

# An Improved Three-Step Search Block-Matching Algorithm for Low Bit-Rate Video Coding Applications

Donglai Xu, Chris Bailey and Reza Sotudeh

Computer Architecture Research Unit, School of Science and Technology  
University of Teesside, Middlesbrough, Cleveland, TS1 3BA, United Kingdom

Phone: +44 1642 342496, Fax: +44 1642 232401, e-mail: d.xu@tees.ac.uk

## ABSTRACT

*An improved three-step search (ITSS) block-matching algorithm for motion estimation is described in this paper, specifically aiming towards low bit-rate video-coding applications. The method is based on the real-world image sequence's characteristic of centre-biased motion vector distribution, and uses centre-biased checking point patterns and a relatively small number of search locations to perform fast block matching. Computational complexity is reduced by employing an  $11 \times 11$  search window rather than the traditional  $15 \times 15$  window. Simulation results are presented which show that the ITSS algorithm provides better performance at faster speed than the well-known three-step search (TSS) algorithm and the recently developed new three-step search (NTSS) algorithm when used for low bit-rate video coding, such as video telephone and video conferencing.*

## 1. INTRODUCTION

Motion estimation plays an important role in motion-compensated video compression, because of its ability to exploit high temporal correlation between successive frames of an image sequence. Although many types of motion estimation algorithm have been developed, the simplicity of the block-matching technique has made it a natural choice for most video compression standards, including MPEG [1], [2], H.261 [3], and H.263 [4]. The approach adopted in block-matching algorithms is first to divide each frame into blocks, typically  $16 \times 16$  pixels. A motion vector is then calculated for each block in the current frame by searching for the best matching block within a limited search area in the reference/previous frame. Compression is achieved by using this best-matched block, indicated by the motion vector, as the predictor for the current block.

Of the block-matching techniques reported in the literature, the full search (FS) method provides the optimal solution by exhaustively evaluating all the possible candidate blocks within the search range in the reference frame. However, massive computation is required in the implementation of FS. In order to speed up the process by reducing the number of search locations, many fast algorithms have been developed, such as the existing three-step search (TSS) algorithm [5] and the recently proposed new three-step search (NTSS) algorithm [6]. Recent studies show that the motion-vector distribution of a real-world image sequence, within the search window, is highly centre-biased. Based on this fact, we propose an improved

version of the well-known TSS method, the improved three-step search (ITSS) algorithm, specifically aiming towards low bit-rate video coding applications. The ITSS has much better performance and faster speed than the original, and compared to the NTSS, its performance is better when used for our intended applications, such as video telephone and video conferencing, and its speed is faster as well, without any direct or hidden costs.

## 2. ITSS ALGORITHM

The experimental results in [6] have shown that the block motion field of real-world image sequences is usually gentle, smooth, and varies slowly. It results in a centre-biased global minimum motion vector distribution instead of a uniform distribution. This is particularly true in low bit-rate video applications, including video telephone and video conferencing, where fast and complex movements are involved rarely. For most video sequences, nearly 80% of blocks can be regarded as stationary or quasi-stationary. Most of the motion vectors are enclosed in the central  $5 \times 5$  area. For such a distribution, we have developed the ITSS algorithm, which uses a centre-biased checking point search pattern, adapted to the centre-biased motion vector distribution. Hence performance might be expected to be far better than that of TSS which uses a uniformly distributed checking point search pattern. Additionally, the ITSS employs a smaller number of search points than TSS in order to speed up block matching. The details of the algorithm are given in the example described below.

Following earlier block-matching techniques, our example takes a block size of  $16 \times 16$  pixels and a maximum search range of  $\pm 7$  pixels in both horizontal and vertical directions. The mean absolute error (MAE) is used as an appropriate estimate of the block distortion measure (BDM). For a given  $(x, y)$ , the MAE between  $block(m, n)$  of the current frame and  $block(m+x, n+y)$  of the previous (reference) frame is defined as:

$$MAE_{m,n}(x,y) = \frac{1}{256} \sum_{i=0}^{15} \sum_{j=0}^{15} |f_k(m+i, n+j) - f_{k-1}(m+x+i, n+y+j)| \quad (1)$$

where  $f_k(i, j)$  and  $f_{k-1}(i, j)$  are the pixel intensities at position  $(i, j)$  of the current frame  $k$  and the previous frame  $k-1$  respectively, and the

$block(m,n)$  is the block with its upper left corner at position  $(m,n)$  of a frame. The first step of the algorithm employs a centre-biased search pattern with nine checking points on a  $5 \times 5$  window in contrast to the  $9 \times 9$  window of TSS. The centre of the search window is then shifted to the point with minimum BDM. The search window size of the next two steps depends on the location of the minimum. If the minimum lies at the centre of the search window, the search will go to the final step (step 3) with a  $3 \times 3$  search window. Otherwise, the search window size is maintained at  $5 \times 5$  for step 2. In the final step, the search window is reduced to  $3 \times 3$  and the search stops at this small search window. The three search steps of ITSS can be summarised as follows:

Step 1: The minimum BDM point is found from a nine-checking-points pattern on a  $5 \times 5$  window located at the centre of the  $15 \times 15$  searching area as shown in Figure 1(a). If the minimum BDM point is found to coincide with the centre of the search window, then go to step 3 else go to step 2.

Step 2: The search window size is maintained at  $5 \times 5$  with a search pattern chosen after considering the two alternatives:

- a) If the previous minimum BDM point is located at one of the corners of the previous search window, then five additional checking points are considered. An example is shown in Figure 1(b) where black circles and grey circles represent additional and previously evaluated pixels respectively.
- b) If the previous minimum BDM point is located at the middle of any horizontal or vertical edge of the previous search window, then three additional checking points are considered. An example is shown in Figure 1(c) where black circles and grey circles represent additional and previously evaluated pixels respectively.

Step 3: The search window is reduced to  $3 \times 3$  around the minimum BDM point found in step 2 as shown in Figure 1(d) and the direction of the overall motion vector is taken to be the minimum BDM point among these nine search points, of which eight are new.

Because there are some overlapping check points on the  $5 \times 5$  search window in the second step of ITSS, the total number of checking points will vary from a minimum of  $(9+8)=17$ , when step 2 can be skipped, to  $(9+5+8)=22$  in the worst case. Pictorial demonstrations of two search paths are provided by the examples shown in Figure 2. In the upper search path, a total of 20 checking points are needed to estimate that the motion vector is  $(1, -5)$ . For the worst case example shown in the lower search path, 22 checking points are required

when the estimated motion vector is  $(-5, 5)$ . For this latter case, the computational complexity of ITSS is three block matches less than TSS and eleven block matches less than NTSS.

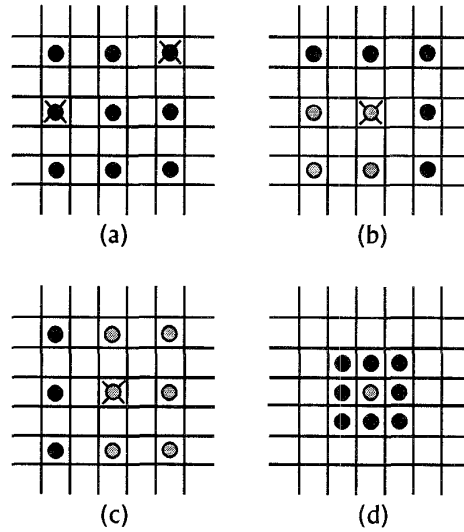


Figure 1: Search patterns of the ITSS. (a) First step centred on centre pixel; (b) Second step centred on a corner pixel; (c) Second step centred on a middle pixel; (d) Third step

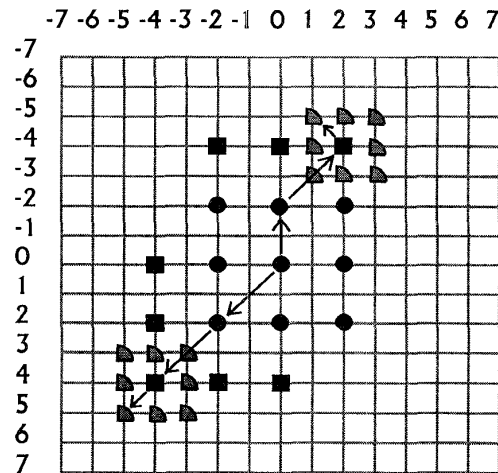


Figure 2: Two examples of ITSS search paths

### 3. SIMULATION RESULTS

In order to test the accuracy and speed of ITSS and to measure its relative performance against other block matching techniques such as FS, TSS and NTSS, simulations were carried out using four well-known Common Intermediate Format (CIF) (30 frames/second

with 352×240 pixels/frame) video sequences, namely “Flower”, “Bike”, “Football”, and “Tennis”. It is noted that these sequences contain different combinations of still, slow, and fast moving objects, camera zoom, and panning. Of these, “Flower” is a slow motion sequence with the lowest proportion of fast motion, while “Tennis” includes many large displacements and involves complex motions, including camera zooming and panning. In each case the block size was fixed at 16×16 and the maximum motion in the search area was  $\pm 7$  pixels in both the horizontal and vertical directions. Moreover, only the first 85 frames of these video sequences were used for test purposes, since they are sufficiently representative of the overall sequences. The performance of ITSS, FS, TSS and NTSS in terms of mean square error (MSE) between the estimated frames and the original frames are compared pictorially in Figures 3(a), 3(b), 3(c) and 3(d).

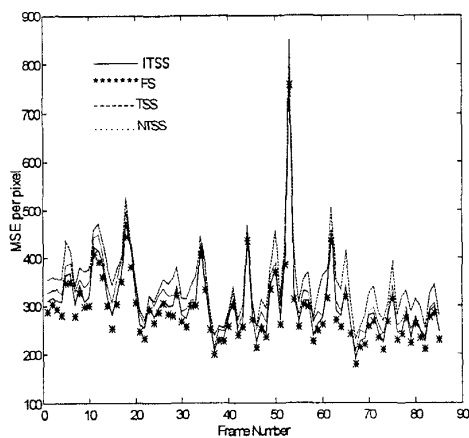


Figure 3(a): MSE comparisons of ITSS, FS, TSS, and NTSS for “flower” sequence

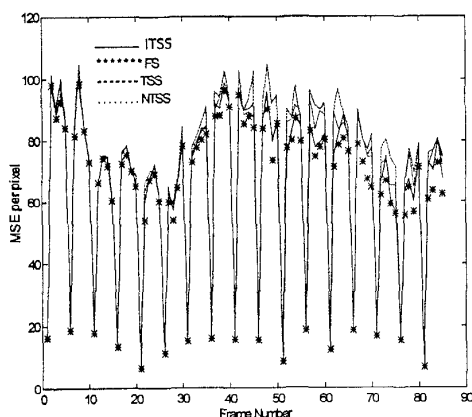


Figure 3(b): MSE comparisons of ITSS, FS, TSS, and NTSS for “bike” sequence

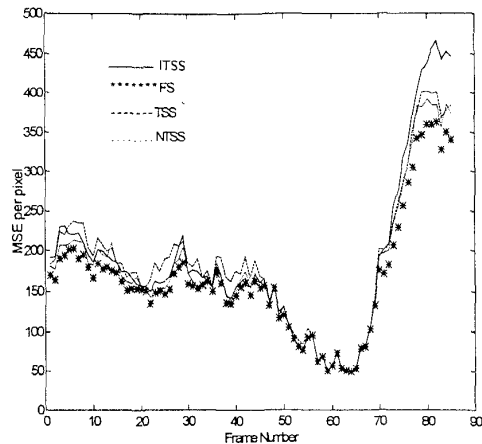


Figure 3(c): MSE comparisons of ITSS, FS, TSS, and NTSS for “football” sequence

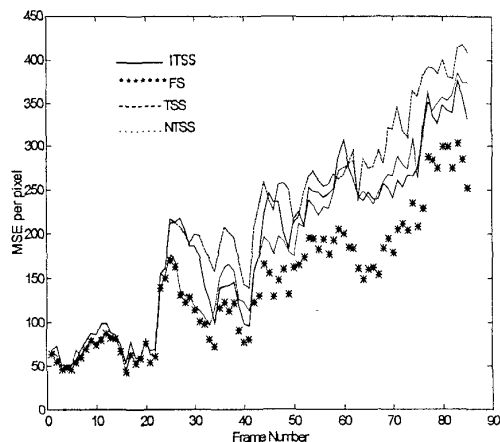


Figure 3(d): MSE comparisons of ITSS, FS, TSS, and NTSS for “tennis” sequence

The figures show that the FS is always the most accurate technique as expected, and that the ITSS provides better performance than the TSS and has similar distortions to the NTSS. For “Flower” and “Bike” sequences where the few fast movement is involved, the ITSS outperforms the NTSS as a whole, although it is slightly less accurate than NTSS for small fractions of the total time of the two sequences. For “Tennis” and “Football” sequences where the very fast and complex movement is involved, the MSE of the ITSS is on the whole greater than that of NTSS, although the difference between them is very small. However, these are not our intended applications. A full comparison of the relative accuracy of all the relevant block-matching techniques as measured by the average MSE over the first 85 frames is presented in Table 1.

Table 1: Average MSE and performance degradation (PD) compared with FS for the first 85 frames of four CIF sequences

Algorithm	Tennis	Football	Bike	Flower
FS	MSE			
	143.93	175.71	56.87	298.27
	PD (%)			
	0	0	0	0
TSS	MSE			
	222.89	219.21	71.19	325.20
	PD (%)			
	54.17	24.76	25.18	9.03
NTSS	MSE			
	203.02	206.69	60.11	309.18
	PD (%)			
	40.05	17.63	5.7	3.66
ITSS	MSE			
	206.77	208.95	59.12	305.88
	PD (%)			
	43.66	18.92	4.0	2.55

From this table, it is clear that the distortions of ITSS with respect to FS are 11% better on the average of all four sequences than those of TSS, and 1.4% better on the average of "Flower" and "Bike" sequences than those of NTSS but 1.95% worse on the average of "Tennis" and "Football" sequences than those of NTSS. The table also shows the tendency that the degradation in distortion performance of ITSS compared with FS would be quite little for the slow motion sequences, such as only 2.55% and 4% for "Flower" and "Bike" sequences, respectively.

Speed of operation is a vitally important parameter for fast block-matching algorithms and a convenient measure of this is provided by the average number of search points required for estimating a motion vector. Table 2 contains relevant data on the average number of search points over the first 85 frames using the four algorithms of interest, and shows that ITSS is the fastest technique for all the four CIF sequences considered. Averaged over these sequences, the speed of ITSS is 15.6% and 42% greater than NTSS and TSS, respectively. Table 2 also demonstrates that the search speed of ITSS for motion vector is higher for the slow motion sequences than for those containing the fast movements or large displacements. In addition, it is noteworthy that the average number of search points using ITSS is usually close to its minimum value of 17. This provides a strong argument in favour of the decision to use a reduced search window area. Overall, in terms of both accuracy and speed, ITSS represents a significant improvement over all previously reported fast block-matching algorithms.

Table 2: Average number of search points per motion vector estimation for the first 85 frames of four CIF sequences

Algorithm	Tennis	Football	Bike	Flower
FS	225	225	225	225
TSS	25	25	25	25
NTSS	22.68	19.78	19.58	19.37
ITSS	17.93	17.59	17.52	17.41

#### 4. CONCLUSIONS

Based on the centre-biased global minimum motion vector distribution characteristic of a real-world image sequence, an improved three-step search algorithm for fast block-based motion estimation, especially for low bit-rate coding such as video telephone and video conferencing applications, has been developed. Experimental results show that this ITSS algorithm performs better than well-known TSS and outscores NTSS for the slow movement sequences in terms of average mean square error. Although it exhibits marginally higher distortion errors than NTSS for frames containing fast motion or large displacements, this is not aimed at our intended applications. Furthermore, it has also been demonstrated that this algorithm is significantly faster, as measured by the average number of search points, than other known three step search methods such as TSS and NTSS.

#### REFERENCES

- [1] ISO/IEC JTC1/SC29/WG11, "ISO/IEC CD 11172: Information Technology", MPEG-1 Committee Draft, 1991.
- [2] ISO/IEC JTC1/SC29/WG11, "ISO/IEC CD 13818: Information Technology", MPEG-2 Committee Draft, 1993.
- [3] International Telecommunication Union, "Video Codec for Audio-visual Services at p×64 kbits", ITU-T Recommendation H.261, 1993.
- [4] International Telecommunication Union, "Video Coding for Low Bit-Rate Communication", ITU-T Draft H.263, 1995.
- [5] J. Jain and A. Jain, "Displacement Measurement and its Application in Interframe Image Coding", IEEE Transactions on Communication, Vol. COM-29, 1981, pp. 1799-1808.
- [6] R. Li, B. Zeng, and M.L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, 1994, pp. 438-442.