Fast Motion Estimation in Video Coding

Kun-Chih Tsai
Electrical Eng. Dept.
National Taiwan Ocean University
2 Penning Rd., Keelung, Taiwan
andrue@celab1.ee.ntou.edu.tw

Jung-Hua Wang
Electrical Eng. Dept.
National Taiwan Ocean University
2 Penning Rd., Keelung, Taiwan
jhwang@mail.ntou.edu.tw

Abstract— A novel algorithm for improving block-matching motion estimation is presented. Two different search patterns, namely the Large Cellular Search Pattern (LCSP) and the Small Cellular Search Pattern (SCSP), are employed to determine the best matching block. We show that the proposed algorithm is computationally efficient, and it requires less computation time than other fast search algorithms such as the three-step search (TSS) [4], the new three-step search (NTSS) [2], and the four-step search (4SS) [1].

Keywords— Motion estimation, Video compression, Cellular search, Full search.

1. Introduction

Modern video compression standards, such as MPEG-1, MPEG-2, MPEG-4, and H.26x, all employ some sort of motion compensation techniques in order to reduce the temporal redundancy between frames (i.e., intra-frame residual). It is widely known that Motion estimation plays an essential role in the performance of motion compensation. In general, motion estimation accounts for nearly 35% of total computation time required in a MPEG2 encoder. It is in fact the most computation demanding module in implementing MPEG series. For this reason, many research efforts have been directed to improving the performance of motion estimation. Among them, a full search (FS)[1] strategy has been shown useful in determining the accurate motion displacement of the prediction, because it calculates every possible motion displacement to obtain the *best* possible value. This assumes, however, that the criterion for *best* is known *a priori*, it also implies that the computational load for a full search can be impractically heavy for implementing a real-time MPEG video encoder.

Due to the heavy computation load in the Full Search (FS) method, availability of fast search algorithms has become increasingly important in the last decade. For example, three-step search (TSS) [4], new three-step search (NTSS) [2], four-step search (4SS) [1], and diamond search (DS) [3] algorithms have been proposed for this purpose.

In this paper we present a novel algorithm called Cellular Search (CS) that is useful for block-matching motion estimation in real-time video coding systems. We show that the average number of searched blocks by using CS is less than those by using other fast search algorithms.

2. Motion Estimation

A. The Error Measure

To determine the best matching block, a block distortion measurement (*BDM*) is commonly used in motion estimation. For an $N \times N$ search window, values of *BDM* of all points in the search window form an error surface defined as the following

$$E(u,v) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} | f_{t-\tau}(i+u, j+v) - f_t(i,j) |,$$

$$-W \le u, v \le W \qquad (1)$$

where $f_{t-\tau}$ and f_t refer to the blocks in the precedent frame and the current frame, respectively, and (u,v) denotes the possible motion vector location where the best matching block B_m resides. Window W denotes the maximum motion in vertical and horizontal directions. That is, the movement of a block may lie within a $(2W+1)^2$ window.

B. The Search Patterns

For most motion estimation algorithms, both the accuracy and search speed rely on the search patterns used. To explore the whole search window, scan area of the search pattern should be as wide as possible, and the scanning speed as fast as possible. In order to achieve this goal, many algorithms such as the TSS, NTSS, 4SS, and DS employed two or more different search patterns at different stage. They assumed that in a small area (search window) of a moving sequence, the block distortion measurement increases monotonically as the block located farther away from the best matching block B_m. Furthermore, the best matching block B_m can be located at any direction, thus the search pattern should move toward B_m as fast as possible.

The above two observations lead us to propose the CS algorithm, which adopted two search patterns named Large Cellular Search Pattern (LCSP) and Small Cellular Search Pattern (SCSP). The two search patterns are shown in Fig.1(a) and Fig.1(b), respectively. The LCSP assumes that the best matching block can be located at any direction pointed away from the LCSP center. Due to its symmetry property, it can be shown that the number of search blocks via LCSP is less than other patterns.

3. The Cellular Search Algorithm

Details of the proposed CS algorithm are given below:

Step 1. Find the block with min(*BDM*) by LCSP. If it is one of the six corners of LCSP, goto Step 2; else goto Step 3.

Step 2. Move LCSP center to the block found in Step 1, and do Step 1 again. Note that at this step, we have three more blocks to be processed. Fig. 2(a)-(c) show three locations that the best matching block is likely located, namely upper right/left, lower right/left, and upper/lower corner of LCSP center.

Step 3 Scan the 8 neighbors of the center block by SCSP (Fig. 2(d)). Identify the block with minimum *BDM* as the matching block B_m. The location of B_m provides the desired motion vector in this moving sequence.

In the following, we explain why LCSP and SCSP are ideal for the CS algorithm. Starting at Fig. 1(a), LCSP initially searches 7 blocks within a 5×5 window. For the same coverage in FS, it requires 25 search blocks, which is 3.5 times that of the CS algorithm. In addition, since

LCSP is symmetric, every subsequent movement would need to search only three more blocks (Fig. 2(a)-(c)). In comparison, in DS the number of required search blocks after the first movement depends on where the block with minimum BDM is located. Finally, one should note that, for real time encoding, speed is the top priority, and visual quality comes the second. For high speed encoding, some trade-off is inevitable. Thus, every movement of LCSP runs across two blocks, compared to the one-block movement via the LDSP in DS.

Therefore, we expect to see that the CS algorithm performs better in terms of the required number of blocks. With the fine-tuning process, CS should achieve similar accuracy performance as the DS algorithm. The following quantitative experiments have been conducted to verify this viewpoint.

4. Experimental Results

Using various input sequences as test inputs, results of Table 1 shows that CS requires less than one tenth search blocks of FS method. In addition, a real world video sequence *'Dolby Digital Trailer'* was used as the test stream. Comparison results (frame by frame) are shown in Figure 3.

5. Conclusion

Two novel search patterns are proposed for motion estimation in video coding. Results have shown that the search patterns and the associated CS algorithm outperform other search algorithms.

References

[1] L. M. Po, and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 6, no. 3, 1996.

[2] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol..4, 438-442, 1994.

[3] S. Zhu, and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, Vol. 9, No. 2, 287-290, 2000.

[4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Tele. Conf.*, New Orleans, 1981.

Table 1 Average number of search blocks for the first 100 frames of different video sequences by using FS, DS, and CS algorithms.

	FS	CS	DS
Tennis	204.28	21.74	43.06
Susie	214.52	17.80	26.57
Flower	204.28	15.72	17.95
Dolby Digital	213.48	21.41	43.12

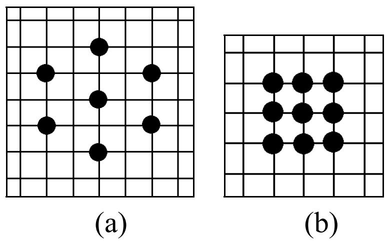


Figure 1. (a) LCSP Pattern (b) SCSP Pattern

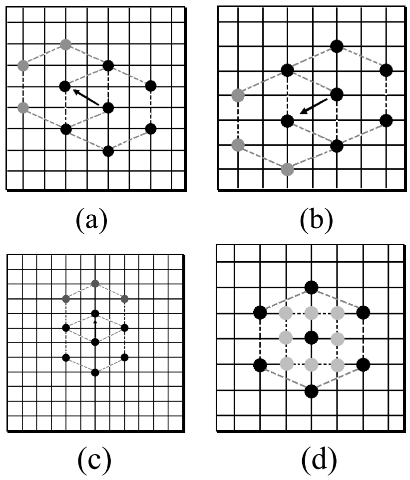
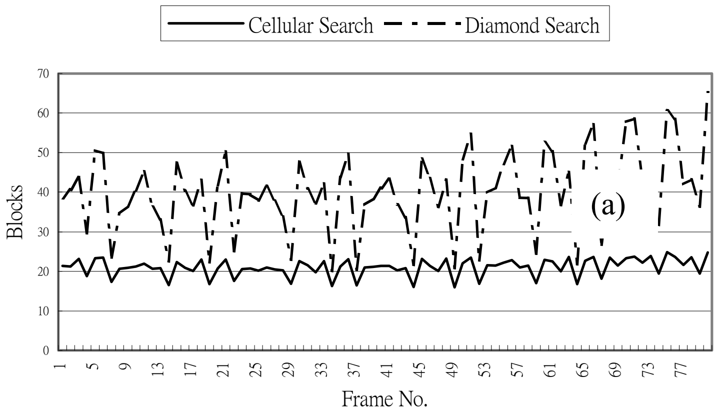


Figure 2. Illustration of moving LCSP for matching blocks located at (a) upper left/right (b) lower left/right (c) upper (d) center of LCSP.



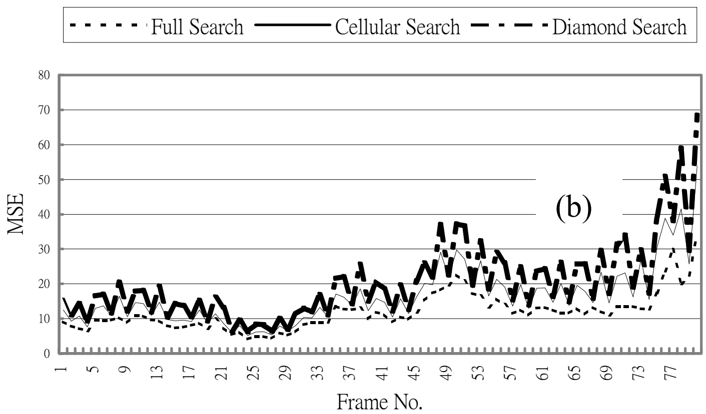


Figure 3. Comparison between CS, FS, and DS algorithms, in terms of (a) search blocks numbers and (b)MSE. Frame distance = 1, input sequence: Dolby Digital Trailer.