A FULLY ADAPTIVE PERFORMANCE-SCALABLE DISTANCE-DEPENDENT THRESHOLDING SEARCH ALGORITHM FOR VIDEO CODING

Golam Sorwar, Manzur Murshed and Laurence Dooley

Gippsland School of Computing and Info. Tech., Monash University, Churchill Vic 3842, Australia
{Golam.Sorwar,Manzur.Murshed,Laurence.Dooley}@infotech.monash.edu.au

ABSTRACT

Trading-off computational complexity and quality is an important performance constraint for real time application of motion estimation algorithm. To address this issue, a *distance dependent thresholding search* (DTS) algorithm has been proposed for fast and robust true motion estimation in video coding/indexing applications. DTS encompassed both the *full search* (FS) as well as fast searching modes, with different threshold settings providing various quality-of-service levels. The main drawback of DTS was that the threshold value was manually defined. In this paper, the DTS algorithm has been extended to a *fully adaptive distance dependent thresholding search* (FADTS), a key feature of which is the automatic adaptation of the threshold using the desired target and the content from the actual video sequence, to achieve a guaranteed level of quality or processing complexity. Experimental results confirm the performance of the FADTS algorithm in achieving this objective with minimal additional computational cost.

1. INTRODUCTION

Motion estimation (ME) plays a vital role in video coding standards, such as MPEG-1/2 [1] [2] and H.261/3 [3][4], in exploiting latent temporal redundancy in video sequences. Most ME techniques use *block matching algorithms* (BMA) to compute motion vectors on a block-by-block basis. The most straightforward method, known as full search (FS), provides optimal performance by searching all possible locations within a given search area, but at the expense of very high computation. It is for this reason that FS is not used in real-time systems. Indeed ME is the major bottleneck in real-time video coding applications, hence the need for faster algorithms.

A number of fast block ME algorithms [5]-[10] have been proposed to lower the computation complexity by sacrificing quality. Among these, *three-step search* (TSS) [6] and *new three-step search* (NTSS) algorithms [7]

become more mainly due to their simplicity. However, these motion estimation algorithms are not designed to provide flexible and predictable control of performance in terms of picture quality and computational cost (speed). There is no facility to trade system parameters depending upon a particular application or to preset a user-defined level of picture quality or computational complexity. Such a feature would be very advantageous in facilitating scalable performance management especially in the area of computational complexity management in real time video encoders.

It has been observed that the distortion of an object in a video frame is proportional to its velocity as well as the camera parameters (zoom and pan) and thus, as the length of a motion vector grows so does the block distortion error. Sorwar *et al*. [11]-[14] have addressed this issue by introducing the concept of a distance-dependent thresholding search (DTS) algorithm for fast and robust true motion estimation in object-based video indexing and coding applications. By varying the value of the threshold, the DTS algorithm provides both a FS capability for maximum quality as well as fast searching modes for ME (faster than most traditional algorithms [12]). The main drawbacks associated with DTS are that the threshold value has to be manually selected and cannot be adapted to the content of a particular video sequence.

This paper presents a new *fully automatic adaptive distance-dependent thresholding search* (FADTS) algorithm, which can dynamically adjust the threshold to achieve any level of service required in terms of both quality and processing speed. This means for example, that a higher (lower) error or speed can be achieved by automatically adapting the threshold to a correspondingly level, depending on video content so providing the potential for performance management real time video coding.

The paper is organized as follows. Section 2 briefly describes the basic distance dependent thresholding search (DTS) algorithm, while Section 3 details the new fully adaptive DTS (FADTS) algorithm. Section 4 includes both experimental results and analysis of the performance,

including a computational cost analysis of FADTS for various levels of quality and speed. Section 5 presents the conclusions.

2. DISTANCE-DEPENDENT THRESHOLDING SEARCH (DTS) ALGORITHM INTRODUCTION

A detailed description of DTS algorithm can be found in [11-14], where a technique is presented to estimate the motion vector by introducing the concept of distance-dependent threshold search for variable performance video encoder. This algorithm searches spirally starting from the center of the search window and the search terminates when the block distortion measure (BDM) becomes less than a predefined threshold.

Let the centre of the search region be at pixel $p_{cx,cy}$, which also defines the starting point of the spiral search starting point. In DTS, the spiral search terminates at search square SS_τ [12] when the mean absolute error (MAE), used as the BDM, is:-

$$MAE_{(cx,cy)}(x - cx, y - cy) \leq C \times \tau \qquad (1)$$

where C is the threshold value and τ is the concentric square index. Assuming b-bit gray level intensity, the maximum value of the MAE is (2^b-1), since the pixel intensity is measured using 2^b levels. As SS_d is the outermost search square where d is the maximum displacement, an upper bound for the constant C can be set as:-

$$C < \frac{2^b}{d} \qquad (2)$$

Note, that by setting $C = 0$ in (1), it transforms the DTS algorithm into the exhaustive FS algorithm. It is clear that the search time reduces as C increases and interesting to note that if C is set higher than the upper bound in (2), the search will not explore the entire search area defined by the maximum displacement d.

3. PROPOSED ADAPTIVE THRESHOLD MODEL

The approach adopted for embedding an adaptive threshold into the DTS algorithm is based upon the normalized least-mean-square (NLMS) algorithm [15]-[18]. The threshold is automatically adjusted between frames to achieve either a target level of prediction error (quality) or computation by considering specifically the number of search points per MV.

The block diagram of the proposed model is shown in Fig. 1, and has two modules: (i) motion estimation and (ii) threshold control. In the former, K is the sample vector length, which governs the number of consecutive frames

that use the same threshold value, where sample means a pair of frames between which motion has to be calculated. Thus for $K=1$, a particular threshold value is used to calculate the motion between two consecutive frames, while $K=L$ means the same threshold is used for L-1 consecutive frames (ME always being calculated between two successive frames).
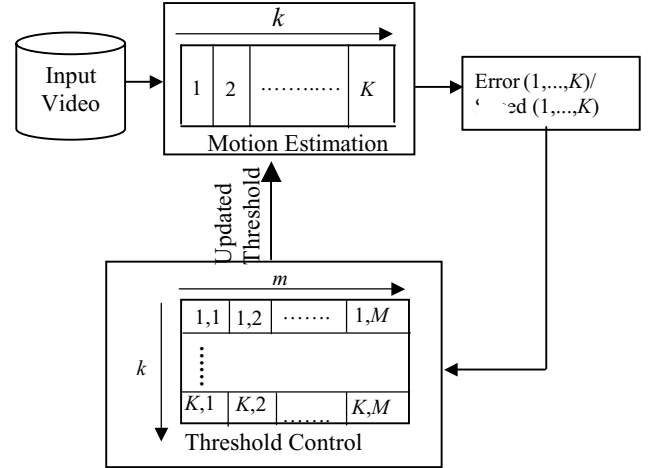

Fig. 1: The proposed DTS adaptive model.

The sample window size is M in the threshold control module, so the total memory requirement for this module is KM. Based on the NLMS method in [16][17] the following is used for threshold adaptation:-

$$C_{j+1} = C_j + \mu e_j(\overline{\mathbf{X}}_j) \frac{1}{\dfrac{\sum_{k=1}^{K}\sum_{m=1}^{M} \mathbf{X}_{k,m,j}^2}{KM}} \qquad (3)$$

where

$$e_j = Desired_j - Actual_j, \quad Actual_j = \frac{\sum_{k=1}^{K}\sum_{m=1}^{M} X_{k,m,j}}{KM}, \ j \text{ is}$$

the number of iterations, μ is the step size, $\overline{\mathbf{X}}_j$ represents the average value of input vector (output of motion estimation module) $\mathbf{X}$, where the total number of elements of $\mathbf{X}$ is K.

The output of the motion estimation module is either prediction quality (*mean square error* (MSE)) per pixel or computational time (the number of *search points* (SP) per MV). This information is used to update the threshold for the following frames. The threshold control module selects whether the threshold for the next iteration is to be either increased or decreased depending on the average error or average number of search points so far calculated (*Actual*) and the target (*Desired*). As C decreases, the number of search points corresponding increase and the update factor

for speed adaptation is therefore negative.

The update term also depends on the value of M. The higher the value of M, the larger the update factor while other parameters remain constant. So the performance of the adaptive algorithms depends on the initial threshold constant selection, the step size and the values of K and M.

4. EXPERIMENTAL RESULTS

The performance of the FADTS algorithm was evaluated using the luminance (Y-component) signal of the following standard test video sequences:- "Football" (320×240 pixels), "Flower garden" (352×240 pixels), "Salesman" (360×288 pixels), "Miss America" (176×144 pixels), "Tennis" (352×240 pixels) and "Foreman" (176×144 pixels). In this paper only the results for the "Football" and "Flower Garden" are presented. The "Football" sequence contains various kinds of motion, including translation, zooming, and panning, while the "Flower Garden" sequence comprises high panning.

In the experiments, all sequences were uniformly quantised to an 8-bit gray level intensity. The block size dimensions were 16×16 and d = ±7, i.e., within each 16×16 block, a maximum of $(2d+1)^2 = 225$ checking points were used. The MSE measure was used to represent the prediction quality for the best motion vector for each block and the value of K and M are selected as 4 and 1 respectively based on the experiments. All results are shown using half-pel motion accuracy.

The performance of FS, TSS and NTSS algorithms are contrasted in Table I, for showing the comparative performance of FADTS algorithm.

Table I: Avg. MSE and SP of FS, TSS and NTSS algorithms for "Football" (344 frames) and "Flower garden" (150frames) video sequences.

BAM	Football		Flower garden	
	MSE	SP	MSE	SP
FS	218.88	160.05	208.91	209.73
TSS	240.79	25.63	242.97	31.20
NTSS	239.15	26.9	213.28	28.98

The performance of the FADTS algorithm was evaluated for both quality and speed adaptation as follows.

4.1. Quality Adaptation

The FADTS algorithm results in terms of quality adaptation are presented in Table II for a number of different target values for the high motion "Football" and "Flower Garden" sequences. This reveals the FADTS algorithm is able to reach any bounded target level of quality, with the implicit assumption that the minimum target error obtained by FS is the lower bound.

If the target is set so high that the resultant threshold

constant will exceed the maximum threshold C_{max}, FADTS algorithm limits the upper bound to C_{max}. However, defining such a high target is unrealistic, because it will produce an extremely poor picture quality output.

Table II: Prediction error adaptation for "Football" a "Flower garden" Video sequences (344 and 150 Frames respectively) with $K= 4$ and $M=1$

Football			Flower garden		
Target quality	Actual	Search Point (SP)	Target quality	Actual	Search Points (SP)
MSE	MSE		MSE	MSE	
220	220.35	34.77	210	212.13	49.82
230	228.95	25.90	215	215.52	26.68
240	240.77	20.89	230	229.34	17.68
250	252.23	18.46	240	237.78	16.18

The corresponding adaptive threshold values for different frames are plotted in Fig. 2. This shows clearly the adaptive nature of the FADTS algorithm as content varies between different frames. It also confirms that FADTS automatically computes a different starting threshold value directly proportional to the target value. Thus initial thresholds are adaptive based on both the content of the video sequence and the desired target.
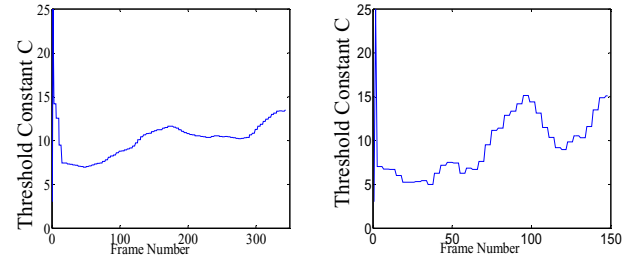

Fig. 2: Threshold constant adaptation for Football (left, 240 MSE) and Flower garden (right, 215 MSE) sequences.
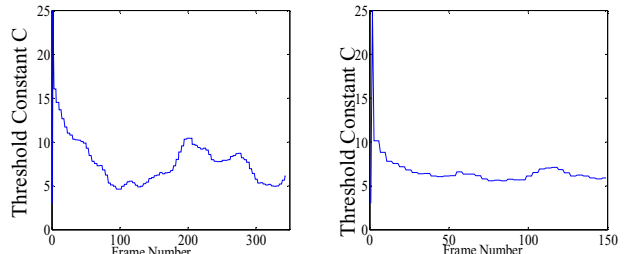

Fig. 3: Threshold constant adaptation for Football (left, 25 SP) and Flower garden (right, 30 SP) sequences.

4.2. Computational complexity adaptation

The computational performance of the FADTS algorithm for a number of different target speeds (average number of search points per MV) is shown in Table III. The table proves that FADTS can reach any average target level of

speed within the bounds (depends on d) by varying the threshold constant. Fig. 3 clearly shows both the adaptive nature of the algorithm as the content in the video sequence varies and also its ability to meet the user-defined target.

Table III: Processing speed adaptation For "Football" and "Flower garden" video sequences (149 Frames) with $K= 4$ and $M=1$

Football			Flower garden		
Target Speed (SP)	Actual	Actual Error MSE	Target Speed (SP)	Actual Speed (SP)	Actual Error MSE
SP	SP		SP	SP	
20	19.89	250.93	20	19.55	219.47
25	24.86	234.93	25	24.53	215.68
30	29.82	227.09	30	29.56	214.28
40	40.97	220.22	40	39.78	212.99

Table I, II and III also prove that FADTS does not only reach any user defined target, but also shows better error performance when complexity is comparable to TSS or NTSS. Another noteworthy point is that FADTS can achieve the same MSE performance as FS but with reduced complexity (SP) by a factor of 4 (approx.).

The total number of operations for (3) is only $(3KM + 4)\dfrac{f}{K}$ per second where f is the number of frames per second. Since in the experiments, $M=1$ and $K=4$, this means a total of only 120 additional operations per second. This is negligible compare to the complexity involved in MAE distortion calculation for motion estimation, where one MAE calculation requires 511 additions, 256 absolute operations, and one comparison for a 16×16 block.

In summary therefore, the FADTS algorithm consumes minimal additional computational overhead, while providing significant performance benefits including user-definability of key parameters.

5. CONCLUSIONS

This paper has presented a fully adaptive distance dependent thresholding search (FADTS) algorithm for real-time block-based motion estimation in video coding. The performance of FADTS has been examined and proven that it affords a unique feature in being able to trade-off freely between the two key system parameters, namely prediction quality and search speed, for the entire range of threshold values. A key feature of this novel algorithm is its ability to progressively adjust the required threshold value based on the actual video content to achieve any user-specific level-of-service, in terms either

prediction quality or processing speed. FADTS can therefore be used as an optimum algorithm for high quality prediction as well as a very fast algorithm. The algorithm proposed in this paper could also form part of a video encoder that can optimize performance in scenarios where computational resources are restricted. Further work is required to integrate this algorithm with other functions of the encoder such as the DCT and quantization scale to control the rate, complexity and distortion performance.

6. REFERENCES

[1] JTC1/SC29/WG11, I.I., Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit. 1993.

[2] JTC1/SC29/WG11, I.I., Generic Coding of moving pictures and associated audio. 1993, ISO/IEC.

[3] H.261, I.-T.R., Video codec for audiovisual services at p´64 kbit/s. 1993.

[4] H.263, D.I.-T.R., Video coding for low bitrate communication. 1996.

[5] J.R. Jain and A.K. Jain, "Displacement measurement and its application in inter frame image coding," IEEE Trans. Commun., vol. COM-29, pp. 1799-1808, Dec. 1984.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima and T. Ishiguro, "Motion-compensated inter frame coding for videoconferencing," in Proc. NTC81, Nov. 1981, pp. G5. 3.1-G5.3.

[7] 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, pp .438-442, Aug. 1994.

[8] H. Nisar and T.-S. Choi, "An advanced center biased search algorithm for motion estimation," in Proc. ICIP, 2000, vol.1, pp.832-5.

[9] L. M. Po and W. C. Ma, "Novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317, June, 1996.

[10] M. Ghanbari, "The cross-search algorithm for motion estimation (image coding)," IEEE Trans. Commun., vol. 38, no. 7, pp. 950-953, July, 1990.

[11] G. Sorwar, M. Murshed, and L. Dooley, "Modified Full-Search Block-Based Motion Estimation Algorithm With Distance Dependent Thresholds," in proc. ICASSP. 2002.

[12] G. Sorwar, M. Murshed, and L. Dooley, "Distance Dependent Thresholding Search For Fast Motion Estimation in Video Coding," accepted in IEEE Asia-Pacific Conf. on Circuits and System (APCCAS'02). 2002. Kartika Plaza Hotel, Bali, Indonesia.

[13] G. Sorwar, M. Murshed, and L. Dooley, "Fast block-based true motion estimation using distance dependent thresholds (DTS)," accepted in the 6th International Conference on Signal Processing (ICSP'02). 2002. Beijing, China.

[14] G. Sorwar, M. Murshed, and L. Dooley, "Fast Motion Estimation in Video Coding Using Distance Dependent Thresholding," submitted in IEEE Trans. on circuit and systems for video tech., 2002.

[15] S. Haykin, Adaptive Filter Theory, Englewood Cliffs, NJ: Prectice-Hall, 1991:

[16] B. Widrow and J. M. E. Hoff, "Adaptive Switching Circuits," in IRE WESCON Conv. Rec. 1960.

[17] B. Widrow and S.D. Stearns, Adaptive Signal Processing, NJ: Prentice Hall,Englewood Cliffs, 1985.

[18] K. Meghriche, S. Femmam, B. Derras and M. Haddadi, "A non linear adaptive filter for digital data communication," Proc. of the Sixth Int. Symp. on Signal Proc. and its Appl., vol.1, pp.304-6, Piscataway, NJ, USA, 2001.