CHAPTER 3

TEMPORAL REDUNDANCY REDUCTION METHODS

This chapter gives the motivation behind the improvement on block matching motion estimation algorithms. The proposed block matching algorithm along with implementation issues are presented. Experimental results conducted on various test sequences are also illustrated with charts and tables.

3.1 Introduction

In a conventional hybrid MCP DCT DPCM motion compensated predictive coding, the difference between the current and the motion compensated predicted frame is encoded [81-83]. The prediction process is accomplished by motion estimation and motion compensated prediction methods. Motion estimation is normally achieved by using any one of the BMAs. The fast block matching motion estimation algorithms discussed in the survey have addressed the following observations:

1. Efficient motion estimation algorithms developed based on providing less number of searching points in search window (W = ±7, ±15) and trying to obtain the position of moving object with minimum distortion. This intends to reflect in the construction of motion compensated predicted frame. This indeed helps to obtain residual frame. This residual frame is not only applied for spatial redundancy reduction and entropy encoding, it also decoded and brought into the frame memory for further process of predicting the subsequent frames of the input video sequence.
(ii) Fast block matching motion estimation algorithms such as 2DLOG, 3SS, 4SS, E3SS have the restricted number of searching steps in the search window to determine the position with minimum distortion measure. Of these algorithms, 2DLOG and 3SS have sparsely checking search points in the first step is very much suitable for video sequence with large motion. However, stationary or quasi-stationary video sequence can be easily trapped into a local minimum instead of leading to global minima, which degrades the performance of the motion estimation.

(iii) Moving object with minimum distortion measure is also determined in the search window by unrestricted number of searching points repeatedly by applying individually or the combination of different search patterns such as small and large rectangular, diamond and hexagon shapes.

Searching points of the candidate blocks in the search window of the reference frame are the main constraints in the fast block matching motion estimation algorithms. To increase the searching speed of the motion estimation algorithm, direction and value of the previous motion vector are considered for the motion vector of the current frame. This method consistently minimizes the searching speed as well as predicting the motion vector with minimum distortion measure.

3.2 Direction-based Block Matching

A simple relationship that exists between neighbouring frames of video sequence is harnessed in the proposed Direction-based Block Matching (DBM) algorithm to reduce the search time of block matching process.

3.2.1 Relationship between Neighbouring Frames

An object in a video sequence continues to move in the same direction or may be passive for a period of time. Turbulence is a rare phenomenon. Hence, for
comparison of the matching block and the target block, it is not required to search the entire set of candidate blocks in the search area. Instead, the candidate block to be searched can be predicted with a high probability of accuracy using the motion vectors of the previous frame as shown in Figure 3.1 (a). If the prediction coincides with the matching block, then the motion vectors of the matching block in the previous frame can be considered as the motion vectors of the target block in the current frame, else the motion vectors may require some refinement as shown in Figure 3.1 (b). This refinement of the motion vectors is achieved using any of the existing fast block matching algorithms with a smaller search area thus contributing to the reduction in time.

3.2.2 The DBM Algorithm

The DBM algorithm reduces search time by making use of motion vector positions using the relationship between frames. The reduction of search time and the prediction of motion vector are achieved as follows. The proposed algorithm involves two steps namely, prediction step and refinement step.

Prediction Step

The DBM algorithm utilizes the motion vectors of the previous frame to predict the motion vectors of the current frame. Before executing the prediction step, the predicted motion vectors for all the macro blocks of the current frame will be assigned as $(0, 0)$.

Consider the block $(i, j)$ of the both previous frame $I_{k-1}$ and current frame $I_k$. The values of motion vectors of $I_{k-1}$ (previous frame) are used to predict the values of motion vectors of $I_k$ (current frame). If the motion vector of the $k-1^{th}$ frame ($PMV_{k-1}(i, j))$ is $(m, n)$ then the Predicted Motion Vector (PMV) of the $k^{th}$ frame as illustrated in Figure 3.1 is given by

$$PMV_k\left(\frac{i + m}{M}, \frac{j + n}{N}\right) = (m, n)$$  \hspace{1cm} (3.1)
Figure 3.1  Direction-based Block Matching procedure: (a) Position of objects in k-1\textsuperscript{th} frame and k-2\textsuperscript{nd} frame with motion vector and (b) Search procedure of the DBM algorithm
Here, \( m/M \) signifies the number of blocks in the vertical direction, the object at position \((i, j)\) moved in the next frame. Similarly, \( n/N \) signifies the number of blocks in the horizontal direction.

The Equation (3.1) derived from the fact that the block \((i, j)\) in frame \( I_{k.1} \) is obtained by moving the block \((i - m/M, j - n/N)\) from the frame \( I_{k.2} \) by a distance \((m, n)\). The block \((i - m/M, j - n/N)\) in frame \( I_k \) is obtained by moving the block \((i, j)\) in frame \( I_{k.1} \) by a distance \((m, n)\) if the block continues to move in the same direction. This step is repeated for all the blocks in the current frame \( I_k \).

During the prediction step, some of the macro blocks may not be referred. For those macro blocks that are not referred, the prediction vectors will be \((0, 0)\). For such macro blocks, the corresponding matching will be searched during the refinement step to get their motion vectors. There is a chance for more than one predicted motion vector for the same macro block. In that case, the motion vector which gives minimum Mean Absolute Difference (MAD) is considered.

**Refinement Step**

Let \((m, n)\) be the predicted motion vector for the block \((i, j)\) in the current frame \( I_k \). The accuracy of the predicted vector can be improvised by refining:

(i) Compute \( \text{MAD}_{i,j} \), \((0, 0)\) and \( \text{MAD}_{i,j} \), \((m, n)\) for the current frame \( I_k \).

(ii) If \( \text{MAD}_{i,j} \), \((0, 0)\) is minimum, then the refinement is centered on the block \((i, j)\) else the refinement is done around the block \((i - m M, j - n N)\).

Most probable searching points used in recent existing fast BMAs are categorized as shown in Figure 3.2 (a). Fast block-matching motion estimation algorithms applying any one of the searching patterns to determine the motion vector for the candidate block of the current frame in the reference frame in a search window \( W = \pm 7 \) is illustrated in Figure 3.2 (b).
Pattern (I) - 8 search-points around center (11) : 1, 3, 5, 13, 21, 19, 17, 9
Pattern (II) - 8 search-points around center (11) : 9, 10, 12, 13, 3, 7, 15, 19
Pattern (III) - 8 search-points around center (11) : 6, 7, 8, 12, 16, 15, 14, 10
Pattern (IV) - 4 search-points around center (11) : 10, 7, 12, 15
Pattern (V) - 8 search-points around center (11) : 9, 6, 3, 8, 12, 16, 19, 14
Pattern (VI) - 6 search-points around center (11) : 9, 2, 4, 13, 20, 18

Figure 3.2 Refinement patterns of DBM: (a) Searching points in fast BMAs and (b) Different searching patterns

DBM algorithm is extended with its incremental versions namely, DBM1, DBM2, DBM3, DBM4 and DBM5. For convenience, these versions are referred to as *DBM schemes*. Thus in this thesis, the terms *schemes* and *algorithms* are used interchangeably. Various searching patterns used in refinement step of these schemes are explained hereunder.

**Searching patterns used in refinement step**

**DBM1**  All pixels in the small search window centered around the MAD $(m, n)$ region are searched with $(i - m \cdot M, j - n \cdot N)$ displacements to locate the best match block.
DBM2 - Apply searching patterns (V) and (IV) in refinement step

(i) Searching originates from centre and checks other eight search points of pattern (V). If the minimum MAD (MMAD) point calculated is at centre position, then go to step (iii) otherwise continue.

(ii) The MMAD determined in step (i) is re-assigned as the center for the new search with pattern (V). If the new MMAD is at the centre position, then continue otherwise repeat step (ii).

(iii) Apply pattern (IV) centered around the new MMAD to check all four search points for best matching block.

DBM3 - Apply searching patterns (II), (V) and (IV) in refinement step

(i) Searching originates from centre and checks other eight search points of pattern (II). If the minimum MAD (MMAD) point calculated as centre position, then goes to step (iii) otherwise continue.

(ii) The MMAD determined in step (i) is re-assigned as the centre for the new search with pattern (V) and checks other eight search points. If the new MMAD is at the center position, then continue otherwise recursively repeat step (ii).

(iii) Apply searching pattern (IV) centered around the new MMAD to check all four search points for best matching block.

DBM4 - Apply searching patterns (VI) and (III) in refinement step

(i) Searching originates from center and checks other six search points of pattern (VI). If the MMAD point calculated is center position, then go to step (iii) otherwise continue.

(ii) The MMAD determined in step (i) is re-assigned as the center for the new search with pattern (VI) and check other six search points. If the new MMAD is at the center position, then continue otherwise recursively repeat step (ii).

(iii) Apply pattern (III) centered around the new MMAD to check all eight search points for best matching block.
DBMS - Apply search patterns (I) and (IV) in refinement step

(i) Searching originates from centre and checks other eight search points of pattern (I). In addition to this, four more points of pattern (IV) are also checked for MMAD. If the MMAD point calculated is at center, it is found to be a best matching block and terminates the searching process. Otherwise, continue.

(ii) By reducing the large 9x9 search window size by half, the new MMAD determined in step (i) is re-assigned as the center for the new search with pattern (I) and check other eight points. If the new MMAD is at the center position, then continue otherwise repeat step (ii).

(iii) The new MMAD determined in step (ii) is re-assigned as the center for the new search with pattern (VI) and check all other four search points. If the MMAD point calculated is at center, it is found to be a best matching block and terminates the searching process. Otherwise recursively repeat step (iii).

Refinement is achieved by applying any one the above procedure at the minimum distortion position \((i, m, n)\) or \((0, 0)\) as the centre, but with minimum number of search points to reduce the motion estimation search time consistently.

3.3 Experimental Results and Discussion

The proposed DBM algorithm simulated using the video sequences are listed in Table 3.1. The simulation has been conducted for 14” frames of the “Bike” Source Input Format (SIF) sequence, which is a typical slow varying with bike object motion and most of the background objects are stationary or quasi-stationary. No foreign object intervention is anticipated in the video sequence. “Flower Garden” sequence with 14” frames consists of mainly stationary objects, but with a fast camera pan motion. There is a lot of new foreign object intervention in the middle of video sequence. The simulation was also conducted on 14” frames SIF “Table Tennis” sequence, which contains different combinations of still, slow,
panning and fast moving objects with camera zoom. "Football" sequence contains large displacement and fast local object motion, different combinations of still, slow and fast moving objects, camera zoom and panning.

In order to obtain the best performance of developed DBM schemes, experiments were carried out with different SIF and CIF sequences like "Bike", "Flower Garden", "Table Tennis" and "Football". The experimental results are explained hereunder.

Table 3.1  Video sequences used for simulations: Standard (Source Input Format (SIF) – 352x240 and Common Intermediate Format (CIF) – 352x288) video sequences

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Number of pixels/line</th>
<th>Number of lines</th>
<th>Frame rate</th>
<th>Video length (sec)</th>
<th>No. of frames*</th>
<th>BPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bike</td>
<td>352</td>
<td>240</td>
<td>30</td>
<td>4.9</td>
<td>147</td>
<td>24</td>
</tr>
<tr>
<td>Flower Garden</td>
<td>352</td>
<td>240</td>
<td>30</td>
<td>4.9</td>
<td>147</td>
<td>24</td>
</tr>
<tr>
<td>Table Tennis</td>
<td>352</td>
<td>240</td>
<td>30</td>
<td>4.9</td>
<td>147</td>
<td>24</td>
</tr>
<tr>
<td>Football</td>
<td>352</td>
<td>288</td>
<td>25</td>
<td>2.0</td>
<td>50</td>
<td>24</td>
</tr>
</tbody>
</table>

* Maximum number of frames considered for simulation

The video sequences considered for simulation consist of different degrees and types of motion contents. In all simulations, the Mean Absolute Difference (MAD) is the Block Distortion Measure (BDM), block size of 8x8, and search window \( W = 7 \) are considered.

The proposed DBM schemes are compared against optimal Full Search and sub-optimal fast BMAs such as DS, CDS, NHS and E3SS. The two important measures considered for analysis are average MAD per pixel and average number of search points per block.
Figure 3.3 A frame of “Bike” SIF sequence

Figure 3.3 shows an intraframe of “Bike” SIF sequence and the detailed frame by frame performance comparison among existing fast BMAs with proposed DBM schemes in terms of MAD per pixel and average search points per block are illustrated from Figure 3.4 to Figure 3.11.

Figure 3.4 Frame by frame performance comparison between FS and DBM1 for “Bike” SIF sequence on MAD per pixel
The brute-force exhaustive FS algorithm is compared with the proposed DBM algorithms with minimum distortion. Figure 3.4 shows the frame by frame comparison between FS and the proposed DBM1 algorithm in terms of MAD per pixel. It has been found that the DBM1 algorithm significantly achieves closer performance with Full Search in terms of prediction error from 0.12 to 2.15 in terms of MAD per pixel. MAD performance of DBM1 is compared with fast BMA like DS algorithm is shown in Figure 3.5. Unrestricted diamond-search pattern used in DS algorithm gives 0.016 to 1.356 more prediction error than DBM1 algorithm.

Figure 3.5  Frame by frame performance comparison between DS and DBM1 for "Bike" SIF sequence on MAD per pixel

Figure 3.6 shows the MAD performance of DBM1 with CDS. Cross-shaped search points used in CDS algorithm gives better performance with DS. However, it gives 0.008 to 0.886 more prediction error than the proposed DBM1 algorithm.
Figure 3.6  Frame by frame performance comparison between CDS and DBM1 for "Bike" SIF sequence on MAD per pixel

Figure 3.7  Frame by frame performance comparison between NHS and DBM1 for "Bike" SIF sequence on MAD per pixel
MAD performance of DBM1 is compared with NHS as shown in Figure 3.7. Hexagon-shaped search points used in NHS algorithm gives comparable performance with DBM1. However, it gives 0.010 to 0.925 more prediction error than the proposed DBM1 algorithm. Figure 3.8 shows the MAD performance of DBM1 with E3SS. Unrestricted searching points and small diamond shaped pattern used in E3SS algorithm gives 0.011 to 0.867 more prediction error than DBM1 algorithm.

Figure 3.8  Frame by frame performance comparison between E3SS and DBM1 for “Bike” SIF sequence on MAD per pixel

Figure 3.9 shows the overall performance of DBM4 algorithm with other sub-optimal fast BMAs in terms of average search points per block for first 50 frames. The comparison shows that DBM4 scheme outperforms all other fast BMAs by a minimum of 14 to a maximum of 21 search points. The DBM4 scheme is also found to be a fastest scheme among the proposed DBM schemes for “Bike” sequence.
Figure 3.9 Frame by frame performance comparison among DBM4 and other fast BMAs (for first 50 frames) for "Bike" SIF sequence on average search points per block.

Figure 3.10 shows the overall performance of DBM5 with other sub-optimal fast BMAs such as DS and CDS in terms of MAD per pixel with a minimum of 3.6920 to a maximum of 6.1616, which is comparable performance with CDS and outperforms DS. It also gives better performance in terms of average search points per block.

Figure 3.11 shows the overall performance of all DBM algorithms in terms of MAD per pixel and average search points per block. Out of these five DBM algorithms, DBM1 gives better performance on MAD per pixel and DBM4 outperforms than other algorithms in terms of average search points. From the Figure 3.11, it is observed that the DBM2 would be the best scheme among the proposed DBM schemes in terms of reasonable prediction error and average number of searching operations for "Bike" sequence.
Figure 3.10 Frame by frame performance comparison among DBM3 and other fast BMAs such as DS and CDS for "Bike" SIF sequence: (a) MAD per pixel and (b) Average search points per block.
Figure 3.11 Frame by frame performance comparison among five DBM algorithms (for first 59 frames) for "Bike" SIF sequence: (a) MAD per pixel and (b) Average search points per block.
Table 3.2 Performance comparison for “Bike” SIF sequence

<table>
<thead>
<tr>
<th>BMA</th>
<th>FS</th>
<th>DS</th>
<th>CDS</th>
<th>NHS</th>
<th>E3SS</th>
<th>DBM1</th>
<th>DBM2</th>
<th>DBM3</th>
<th>DBM4</th>
<th>DBM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAD per pixel</td>
<td>4.05</td>
<td>5.028</td>
<td>4.742</td>
<td>4.805</td>
<td>5.041</td>
<td>4.587</td>
<td>5.178</td>
<td>4.76</td>
<td>5.197</td>
<td>5.411</td>
</tr>
<tr>
<td>Average search points</td>
<td>225</td>
<td>47.72</td>
<td>45.99</td>
<td>32.21</td>
<td>41.04</td>
<td>49</td>
<td>22.86</td>
<td>44.43</td>
<td>16.83</td>
<td>31.67</td>
</tr>
</tbody>
</table>

Observations: DBM1 is 4.59183 times faster than FS, DBM2 is 9.84251 times faster than FS, DBM3 is 5.06414 times faster than FS, DBM4 is 13.36898 times faster than FS, and DBM5 is 7.10451 times faster than FS. It is also found from the above that DBM4 is a fastest scheme for “Bike” sequence over fast BMAs such as DS by 2.83541, CDS by 2.73262, NHS by 1.91384 and E3SS by 2.43850 times. Every scheme has the advantage of either minimum MAD or less number of search points for some of the fast BMAs. From the Table 3.2, it has been observed that DBM3 scheme outperforms DS in both MAD and NOP, and provides comparable performance with CDS in terms of MAD.

Figure 3.12 shows an intraframe of “Flower Garden” SIF sequence and the detailed frame by frame performance comparison among existing fast BMAs with proposed DBM schemes in terms of MAD per pixel and average search points per block are shown from Figure 3.13 to Figure 3.18.

Figure 3.12 A frame of “Flower Garden” SIF sequence
Figure 3.13  Frame by frame performance comparison among DBM1 and other fast BMAs for “Flower Garden” SIF sequence on MAD per pixel

MAD performance of DBM1 with other sub-optimal fast BMAs considering DS, CDS, NHS and E3SS is shown in Figure 3.13. Diamond-shaped search points used in DS algorithm gives 1.419 to 3.928 more prediction error values. where as CDS gives 0.032 to 4.521 more prediction error values than the proposed DBM1 scheme. Hexagon-shaped search points used in NHS algorithm performs 0.105 to 0.751 more prediction error values and E3SS gives 0.192 to 2.370 more prediction error values than DBM1 algorithm. MAD performance of DBM3 with other sub-optimal fast BMAs such as DS, CDS, NHS and E3SS is also shown in Figure 3.14. Diamond-shaped search points used in DS algorithm gives 1.221 to 3.969, where as CDS gets 0.004 to 4.660 more prediction error values than the proposed DBM3 scheme. Hexagon-shaped search points used in NHS algorithm gives 0.004 to 0.538 more error prediction values and E3SS gives 0.048 to 2.414 more prediction error values than DBM3 scheme.
Figure 3.14 Frame by frame performance comparison among DBM3 and other fast BMAs for “Flower Garden” SIF sequence on MAD per pixel

Figure 3.15 Frame by frame performance comparison among DBM2 and other fast BMAs for “Flower Garden” SIF sequence on average search points per block
Figure 3.15 gives the average number of search points per block among DBM2 and other faster BMAs. It is found that DBM2 is 1.84899 times faster than DS, 1.47046 times faster than CDS, 1.13970 times faster than NHS and 1.44478 times faster than E3SS algorithms.

Figure 3.16 Frame by frame performance comparison among DBM4 and other fast BMAs for “Flower Garden” SIF sequence on average search points per block

Figure 3.16 shows the average number of search points per block among DBM4 and other fast BMAs. It is found that the DBM4 scheme is 2.32708 times faster than DS, 1.85067 times faster than CDS, 1.43438 times faster than NHS and 1.81835 times faster than E3SS algorithms.

Figure 3.17 shows the frame by frame performance of DBM5 with other sub-optimal fast BMAs considering DS, CDS and E3SS in terms of MAD per pixel with a minimum of 12.456 to a maximum of 18.416 and also it predicts the required target block in the reference frame with faster rate over DS, CDS and E3SS. A comparable performance is also obtained with NHS on both MAD per pixel and average search points per block for “Flower Garden” SIF sequence.
Figure 3.17 Frame by frame performance comparison among DBMS and other fast BMAs for “Flower Garden” SIF sequence: (a) MAD per pixel and (b) Average search points per block
Figure 3.18 Frame by frame performance comparison among five DBM algorithms for "Flower Garden" SIF sequence: (a) MAD per pixel and (b) Average search points per block.

74
Figure 3.18 shows the overall performance of all DBM algorithms in terms of MAD per pixel and average search points per block. Out of these five DBM algorithms, DBM1 gives better performance on MAD per pixel and DBM4 outperforms other algorithms in terms of average search points per block. From the Figure 3.18, it is also observed that the DBM5 would be the best scheme among the proposed DBM algorithms in terms of reasonable prediction error and average number of searching operations for “Flower Garden” SIF sequence.

Table 3.3 Performance comparison for “Flower Garden” SIF sequence

<table>
<thead>
<tr>
<th>BMA</th>
<th>FS</th>
<th>DS</th>
<th>CDS</th>
<th>NHS</th>
<th>E3SS</th>
<th>DBM1</th>
<th>DBM2</th>
<th>DBM3</th>
<th>DBM4</th>
<th>DBM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average search points</td>
<td>225</td>
<td>36</td>
<td>28.63</td>
<td>22.19</td>
<td>28.13</td>
<td>49</td>
<td>19.47</td>
<td>30.9</td>
<td>15.47</td>
<td>24.23</td>
</tr>
</tbody>
</table>

Observations: DBM1 is 4.59183 times faster than FS. DBM2 is 11.55624 times faster than FS. DBM3 is 7.28155 times faster than FS. DBM4 is 14.54427 times faster than FS and DBM5 is 9.2860 times faster than FS. It is also found from the above that DBM4 is a fastest scheme for “Flower Garden” sequence over fast BMAs. Every scheme has the advantage of either minimum MAD or less number of search points over some of the fast BMAs. From the Table 3.3, it is also found that the DBM5 scheme outperforms DS, CDS, E3SS in terms of both MAD per pixel and average search points per block. and shows comparable performance with NHS in terms of MAD per pixel.

Figure 3.19 A frame of “Table Tennis” SIF sequence
Figure 3.19 shows an intraframe of "Table Tennis" SIF sequence and the detailed frame by frame performance comparison among existing fast BMAs with proposed DBM schemes in terms of MAD per pixel and Average search points per block are illustrated from Figure 3.20 to Figure 3.23.

![Graph showing performance comparison between FS and DBM1 for "Table Tennis" SIF sequence on MAD per pixel](image)

Figure 3.20 Frame by frame performance comparison between FS and DBM1 for "Table Tennis" SIF sequence on MAD per pixel

The brute-force exhaustive FS algorithm is compared with the proposed DBM1 algorithm in terms of MAD per pixel as shown in Figure 3.20. As per minimum MAD per pixel is concerned, the DBM1 algorithm is achieving very closer performance in terms of prediction error with minimum of 0.047, maximum of 2.371 and an average of 0.669 for 145 frames. From the Figure 3.20, it is observed that DBM1 error values are almost equal to FS from frame numbers 1 to 20 and 100 to 125.

MAD performance of DBM1 with other sub-optimal fast BMAs considering DS, CDS, NHS and E3SS is shown in Figure 3.21. Diamond-shaped search points used in DS algorithm gives 0.08 to 1.154, whereas CDS gives 0.003 to 1.240 more...
prediction error values than the DBM1 algorithm. Hexagon-shaped search points used in NHS algorithm gives 0.093 to 1.855 more prediction error values and E3SS gives 0.086 to 1.277 more prediction error values than DBM1 algorithm.

![Frame by frame performance comparison among DBM1 and other fast BMAs for "Table Tennis" SIF sequence on MAD per pixel](image)

**Figure 3.21 Frame by frame performance comparison among DBM1 and other fast BMAs for “Table Tennis” SIF sequence on MAD per pixel**

Figure 3.22 (a) shows the frame by frame performance of DBM5 with other sub-optimal fast BMAs such as DS, CDS, NHS and E3SS in terms of MAD per pixel. Diamond-shaped search points used in DS algorithm gives 0.003 to 1.115, where as CDS gives 0.003 to 1.384 more prediction error values than the DBM5 algorithm. Hexagon-shaped search points used in NHS algorithm gives 0.001 to 2.488 more prediction error values and E3SS gives 0.001 to 1.882 more prediction error values than the DBM1 algorithm.

The performance comparison on average search points per block is also shown in Figure 3.22 (b). The graph demonstrates that the DBM5 algorithm outperforms other sub-optimal fast BMAs by 1.60797 times faster than DS, 1.30564 times faster than CDS, 1.17607 times faster than NHS, and 1.37304 times faster than E3SS algorithm.
Figure 3.22 Frame by frame performance comparison among DBM5 and other fast BMAs for "Table Tennis" SIF sequence: (a) MAD per pixel and (b) Average search points per block.
Figure 3.23 Frame by frame performance comparison between DS and DBM3 algorithms for "Table Tennis" SIF sequence: (a) MAD per pixel and (b) Average search points per block.
Figure 3.23 (a) shows the frame by frame performance of DBM3 with DS in terms of MAD per pixel. Diamond-shaped search points used in DS algorithm gives minimum error of 0.001, maximum error of 1.334 and an average error of 0.111 over the proposed DBM3 algorithm. Figure 3.23 (b) shows the average search points per block to determine the best candidate in the search window \((W = \pm 7)\) of reference frame. The DBM3 algorithm outperforms DS by 1.06 times.

Table 3.4 Performance comparison for “Table Tennis” SIF sequence

<table>
<thead>
<tr>
<th>BMA</th>
<th>FS</th>
<th>DS</th>
<th>CDS</th>
<th>NS</th>
<th>E3SS</th>
<th>DBM1</th>
<th>DBM2</th>
<th>DBM3</th>
<th>DBM4</th>
<th>DBM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average search points</td>
<td>225</td>
<td>33.88</td>
<td>27.51</td>
<td>24.78</td>
<td>28.93</td>
<td>49</td>
<td>25.86</td>
<td>31.77</td>
<td>29.10</td>
<td>21.07</td>
</tr>
</tbody>
</table>

**Observations:** DBM1 is 4.59183 times faster than FS. DBM2 is 8.70069 times faster than FS. DBM3 is 7.08215 times faster than FS. DBM4 is 7.73195 times faster than FS and DBM5 is 10.67869 times faster than FS. From the Table 3.4, it is observed that the DBM5 scheme outperforms all fast BMA's and other DBM schemes in terms of MAD per pixel and average search points. Out of five DBM schemes, DBM1 gives better performance on MAD per pixel and DBM5 outperforms other algorithms in terms of average search points per block. Marginal improvement is also observed between DBM5 and DBM1 in terms of MAD per pixel. For the “Table Tennis” SIF sequence, DBM5 would be the best among the proposed DBM schemes in terms of reasonable prediction error and average search points per block.

Figure 3.24 A frame of “Football” CIF sequence
Figure 3.24 shows an intraframe of "Football" CIF sequence and the detailed frame by frame performance comparison among existing fast BMAs with proposed DBM schemes in terms of MAD per pixel and average search points per block are shown from Figure 3.25 to Figure 3.30.

The brute-force exhaustive FS algorithm is compared with the proposed DBM3 algorithm in terms of MAD per pixel as shown in Figure 3.25. As per minimum MAD per pixel is concerned, the DBM3 algorithm is achieving very closer performance in terms of error with minimum of 0.044, maximum of 1.146 and an average of 0.540 for 50 frames. From the Figure 3.25, it is also observed that DBM3 error values are close to Full Search algorithm for frame numbers from 8 to 23.

![Graph showing comparison between FS and DBM3 for MAD per pixel](image)

Figure 3.25 Frame by frame performance comparison between FS and DBM3 for "Football" CIF sequence on MAD per pixel
MAD performance of DBM3 with other sub-optimal fast BMAs such as DS, CDS, NHS and E3SS is shown in Figure 3.26. Diamond-shaped search points used in DS algorithm gives 0.015 to 0.630, where as CDS gives 0.005 to 2.851 more prediction error values than the proposed DBM3 algorithm. Hexagon-shaped search points used in NHS algorithm gives 0.025 to 0.759 more prediction error values and E3SS gives 0.046 to 0.606 more prediction error values than DBM3 algorithm. It is also found that the DBM3 gets very close performance for first 31 frames of CIF sequence and outperforms all other fast BMAs from 32 frames onwards.

Figure 3.26 Frame by frame performance comparison among DBM3 and other fast BMAs for “Football” CIF sequence on MAD per pixel

Figure 3.27 shows the average number of search points per block comparison among the DBM2 scheme and other sub-optimal fast BMAs. It is found that DBM2 scheme is 1.91402 times faster than DS, 1.68552 times faster than CDS, 1.19909 times faster than NHS and 1.92760 times faster than E3SS algorithms. Figure 3.28 gives the average number of search points per block comparison among the DBM4 scheme and other sub-optimal fast BMAs. It is observed that DBM4 scheme is 2.32417 times faster than DS, 2.04670 times faster than CDS, 1.45604 times faster than NHS and 2.34065 times faster than E3SS algorithms.
Figure 3.27 Frame by frame performance comparison among DBM2 and other fast BMAs for “Football” CIF sequence on average search points per block.

Figure 3.28 Frame by frame performance comparison among DBM4 and other fast BMAs for “Football” CIF sequence on average search points per block.
Figure 3.29 Frame by frame performance comparison among DBM3 and other fast BMAs such as DS and CDS for “Football” CIF sequence: (a) MAD per pixel and (b) Average search points per block.
Figure 3.30  Frame by frame performance comparison between DBM4 and CDS for "Football" CIF sequence: (a) MAD per pixel and (b) Average search points per block
Figure 3.29 shows the overall performance of DBM3 is compared with DS and CDS in terms of MAD per pixel and average search points. DS follows a diamond-shape for searching best candidate block in search window \((W = \pm 7)\) with a minimum value of 0.015 to a maximum of 0.630 more prediction error values than the DBM3 scheme. CDS apply cross-shaped search pattern for best candidate block search with a minimum value of 0.005 to a maximum of 2.851 more prediction error values than the DBM3 scheme. It is also observed that the DBM3 scheme is 1.13939 times faster than DS with almost similar speed with CDS in terms of average search points per block.

Figure 3.30 shows the overall performance of DBM4 with CDS in terms of MAD per pixel and average search points per block. CDS apply cross-shaped search pattern for best candidate block search with a minimum value of 0.021 to a maximum of 2.375 more prediction error than the DBM4 scheme. It is found that the DBM4 scheme is 2.0167 times faster than CDS algorithm in terms of average search points per block.

Table 3.5 Performance comparison for “Football” CIF sequence

<table>
<thead>
<tr>
<th>BMA</th>
<th>FS</th>
<th>DS</th>
<th>CDS</th>
<th>NHS</th>
<th>E3SS</th>
<th>DBM1</th>
<th>DBM2</th>
<th>DBM3</th>
<th>DBM4</th>
<th>DBM5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAD per pixel</td>
<td>4.023</td>
<td>4.616</td>
<td>5.318</td>
<td>4.666</td>
<td>4.599</td>
<td>4.896</td>
<td>5.993</td>
<td>4.564</td>
<td>4.986</td>
<td>5.653</td>
</tr>
<tr>
<td>Average search points</td>
<td>22.5</td>
<td>33.84</td>
<td>29.8</td>
<td>21.2</td>
<td>34.08</td>
<td>49</td>
<td>17.68</td>
<td>29.7</td>
<td>14.56</td>
<td>27.46</td>
</tr>
</tbody>
</table>

Observations: DBM1 is 4.59183 times faster than FS. DBM2 is 12.72624 times faster than FS. DBM3 is 7.57575 times faster than FS. DBM4 is 15.45329 times faster than FS and DBM5 is 8.19373 times faster than FS. It is also found from the above that DBM4 is a fastest scheme for “Football” CIF sequence over fast BMAs such as DS by 2.32417, CDS by 2.04670, NHS by 1.45604 and E3SS by 2.34065 times. Every proposed scheme has the advantage of either minimum MAD or less number of search points over some of the fast BMAs. From the Table 3.4, it is observed that DBM5 scheme outperforms all fast BMAs and other DBM schemes in terms of MAD per pixel and average search points per block. Marginal improvement is also observed between DBM5 and DBM1 in terms of MAD per pixel.
Table 3.6 Average MAD per pixel for different DBM algorithms and video sequences

<table>
<thead>
<tr>
<th>BMA</th>
<th>Bike</th>
<th>Flower Garden</th>
<th>Table Tennis</th>
<th>Football</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS</td>
<td>4.05</td>
<td>13.054</td>
<td>5.943</td>
<td>4.023</td>
</tr>
<tr>
<td>DS</td>
<td>5.028</td>
<td>16.305</td>
<td>6.895</td>
<td>4.616</td>
</tr>
<tr>
<td>CDS</td>
<td>4.742</td>
<td>15.033</td>
<td>7.144</td>
<td>5.318</td>
</tr>
<tr>
<td>NHS</td>
<td>4.805</td>
<td>14.0</td>
<td>7.355</td>
<td>4.666</td>
</tr>
<tr>
<td>E3SS</td>
<td>5.041</td>
<td>14.895</td>
<td>7.303</td>
<td>4.599</td>
</tr>
<tr>
<td>DBM1</td>
<td>4.587</td>
<td>13.591</td>
<td>6.613</td>
<td>4.896</td>
</tr>
<tr>
<td>DBM2</td>
<td>5.178</td>
<td>20.564</td>
<td>7.390</td>
<td>5.993</td>
</tr>
<tr>
<td>DBM3</td>
<td>4.76</td>
<td>13.735</td>
<td>6.782</td>
<td>4.564</td>
</tr>
<tr>
<td>DBM4</td>
<td>5.197</td>
<td>20.501</td>
<td>7.384</td>
<td>4.986</td>
</tr>
<tr>
<td>DBM5</td>
<td>5.411</td>
<td>14.017</td>
<td>6.695</td>
<td>5.653</td>
</tr>
</tbody>
</table>

Table 3.7 Average number of search points per block for different DBM algorithms and video sequences

<table>
<thead>
<tr>
<th>BMA</th>
<th>Bike</th>
<th>Flower Garden</th>
<th>Table Tennis</th>
<th>Football</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>DS</td>
<td>47.72</td>
<td>36</td>
<td>33.88</td>
<td>33.84</td>
</tr>
<tr>
<td>CDS</td>
<td>45.99</td>
<td>28.63</td>
<td>27.51</td>
<td>29.8</td>
</tr>
<tr>
<td>NHS</td>
<td>32.21</td>
<td>22.19</td>
<td>24.78</td>
<td>21.2</td>
</tr>
<tr>
<td>E3SS</td>
<td>41.04</td>
<td>28.13</td>
<td>28.93</td>
<td>34.08</td>
</tr>
<tr>
<td>DBM1</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>DBM2</td>
<td>22.86</td>
<td>19.47</td>
<td>25.86</td>
<td>17.68</td>
</tr>
<tr>
<td>DBM3</td>
<td>44.43</td>
<td>30.9</td>
<td>31.77</td>
<td>29.7</td>
</tr>
<tr>
<td>DBM4</td>
<td>16.83</td>
<td>15.47</td>
<td>29.10</td>
<td>14.56</td>
</tr>
<tr>
<td>DBM5</td>
<td>31.67</td>
<td>24.23</td>
<td>21.07</td>
<td>27.46</td>
</tr>
</tbody>
</table>
Tables 3.6 and 3.7 gives the overall comparison among all optimal and sub-optimal fast BMAs, and proposed DBM schemes for four different SIF and CIF sequences. Statistical comparisons given in the tables show that the proposed DBM schemes outperform the existing fast BMAs in terms of MAD per pixel and average search points per block.

3.4 Summary

It is observed from the experimental results that the developed DBM algorithm with different schemes outperform all other fast BMAs in terms of distortion measures such as MAD per pixel and average search points per block. The developed motion estimation algorithms also outperform the optimal Full Search algorithm in terms of search points from 6.428 to 15.453 times faster for four different standard SIF and CIF sequences.