CHAPTER 4

BLOCK MOTION ESTIMATION ALGORITHM

Summary

In this chapter, a new block motion estimation algorithm using dynamic pattern search (DPS) has been proposed. This algorithm exploits spatial and temporal coherence among macroblocks to precisely estimate motion vector of a candidate block. Further, it uses a halfway stop technique to increase its speed. The advantages of the this algorithm over conventional diamond search (DS) and hexagonal search (HS) methods have also been discussed. Proposed DPS has further been modified to reduce the number of search points which leads to reduction in computation cost.

4.1 Introduction

As discussed in chapter 1, motion estimation plays a vital role in a video encoder and block based motion estimation algorithms are most commonly used for this purpose. Full search algorithm is an optimal choice but it fails in real life because of its large computation. A number of algorithms have been proposed in literature (J.R.Jain and A.K.Jain (1981), Ghanbari (1990), T.Koga et al. (1981), Li et al. (1994), Srinivasan and Rao (1985), Liu and Feig (1996), Po and Ma (1996), Zhu and Ma (2000), Cheung and Po (2002), Zhu et al. (2002), Zhu et al. (2004), Cheung and Po (2005), Xu et al. (1999), Jing and Chau (2004), Chung and Chang (2003), Chan and Siu (2001), Liu and Zaccarin (1993), Yu and Tai (2006), Po et al. (2007), Wang and Kuroda (1999), Cheng and Sun (1999) etc.) which reduce computation time and claim quality results. In most of these algorithms, the assumption is that matching error monotonically decreases towards the location of global minima and error surface is uni–model. Moreover, size of search pattern for each macroblock is fixed.

One of the motion estimation algorithm is based on diamond search (Zhu and Ma (2000)) which uses a diamond shape pattern for estimating motion vector of each block.
The algorithm is faster and produces quality results than some older techniques like TSS (T.Koga et al. (1981)), NTSS (Li et al. (1994)), 4SS (Po and Ma (1996)) etc. by using small and large diamond search patterns for low and high motion respectively. Later a hexagon search algorithm, has been proposed (Zhu et al. (2002)) using hexagon as its search pattern. Computation time of hexagon search technique is less than that of diamond search technique without compromising the output results. In real video inputs, the possibility of a block to have similar motion than its adjacent blocks is very high, but spatial and temporal correlation has not been used in both of these algorithms, therefore, outcome of these algorithms may suffer from oversearch or undersearch problem as they may get trapped in local minima.

Keeping these observations in mind, in this chapter, a fast dynamic pattern search based BMEA has been proposed (R.K.Purwar et al. (2009c)). In the proposed DPS, instead of exclusively using fixed size search pattern, correlations between the current block and its spatial/temporal adjacent blocks is used to dynamically determine the size of search pattern keeping computation as low as possible. In section 4.2, conventional diamond search algorithm has been explained which is followed by hexagonal search algorithm in section 4.3. Proposed DPS algorithm for motion estimation is discussed in section 4.4. In section 4.5, modified form of proposed DPS is presented which is followed by conclusion in section 4.6.

4.2 Diamond Search Algorithm

In literature, it has been found that up to 98.70% motion vectors are centered at the position of zero motion and enclosed within a circle of radius 2 around it. Zhu and Ma (2000), simulated this assumption and proposed a diamond search (DS) pattern algorithm which uses two search patterns as shown in fig. 4.1 – large diamond search pattern (LDSP) and small diamond search pattern (SDSP) consisting of nine and five search points respectively.

As shown in the flowchart of fig. 4.2, searching process starts with LDSP and it is used repeatedly until minimum distortion point coincides with the center point of this pattern. In next step, SDSP is employed with its center coinciding with the minimum
Figure 4.1: Search Patterns used in DS algorithm (a) LDSP (b) SDSP

distortion point found at the center of LDSP in previous step. All five points of SDSP are explored and the location of minimum point is declared as the motion vector. This algorithm is summarized as follows.

1. Initial LDSP is centered at the origin of the search window and the 9 checking points of this pattern are tested. If minimum distortion point is located at the center position, go to step 3, otherwise go to step 2.
2. Minimum distortion point found in previous search step is treated as the center of new LDSP and the process is repeated as in step 1. If center point is the new minimum distortion point, go to step 3, otherwise repeat this step.
3. SDSP is used and all five points are tested. The minimum distortion point found in this step represents the best matching block.

4.3 Hexagon Search Algorithm

It has been found that DS produces good results than other BMEAs like TSS, NTSS or 4SS but it has its drawbacks. It is sensitive to (especially LDSP) motion vectors in different directions, that is, it has non uniform speed as well as number of search points for different directions. Further, it has redundant search points to be processed in each step. Reason behind these drawbacks is that diamond shape does not approximate enough to a circle.
Set center of initial LDSP at the center of the search window and check its nine points for finding minimum distortion point.

Is minimum distortion point located at the center of LDSP?

Yes

No

Set center point of new LDSP at the minimum distortion point found in previous step and check new LDSP points.

Switch the search pattern from LDSP to SDSP and check its five points to find final minimum distortion point.

Figure 4.2: Flow chart of Diamond Search Algorithm
Zhu et al. (2002) proposed a hexagon search (HS) pattern based block motion estimation where authors could succeed to remove above drawbacks up to some extent. HS as shown in fig. 4.3, consists of two search patterns – large hexagon search pattern (LHSP) and small hexagon search pattern (SHSP) with 7 and 5 points respectively.

![Hexagon Search Patterns](image)

(a) Large Hexagon Search Pattern  
(b) Small Hexagon Search Pattern

Figure 4.3: Search Patterns used in HS algorithm (a) LHSP (b) SHSP

It can be seen that all six points around center of LHSP are uniformly distributed and every next search step includes only three non overlapping points to be evaluated unlike DS which uses 3 or 5 such points depending on the location of minimum point to be the diagonal or corner point respectively. Therefore, HS is not direction sensitive to the motion vector.

BMEA using HS pattern is summarized below –

1. Initial LHSP is centered at the origin of the search window and the 7 checking points of this pattern are tested. If minimum distortion point is located at the center position, go to step 3, otherwise go to step 2.

2. Minimum distortion point found in previous search step is treated as the center of new LHSP and the process is repeated as in step 1. If center point is the new minimum distortion point, go to step 3, otherwise repeat this step.

3. SHSP is used and all five points are tested. The minimum distortion point found in this step represents the best matching block.

Corresponding flow chart of HS algorithm is shown in fig. 4.4.
Set center of initial large hexagon search pattern at the center of the search window and check its seven points for finding minimum distortion point.

Is minimum distortion point located at the center of hexagon pattern?

No

Set center point of new large hexagon search pattern at the minimum distortion point found in previous step and check its points for least distortion.

Yes

Switch the search pattern from large hexagon search pattern to small hexagon search pattern and check its five points to find final minimum distortion point.

Figure 4.4: Flow chart of Hexagon Search Algorithm
4.4 Dynamic Pattern Search (DPS) Algorithm

It has been observed that in case of small motion vector small search patterns (made of dense search points) are suitable but for large motion vector, these patterns may be trapped in either local minima or may require a number of unnecessary intermediate searches. On the other hand, large search patterns have advantage of quickly determining large motion vectors but may cause unnecessary searches for small motion. In other words, search pattern and magnitude of motion vector of a block are correlated and therefore it is desirable to have a dynamic search pattern for each block depending on its motion information.

In this section, one such dynamic search pattern based technique (R.K.Purwar et al. (2009c)) has been proposed which is based upon spatial and temporal coherence of a block. Implementation of the proposed technique, known as dynamic pattern search (DPS) requires prediction of motion information of the current block of interest as input. In subsection 4.4.1, technique to predict motion information of the current block has been discussed whereas in section 4.4.2, method of calculating size and shape of search pattern using estimated motion vector is explained. Complete DPS technique has been discussed in section 4.4.3.

4.4.1 Prediction of motion information of the current block

In video sequences, due to coherence property, motion of a particular block is quite similar to the motion of its spatially and temporally adjacent blocks, if both are part of the same object. Therefore, estimated motion vector of the current block may be same as motion vector of the corresponding block in the previous frame (because of the temporal coherence) or it may be same as one of the block in 8 neighborhood in the current frame (because of spatial coherence). At the time of motion estimation of current block, estimated motion of only four (top left, top, top right, left) out of 8 neighborhood in the current frame as shown in fig. 4.5 and corresponding block in the previous frame will be known. Out of these five estimated motion vectors known, for simplicity and low computation, the motion vector of immediate left block in the current
frame along with motion vector of corresponding adjacent block in previous frame has been used as two possible motion approximations.

![Diagram of four spatially adjacent blocks](image)

Figure 4.5: Four spatially adjacent blocks of a current block, processed in raster scan order

### 4.4.2 Selection of Search Pattern Size

Two predicted motion vectors obtained from previous section are split into horizontal and vertical components. Largest of these four components (two vertical and two horizontal) is used to calculate the search pattern size using eq. 4.1.

\[
s = \max(MV_{S(x)}, MV_{S(y)}, MV_{T(x)}, MV_{T(y)})
\]

(4.1)

where \((MV_{S(x)}, MV_{S(y)})\) and \((MV_{T(x)}, MV_{T(y)})\) are horizontal and vertical components of adjacent left block’s motion vector in the current frame \((MV_S)\) and that of corresponding block’s motion vector in the previous frame \((MV_T)\) respectively.

For the leftmost block in each frame, since there is no immediate left spatial adjacent block, only temporal adjacent block is considered. Further, for the I frame blocks, only spatial left adjacent block will be taken into account and for the leftmost blocks in this frame, size of dynamic pattern is chosen as 2 which is the maximum length of LDSP in DS or LHSP in HS algorithms respectively. It is possible that either or both \(MV_S\) and \(MV_T\) may overlap with any of the four symmetrical vertices or among themselves.
and in that case, number of search points will be reduced. Briefly, total number of search points for DPS will be either seven (no overlap) or six (one overlap) or five (two overlap) in its first iteration. Further, it will be reduced to only center point when size \( s \) in eq. 4.1 is zero.

![Figure 4.6: Vertical and horizontal components of a vector](image)

**4.4.3 Proposed DPS algorithm**

Proposed DPS algorithm for motion estimation has following steps.

1. Find motion vectors of left adjacent block in the current frame \( (MV_S) \) and corresponding block in the reference frame \( (MV_T) \).
2. Compute vertical and horizontal components of \( MV_S \) and \( MV_T \) i.e. \( ((MV_{S(x)}, MV_{S(y)}) \) and \( (MV_{T(x)}, MV_{T(y)})) \) as shown in fig. 4.6.
3. Take \( s \) as maximum of \( ((MV_{S(x)}, MV_{S(y)}), (MV_{T(x)}, MV_{T(y)})) \) using eq. 4.1 as size of dynamic search pattern. If \( s = 0 \), search pattern degenerates into single point.
4. Create a dynamic pattern search using four symmetrical horizontal and vertical points (shown as A,B,C and D in fig.4.7) at distance \( s \) from center and two motion vectors \( MV_S \) and \( MV_T \) as shown in fig. 4.7.
5. Find minimum distortion point among them using block matching criterion discussed in chapter 3. If it is same to \( MV_S \) or \( MV_T \), declare motion vector of the current block as location of minimum distortion point and go to step 7.
6. Set minimum distortion point as center of new unit size search pattern including four symmetrical adjacent points in vertical and horizontal directions and check these new

54
Figure 4.7: DPS with spatio temporal motion vectors

points. If new minimum distortion point is not same to the center point of the current search pattern and the search area boundary is not met, recursively use this step, otherwise motion vector of the current block is set equal to the location of center point of the current search pattern and go to step 7. Further, this step will process only four additional points in each iteration as center point has already been evaluated in previous iteration.

7. Exit.

Figure 4.8 shows motion estimation of a particular block using DPS algorithm for a search area of 15x15. Size of DPS is 2 which is the maximum length of all four components of motion vectors – $MV_S$ and $MV_T$. In the 1st iteration, seven points (labelled as 1
Figure 4.8: Motion vector estimation using DPS algorithm
and shown using diamond symbols) are processed to find minimum distortion point. Let the new minimum distortion point is found at location (0,−2) (assuming center point as (0,0)) is taken as center of unit size pattern with four additional search points (labelled as 2 and shown using star symbols) are processed to again find minimum distortion point. After this iteration, minimum point is found at location (−1,−2) which is taken as center of new unit size search pattern with additional points (labelled as 3 and shown using circles) which are explored for minimum distortion point. It has been found after this iteration that minimum distortion point coincides with the center point. So search is stopped and final predicted motion vector of the block is declared as (−1,−2). Entire search path for this motion vector is shown using dotted lines in fig. 4.8.

4.5 Modified DPS Algorithm

DPS algorithm for motion estimation discussed in previous section uses spatial and temporal correlations among macroblocks to find motion vector. This algorithm is faster than conventional DS and HS algorithms. In this section, a modified version of proposed DPS algorithm is being discussed which further improves its speed with almost same quality results.

In proposed DPS algorithm, one can find out that it evaluates 7 search points in its first iteration (four symmetrical points A,B,C,D, two adjacent motion vectors and a center point as shown in fig 4.7). It has been found in literature review (Chung and Chang (2003)) that there is 94% probability that motion vector of a candidate block will lie within a distance of ±2 from the motion vectors of its spatially adjacent blocks. It means if the differential distance between motion vectors of spatial and temporal coherent blocks discussed in previous section is less than or equal to ±2 or if the angle between spatial and temporal motion vectors is less than or equal to $\pi/4$, there is 94% probability that they belong to the same object and motion vector of candidate block will be same to either $MV_S$ or $MV_T$, depending upon which one is the minimum distortion point. In other words, if the angle between these motion vectors is not greater than $\pi/4$, only two points (locations of these motion vectors) need to be evaluated in the first iteration instead of seven points as there is higher probability that the candidate
block and its spatial and temporal coherent blocks belong to the same object. Further, if above angle is greater than $\pi/4$, corresponding blocks may belong to different objects and five points (four symmetrical points along with center) can be evaluated for minimum distortion points in the first iteration of DPS. Therefore, DPS algorithm is modified to reduce search computation and its modified form is presented below.

1. Find motion vectors of left adjacent block in the current frame ($MV_S$) and corresponding block in the reference frame ($MV_T$).
2. Find angle $\theta$ between these vectors using eq. 4.2.
3. If this angle is less than or equal to $\pi/4$ (which will cover differential distance of $\pm 2$), use only $MV_S$ and $MV_T$ as candidate search points to find minimum distortion point and motion vector will be the location of minimum point. Go to step 9. If angle is more than $\pi/4$, go to step 4.
4. Compute vertical and horizontal components of $MV_S$ and $MV_T$ i.e. ($MV_{S(x)}, MV_{S(y)}$) and ($MV_{T(x)}, MV_{T(y)}$) as shown in fig. 4.6.
5. Take $s$ as maximum of ($MV_{S(x)}, MV_{S(y)}), (MV_{T(x)}, MV_{T(y)})$ using eq. 4.1 as size of dynamic search pattern. If $s = 0$, search pattern degenerates into single point.
6. Create a dynamic pattern search with five points – four symmetrical horizontal and vertical points (shown as A,B,C and D in fig.4.7) at distance $s$ from center along with the center point.
7. Find minimum distortion point using these five points.
8. Set minimum distortion point as center of new unit size search pattern including four symmetrical adjacent points in vertical and horizontal directions and check these new points. If new minimum distortion point is not same to the center point of the current search pattern and the search area boundary is not met, recursively use this step, otherwise motion vector of the current block is set equal to the location of center point of the current search pattern and go to step 9. Further, this step will process only four additional points in each iteration as center point has already been evaluated in previous iteration.

Angle between vectors $MV_S$ and $MV_T$, used in step 2, is found by using following
\[ \theta = \cos^{-1} \frac{MV_S \cdot MV_T}{|MV_S||MV_T|} \]  

(4.2)

where $MV_S \cdot MV_T$ is the dot product of $MV_S$ and $MV_T$ which is given by

\[ MV_S \cdot MV_T = MV_{S,x} MV_{T,x} + MV_{S,y} MV_{T,y} \]

and $|MV_S|$ and $|MV_T|$ are magnitude of these vectors.

### 4.6 Conclusion

Fast block based motion estimation algorithms have been discussed in this chapter. A dynamic pattern search based motion estimation algorithm has been proposed which uses the spatial and temporal coherence among macroblocks to find motion vector of a candidate block. The proposed algorithm has used a halfway stop technique to make it faster. A modified version of proposed DPS has also been described to further make it faster. Experimental results showing substantial improvements using proposed algorithm over conventional DS and HS algorithms are shown in chapter 5.