An Improved Three-Step Search Block-Matching Algorithm for Low Bit-Rate Video Coding Applications	Phiên bản nâng cao của thuật toán tìm kiếm ba bước theo sự phù hợp khối phục vụ cho các ứng dụng mã hóa video tốc độ bit thấp
An improved three-step search (ITSS) block-matching algorithm for motion estimation	Trong bài báo này, chúng tôi mô tả phiên bản nâng cao của thuật toán tìm kiếm ba bước

is described in this paper, specifically aiming towards low bit-rate video¬coding applications.	theo sự phù hợp khối (ITSS) để ước lượng chuyển động, nhằm hướng tới một ứng dụng cụ thể, đó là mã hóa video tốc độ bit thấp.
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 77X77 search window rather than the traditional 75X15 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.	Phương pháp này dựa trên đặc tính của chuỗi ảnh tuần tự thế giới thực (có phân bố vector chuyển động lệch tâm), và dùng các mô hình điểm kiểm tra lệch tâm và một số lượng tương đối ít vị trí tìm kiếm để so khớp khối nhanh. Độ phức tạp tính toán giảm do sử dụng cửa sổ tìm kiếm 77X77 thay cho cửa sổ 75X15 truyền thống. Các kết quả mô phỏng cho thấy thuật toán ITSS có hiệu suất tốt hơn với tốc độ nhanh hơn các thuật toán tìm kiếm ba bước phổ biến (TSS) và các thuật toán tìm [redacted]
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	[redacted]

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 16x16 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.

[redacted]

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

[redacted]

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 5x5 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.

[redacted]

[redacted]

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 16x16 pixels and a maximum search range of ± 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:

……………………..

where fk(i,j) and fk-i(i,j) are the pixel intensities at position (/,_/) of the current frame k and the previous frame k — 1 respectively, and the block{m,ri) is the block with its upper left comer 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 5x5 window in contrast to the 9x9 window of TSS. The centre of the search window is then shifted to the point with minimum BDM. The search

[redacted]

[redacted]

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 3x3 search window. Otherwise, the search window size is maintained at 5x5 for step 2. In the final step, the search window is reduced to 3x3 and the search stops at this small search window. The three search steps of ITSS can be summarised as follows:	[redacted]
Stepl: The minimum BDM point is found from a nine-checking-points pattern on a 5x5 window located at the centre of the 15x15 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.	[redacted]
Step 2: The search window size is maintained at 5x5 with a search pattern chosen after considering the two alternatives:	[redacted]
a) If the previous minimum BDM point is located at one of the comers 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	[redacted]

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.	[redacted]
Step 3: The search window is reduced to 3x3 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.	[redacted]
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 comer pixel; (c) Second step centred on a middle pixel; (d) Third step	[redacted]
Because there are some overlapping check points on the 5x5 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	[redacted]

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

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 (GIF) (30 frames/second

[redacted]

[redacted]

[redacted]