms based on the principal of locality [2],[3]. One of the problems with these algorithms is that they can converge to a local minimum rather than to a global minimum [2].

There is another class of the algorithms that seek to exploit the natural spatial dependencies that exist in most images. In these algorithms, the motion vector for a block can be predicted based on the motion vectors of the blocks surrounding it [1]. Most of the existing methods are either relatively fast but have poor prediction accuracy or have fairly good prediction accuracy but are not fast enough.

There is a trade-off between the speed of a motion estimation algorithm and the quality of prediction image. Keeping this trade-off in mind, many fast motion estimation methods have been developed. These algorithms are suboptimal, since even though they are computationally more efficient but their quality is not as good as full search method [1]. Thus, there is a need to develop optimal algorithms that are faster and produce better quality than the full search technique.

Some algorithms reduce the computational complexity by using several stages. They use a different comparison criteria in each stage [6]. This paper presents a new fast two-stage motion estimation algorithm based on a new enhanced motion vector system and using a DC matching methodology, which not only requires less computation but also reduces the prediction error as compared to the full search method.

2 Proposed Scheme
To achieve the objective of designing an optimal motion estimation technique, a better way of
finding motion vector must be introduced; otherwise by using what we call the conventional motion estimation method, there is no technique better than the full search method in terms of quality.

In this section, the idea of an optimal motion estimation technique is introduced. The proposed optimal motion estimation method uses the new idea of the enhanced motion vector to obtain a motion vector with enhanced accuracy. To increase the speed of the algorithm, the proposed method uses a new DC matching algorithm. This leads to an optimal motion estimation technique that is faster and yet provides a better quality as compared to the full search method. The proposed technique is described in terms of quality and computational complexity. The experimental results of implementing the proposed technique are reported in the next section.

2.1 Enhanced motion vector
Block matching motion estimation methods estimate the amount of motion on a block-by-block basis i.e. for each block in the current frame, a block from previous frame as the best match and based on a certain criterion is found.

In an effort to find an enhanced motion vector, a constant value is added to all the pixels intensities of a block in the current frame before the search for finding the best match starts. In this case, the block in the present frame can be predicted from the block in the previous frame, the calculated motion vector and the constant value.

In this paper the difference between the DC value of the block in the current frame and the block in the previous frame is considered as constant value.

The enhanced motion vector considers the possibility of the shading and brightness changes for a movable object in the video sequences and explains the motion vector in a more sophisticated manner. As a result, the enhanced motion vectors produce less prediction error or a better quality for the predicted frame.

If we denote by $A_{ij}$ the intensity of pixels for a $n \times n$ block in the current frame, and by $B_{ij}$ the intensity of pixels for one of the blocks within a search window in the previous frame, then the prediction error using the conventional motion vector (PEC) can be calculated as

$$P_{EC} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (A_{ij} - B_{ij})^2$$  \hspace{1cm} (1)

The prediction error using the enhanced motion vector (PEE) can be calculated as

$$P_{EE} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (A_{ij} - B_{ij} - K)^2$$  \hspace{1cm} (2)

where $K$ is the DC difference of two blocks and can be calculated as

$$K = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (A_{ij} - B_{ij})$$  \hspace{1cm} (3)

It can be shown that the prediction error calculated by the enhanced motion vector is always smaller than or equal to the conventional motion vector. The difference between the prediction errors resulting from the two methods can be calculated as

$$P_{EE} - P_{EC} = -n^2 K^2$$  \hspace{1cm} (4)

2.2 Proposed Algorithm
The scheme of matching DC components for motion estimation has been introduced in many papers [4],[5]. The idea is simple i.e. blocks cannot match well if their corresponding DC components do not match well. By taking advantage of this observation, the expensive 2-D block matching problem is transformed into a simpler 1-D matching one, by quickly eliminating a majority of matching candidates. The DC matching [5] is based on this idea that two matched macroblocks must have a very similar sum of the pixel values (SPV). This method sorts the blocks by matching the SPV of the present frame with SPVs of macroblocks in the reference frame. Then $\alpha$ percent of all the blocks are kept for further consideration. In this paper, we develop this technique to be suitable with the proposed enhanced motion vector discussed earlier.

The block based motion compensation techniques divide a frame of $R \times C$ pixels into macroblocks of $N \times N$ pixels. Each of these
macroblocks is further divided into non-overlapping blocks of \( n \times n \) pixels which can be represented by \( m \times m \) SPVs with \( m = \frac{N}{n} \).

If \( F(i,j) \) is the intensity of the pixel with coordinates \( (i,j) \) in the present frame, \( M(i,j) \) the SPV of the block \( (i,j) \) in the present frame where each block of pixels is referred to by the coordinates of its upper left corner, then the SPVs for the present frame can be determined as

\[
M_t(i+kn,j+ln) = \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} F_t(iN+kn+p, jN+ln+q),
\]

where

\[
i = 0, N, 2N, 3N, \ldots, R-N
\]

\[
j = 0, N, 2N, 3N, \ldots, C-N
\]

\[
k, l = 0, 1, 2, \ldots, m-1
\]

\( R \) and \( C \) are, respectively, the height and the width of the frame.

The SPVs for the reference frame can be evaluated as

\[
M_{t-1}(i,j) = \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} F_{t-1}(i+p, j+q),
\]

where

\[
i = 0, 1, 2, \ldots, R-N
\]

\[
j = 0, 1, 2, \ldots, C-N
\]

The motion vector of a macroblock in the current frame can be found through a full search in the SPV domain, using the sum of square error (SSE) criterion which is defined as

\[
SSE_{i,j}(x,y) = \sum_{k=0}^{m-1} \sum_{l=0}^{m-1} \left[ M_t(i+kn,j+ln) - M_{t-1}(i+kn+x,j+ln+y) \right]^2
\]

where

\[
x, y = -D, -D+1, \ldots, D
\]

According to the SSE values, \( \alpha(2D+1)^2 \) candidates with the lowest SSE values and \( \beta(2D+1)^2 \) candidates with the highest SSE values are picked for further consideration, where

\[
0 < \alpha \leq 1, \ 0 < \beta \leq 1
\]

and \( (2D+1)^2 \) is the total number of blocks in the search window.

For each of the \( \alpha(2D+1)^2 \) and \( \beta(2D+1)^2 \) candidates, the MSE is evaluated based on the enhanced motion vector. The motion vector is finally determined by the relative position of the macroblocks in the reference frame, which minimizes the MSE.

The \( \beta(2D+1)^2 \) candidates are picked from the area that the DC difference between two blocks is big. The reason for choosing this subset of candidates is simple i.e. the big DC difference may happen due to a big change in brightness of the movable object which results in huge prediction error. Using enhanced motion vector for this subset of candidates, reduces the prediction error significantly.

### 2.3 Computational complexity

The computational complexity of the full search method is the direct consequence of the expensive 2-D block matching process. The full search algorithm requires \( 2N^2(2D+1)^2 \) additions for each motion vector, using the MSE as matching criterion. \( D \) is the maximum displacement. It must be mentioned that calculating of the MSE does not need any multiplication, since the squared values can be taken from predefined array.

The proposed technique requires the following computations:

- Calculation of the SPV in the reference frame requires almost \( 2nN^2 \) additions per motion vector.
- Calculation of SPV in the current frame requires \( N^2m^2 \) additions.
- full search in the SPV domain requires \( 2m^2(2D+1)^2 \) additions.
- Sorting the SPVs for \( \alpha(2D+1)^2 \) candidates requires almost \( \alpha(2D+1)^4 \) comparisons.
- Sorting the SPVs for \( \beta(2D+1)^2 \) candidates requires almost \( \beta(2D+1)^4 \) comparisons.
- Evaluation of the \( \alpha(2D+1)^2 \) candidates requires \( 2\alpha N^2(2D+1)^2 \) additions.
- Evaluation of the \( \beta(2D+1)^2 \) candidates requires \( 2\beta(N^2+2)(2D+1)^2 \) additions.
As a result, the number of required additions for the proposed method can be approximately calculated as

\[
2nN^2 + N^2 - m^2 + 2m^2(2D+1)^2 + 2\alpha N^2(2D+1)^2 + 2\beta(N^2+2)(2D+1)^2
\]

(8)

3 Simulation Results

The proposed technique has been tested on several MPEG sequences. The results are obtained based on the mean square prediction error (MSPE), which is the average energy per pixel in the residual image as given by

\[
\text{MSPE}(k) = \frac{1}{RC} \sum_{i=0}^{R-1} \sum_{j=0}^{C-1} (e_k(i,j) - \hat{e}_k(i,j))^2
\]

(9)

where R and C are, respectively, the height and the width of the frame and \(e(i,j)\), \(\hat{e}(i,j)\) are, respectively, the original and predicted pixel intensities at \((i,j)\)th position of the \(k\)th frame. The average mean square prediction error (AMSPE) is accordingly given by

\[
\text{AMSPE} = \frac{1}{K} \sum_{k=2}^{K+1} \text{MSPE}(k)
\]

(10)

where \(K\) is the total number of the frames tested.

In Fig.1 the MSPE obtained from using 51 frames of the MPEG Football sequence is presented. From this figure, it is seen that, the proposed technique always results in a better prediction error. Different parameters of the proposed method are set as \(D=7, N=8, m=2, \alpha=0.1\) and \(\beta=0\). With these parameter values, the proposed method is 6 times faster than conventional full search method.

Figs.2 and 3 show the relation of the \(\alpha\) and \(\beta\) to the prediction error for Football sequence. Obviously, increasing the value of \(\alpha, \beta\) increases the prediction accuracy, but the computational complexity is increased as well. Different parameters of the proposed method are set as \(D=7, N=16\) and \(m=4\).

Table 1 presents the AMSPE for various MPEG sequences. This table shows that on average the proposed method can reduce the prediction error by 10% as compared to the conventional full search method. \(\alpha\) and \(\beta\) are set to 0.16. The other parameters of the proposed method are set as \(D=7, N=8\) and \(m=4\).

To show the efficiency of the method, we replace frame 2 of the Missa sequence with it’s negative and frame 6 is replaced with a brighter frame. The MSPE results for the first 9 frames of this sequence are presented in Table 2. From these results, the proposed algorithm is effective in finding good motion vectors. These results show the efficiency of using the \(\beta(2D+1)^2\) candidates with the highest SSE. As it can be seen, changing \(\alpha\) from 0.1 to 0.3 (\(\beta=0\)), is not as efficient as choosing \(\alpha=0.1\) and \(\beta=0.16\).

4 Conclusion

The proposed block-matching algorithm reduces the computational complexity by using two stages. In first stage, all the blocks are evaluated using a DC matching criterion and then based on the result of this stage, two subsets of candidates are selected. One subset of candidates is picked up according to the lowest SSE values. The other subset of the candidates is chosen according to the highest SSE values. From these two subsets, the block with the minimum MSE is used to determine the enhanced motion vector.

According to simulation result, the proposed method is not only faster but also produces less prediction error as compared to the full search motion estimation technique. The proposed method offers computational scalability through two parameters, so the speed/performance trade-off can be easily controlled.

References: