

A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation

Anastasios Hamosfakidis

*Department of Computer Science, Queen Mary, University of London, Mile End Road, E1-4NS, UK
Email: anastasi@dcs.qmw.ac.uk*

Yakup Paker

*Department of Computer Science, Queen Mary, University of London, Mile End Road, E1-4NS, UK
Email: paker@dcs.qmw.ac.uk*

Received 26 July 2001 and in revised form 5 February 2002

Based on real-world image sequence characteristics of center-biased motion vector distribution, a Hexagonal (HS) algorithm with center-biased checking point pattern for fast block motion estimation is proposed. The HS is compared with full search (FS), four-step search (4SS), new three-step search (NTSS), and recently proposed diamond search (DS) methods. Experimental results show that the proposed technique provides competitive performance with reduced computational complexity.

Keywords and phrases: fast motion estimation, hexagonal search pattern, center-biased motion vector estimation.

1. INTRODUCTION

Motion compensated video coding, which predicts current frame from I and P frames, has been widely used to exploit the temporal redundancy between consecutive frames. Motion estimation plays an important role in such an interframe predictive video coding system. Among different types of motion estimation algorithms, the block matching technique has been adopted in many compression standards, such as H.261 [1], MPEG-1 [2], MPEG-2 [3], and MPEG-4 [4]. In block matching, video data, frames, or video object planes (VOPs) are divided into blocks and one motion vector (MV) is associated with each block—A video object (VO) in MPEG-4 is an arbitrarily shape video segment that has a semantic meaning. A 2D snapshot of a VO at a particular time instant is called video object plane. For each block of the current frame/VOP, an MV is derived which points to the best matching block of the previous (reference) frame/VOP. Then the best match block is used as the predictor for the current block.

The full search (FS) block matching algorithm is the simplest but the most compute-intensive solution as it provides the optimal solution by matching all the possible displaced blocks within a given search range in the reference frame/VOP. In order to speed up MV derivation, many fast block matching motion estimation (BMME) algorithms have been proposed such as three-step search (3SS) [5], one-at-a-time search (OTS) [6], four-step search (4SS) [7], new three-step search (NTSS) [8], and diamond search (DS) [9].

The objective of these fast BMME algorithms is to find the MV that minimizes the image error by reducing the number of checking points within the search window. Of these, 3SS and OTS algorithms are known to have the tendency to be trapped into a local minimum, thereby degrading performance. Based on the analysis of the above-mentioned fast BMMEs and a study of MV distribution of real-world test video sequences, a new hexagonal search (HS) algorithm is proposed in this paper. In Section 2, we introduce the design motivation of the proposed algorithm. The algorithm description and simulation results are presented in Sections 3 and 4, respectively. Finally, we give conclusions in Section 5.

2. DESIGN MOTIVATION

Experimental results [7, 9] have shown that the block motion field of real-world video sequences is usually smooth, and varies slowly. That leads to a center-biased global minimum MV distribution instead of a uniform distribution. This can be observed from the MV distribution based on the FS algorithm for the first 100 frames of two test video sequences: the well-known *News*, and the *Rallycross*. The object-based MPEG-4 video standard segments the frames of video sequences into video objects (VOs). We have studied the MV distribution of *News 0*, *News 1* VOs, and *Rallycross*. For the *News 0* (background object) sequence, nearly all the blocks (97.55%) can be considered stationary (Figure 1a). For the *News 1* (dancer) sequence of faster motion and camera

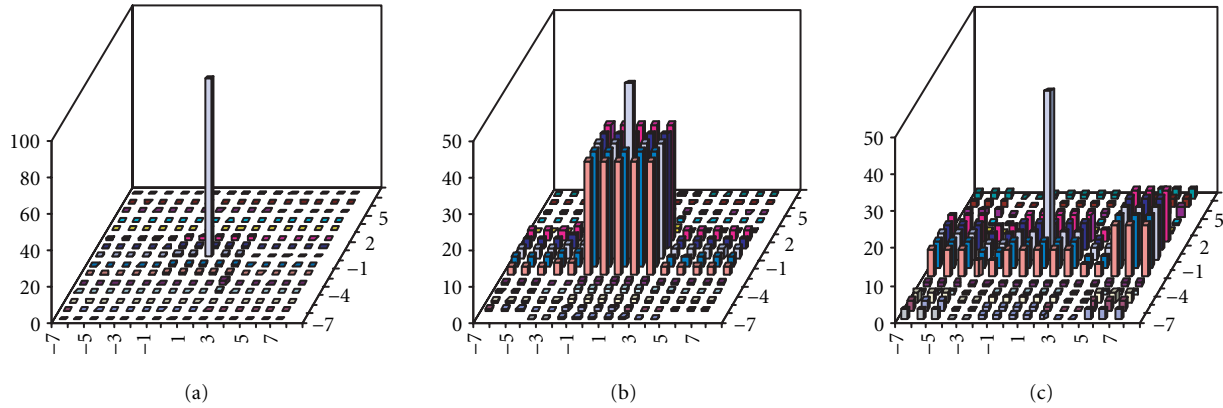


FIGURE 1: Motion vector distribution for (a) *News 0*, (b) *News 1*, and (c) *Rallycross* sequences.

zooming, the MV distribution is still highly center-biased: 48.31% found at the center of the search area, and 80% of them are enclosed in a central 5×5 area (Figure 1b). For the nonsegmented fast motion sequence *Rallycross*, 52.76% of its MVs are enclosed in a central 5×5 area and 61.16% are located in a central search 9×9 area (Figure 1c).

Since the distribution of the global minimum point in real-world video is centered at zero, fast BMME algorithms have been developed using center-biased checking point patterns. Analytically, the NTSS employs a center-biased checking point pattern combined with a halfway-stop technique, and achieves better performance than TSS. Using a moderate search pattern with fixed size of 5×5 , the 4SS obtains a performance that is similar to NTSS. DS employs two diamond search patterns of size 9×9 and 3×3 , respectively, which do not cover edge points of the search area. It becomes clear that the shape and the size of the above mentioned search patterns jointly determine not only the image quality (error performance) but also the computational complexity of fast BMMEs.

Based on the observation that global minimum distribution is centralized in real-world sequences, the search points positions included to a zero-centered search 5×5 area are the most appropriate ones to be chosen to compose the search pattern. This choice is quite crucial in terms of algorithm's complexity and performance. This is the motivation behind the new hexagonal search (HS) algorithm proposed here.

3. THE HEXAGONAL SEARCH ALGORITHM (HS)

Since motion vectors are not evenly distributed in the search area (in fact most of them are located inside a centre-biased window of size 9×9), the HS patterns are designed to take into account the following:

- (i) *reduced computational complexity*: the point where the minimum block distortion (MBD) occurs should be tracked using a small number of checking points, which cover a significant portion of the center-biased search window,

- (ii) *search patterns shape*: when the MBD point for one step is located, the search pattern has to be shaped in such a way that allows a refined search which covers all searching points around the MBD point in order to derive the MBD point for the best matching block.

As shown in Figure 2, the HS algorithm utilises a centre-biased search pattern of seven checking points, out of which six points surround the centre one to compose a hexagon (Step 1). The hexagon points are checked and the centre of the hexagonal search window is then shifted to the point with MBD. The search pattern and its size, for the next two steps of the HS, depend on the location of the MBD points. If the MBD point is found at the center of the hexagonal pattern, the search proceeds to the final step (Step 3), with a smaller search pattern for a refinement search. Otherwise, the hexagonal search pattern is applied repeatedly until the MBD point is found at the center of the hexagon (Step 2). When the final step (Step 3) is reached, the search pattern is changed from hexagonal to a star (Figure 2b) with a variable number of search points, the best case is 4 and the worst is 6. For edge points of the search area the hexagonal search pattern (Step 2) is modified (Figure 3). The HS algorithm is summarised as follows.

Step 1. The initial hexagonal pattern is centered at the origin of the search window and the seven checking points (\bullet) of the hexagon are tested (Figure 2a). If the MBD point is found at the center position then go to Step 3, otherwise go to Step 2.

Step 2. The MBD point found in the previous search step is repositioned as the center point to form a new hexagon. If the new MBD point obtained is located at the centre position, go to Step 3; otherwise, recursively repeat Step 2. The hexagonal pattern is modified on the borders of the search area in order to cover the edge points. Figure 3a presents all the possible shapes of the hexagonal pattern when it reaches the left/right or the up/down borders of the search area. More precisely, there are two different scenarios when the pattern reaches the top or down borders. The first scenario is the centre point of the pattern, when it is shifted towards the up

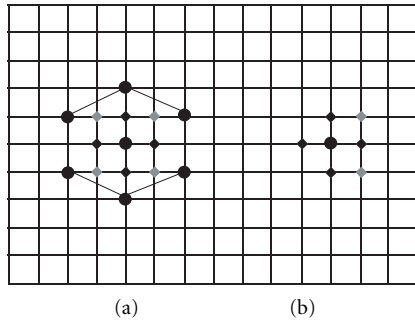


FIGURE 2: (a) Hexagon and initial star patterns, neighbouring points of the initial star are shown by grey colour. (b) Expanded star pattern for no edge points.

or down borders, to be on the border. In this case the new hexagonal pattern employs four checking points. The other scenario is when the checking points of the shifted hexagonal pattern are out of the borders of the search area, in this case the shifted pattern has six checking points. Similarly, there are two cases when the shifted hexagonal pattern reaches the right or left borders of the search area. One case is when the centre of the shifted pattern is on the border of the search area, and the other is where checking points of the shifted pattern lie outside the borders. In both cases the modified pattern has five checking points.

Step 3. Switch the search pattern from hexagon to star (◆) (Figure 2a).

There are two different star patterns that are employed for different locations of the MBD point in the search area:

- The MBD point is not an edge point. The initial star pattern, Figure 2a is centered to the MBD point of the hexagonal pattern and its four checking points are tested. If the new MBD point calculated for the star pattern is located at the centre, then this point is the final solution for the motion vector and the search stops. Otherwise, if the new MBD point is one of the other points of the initial star, then its neighbouring points, excluding the central star point, are checked (Figure 2b). The new derived MBD point is the final solution of the MV, since it generates the smallest MBD in the pattern.
- The MBD point is an edge point. The initial star pattern, adjusted to three checking points, is centered on the MBD of the hexagonal pattern, and its three checking points are tested (Figure 3b). If the MBD point is at the centre of the modified star the search stops, otherwise the neighbouring points of the MBD, excluding the central point, are examined. The new derived MBD point is the final solution of the motion vector that points to the best matching block.

Note that the checking points of the hexagon search pattern are partially overlapping when Step 2 is repeated. Only three checking points need to be calculated in the new pattern. In addition, at Step 3 when the search pattern changes

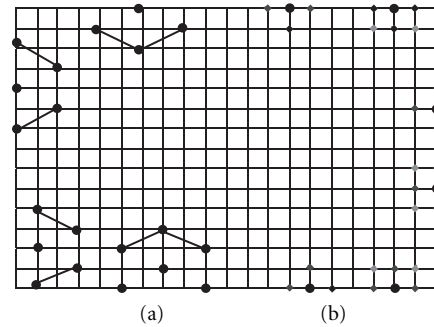


FIGURE 3: (a) All possible shapes of the hexagonal pattern when it reaches the left/right or the up/down limits of the search area. (b) Possible star shapes (initial and expanded) for a down/up or right edge MBD points.

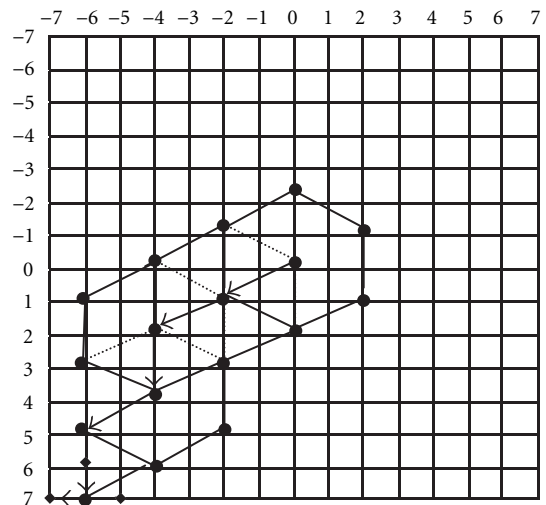


FIGURE 4: HS search path for MV $(-7, 7)$.

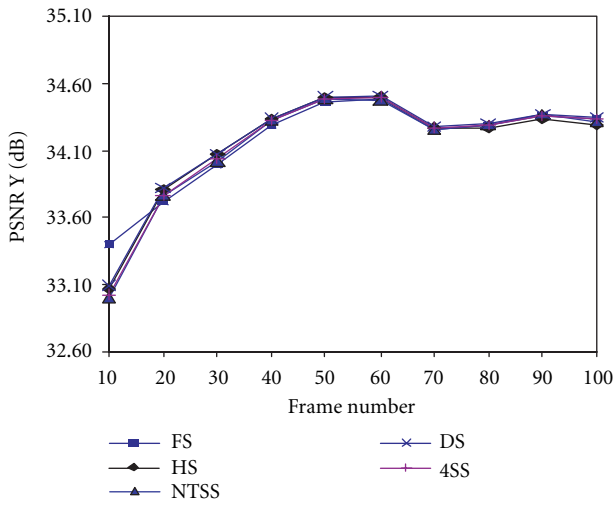
from hexagon to star, three, four, or six points of the star need to be calculated, depending on the star's MBD point location. Figure 4 presents an example of how the HS derives an MV from the borders of the search area, MV $(-7, 7)$.

4. SIMULATIONS

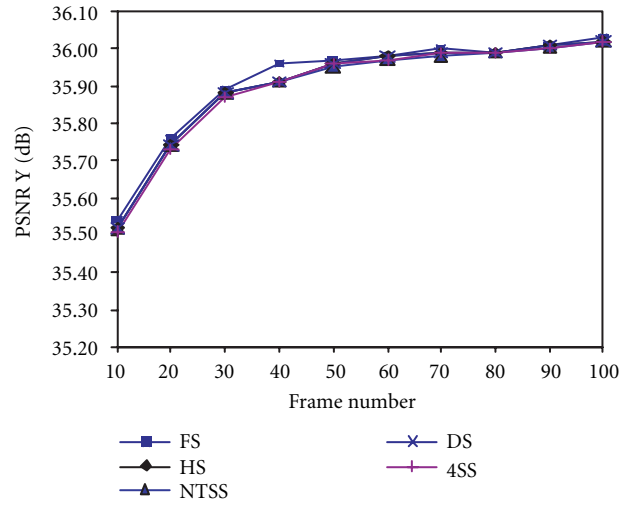
In our simulations, the MBD is defined to be the sum of absolute difference (SAD). For a given displacement (x, y) , SAD is defined as

$$\text{SAD}(x, y) = \sum_{m=x}^{x+N-1} \sum_{n=y}^{y+N-1} |I_k(m, n) - I_{k-1}(m+x, n+y)| \quad (1) \\ \times (\alpha_k \neq 0),$$

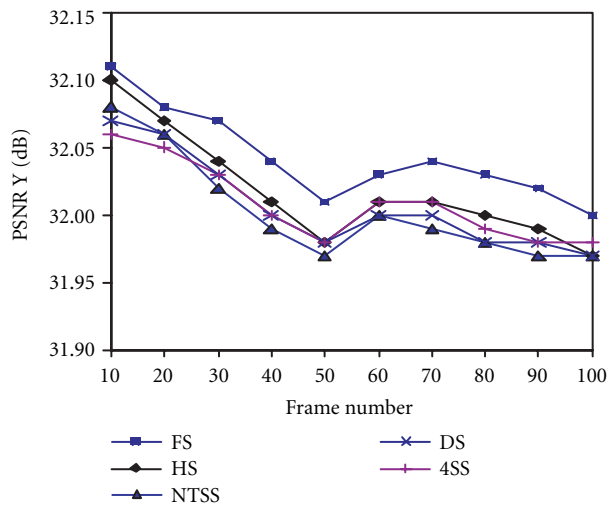
(see [10, 11]), $I_k(x, y)$ is the pixel intensity (luminance or Y component) at location (x, y) in the k th frame/VOP, and α_k refers to the current VOP at time instance k and contains the information of which pixels are inside the object ($\alpha_k > 0$) and which are not ($\alpha_k = 0$). The block size is



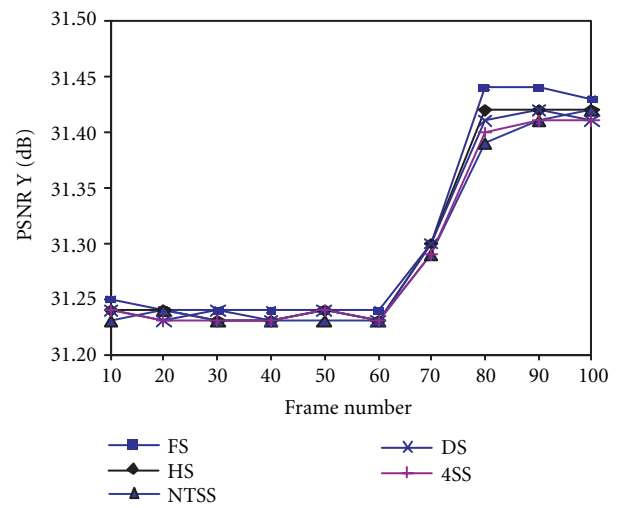
(a)



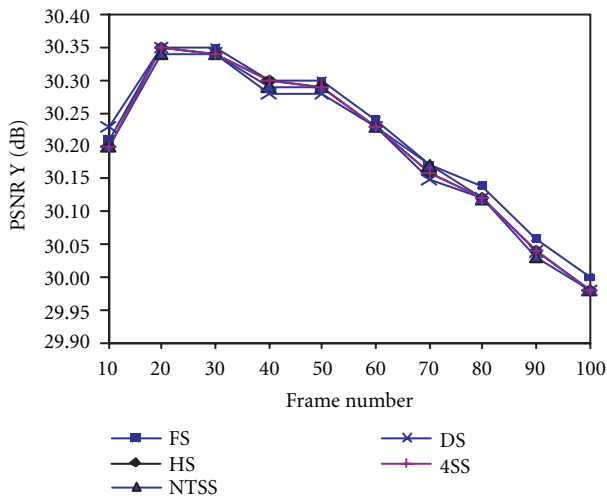
(b)



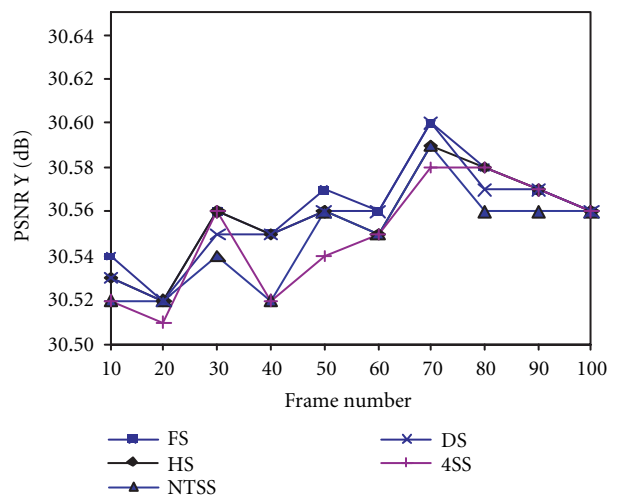
(c)



(d)



(e)



(f)

FIGURE 5: PSNR comparisons of HS, DS, NTSS, 4SS, and FS for (a) Rallycross, (b) News 0, (c) News 1, (d) Coastguard 0, (e) Coastguard 1, and (f) Football.

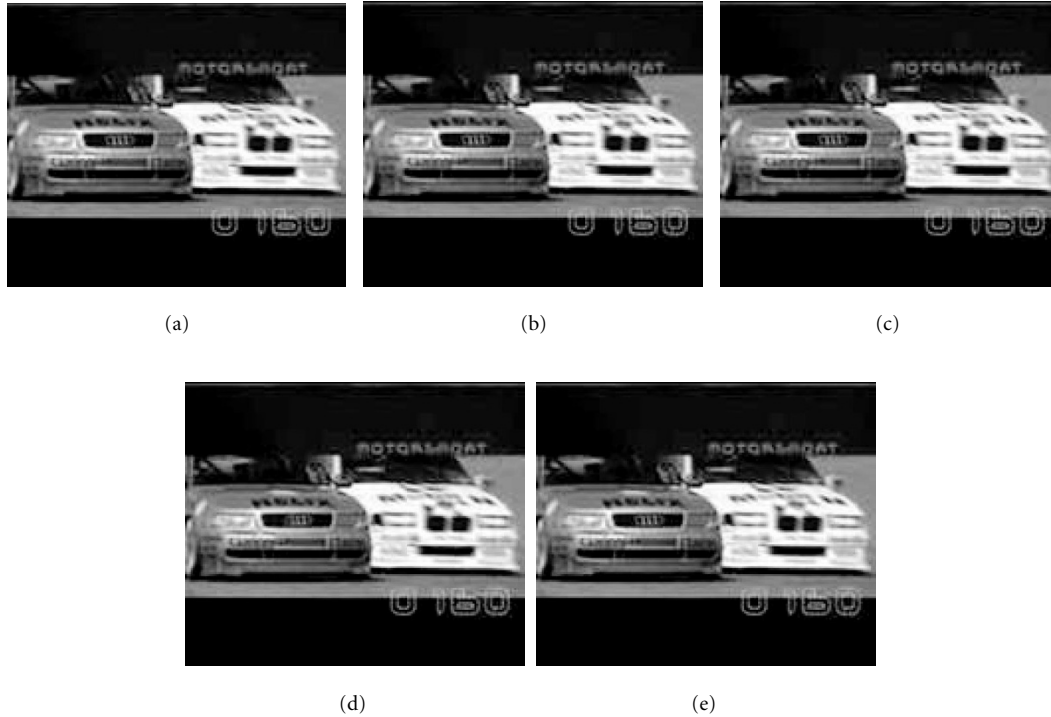


FIGURE 6: The 55th estimated frames for the Rallycross sequence using different searching algorithms. Estimated frames using (a) FS, (b) DS, (c) 4SS, (d) NTSS, and (e) HS.

TABLE 1: Average search points (SP) per motion vector (MV) estimation and PSNR for the first 100 frames.

Algorithms	News 0			News 1			Rallycross		
	Av. SP/MV	Complexity	Average PSNR	Av. SP/MV	Complexity	Average PSNR	Av. SP/MV	Complexity	Average PSNR
FS	225.00	100%	35.91	225.00	100%	32.04	225.00	100%	34.15
4SS	17.03	7.57%	35.89	18.37	8.16%	32.01	20.67	9.18%	34.13
NTSS	17.05	7.58%	35.89	18.96	8.42%	32.00	21.01	9.34%	34.12
DS	13.05	6.00%	35.9	15.25	6.77%	32.00	17.99	7.99%	34.14
HS	11.06	4.91%	35.9	12.67	5.63%	32.02	14.44	6.41%	34.14

TABLE 2: Average search points (SP) per motion vector (MV) estimation and PSNR for the first 100 frames.

Algorithms	Coastguard 0			Coastguard 1			Football		
	Av. SP/MV	Complexity	Average PSNR	Av. SP/MV	Complexity	Average PSNR	Av. SP/MV	Complexity	Average PSNR
FS	225.00	100%	31.30	225.00	100%	30.21	225.00	100%	30.56
4SS	18.16	8.07%	31.29	21.94	9.75%	30.19	20.67	10.18%	30.54
NTSS	18.26	8.11%	31.28	22.31	9.91%	30.19	21.01	10.34%	30.54
DS	15.62	6.94%	31.29	20.97	9.32%	30.20	17.99	9.09%	30.55
HS	11.22	4.98%	31.29	17.21	7.64%	30.20	14.44	7.91%	30.55

fixed at $N \times N$ with $N = 16$, and the maximum motion in row and column is assumed to be ± 7 . The first 100 frames of *News 0*, *News 1*, *Coastguard 0*, *Coastguard 1*, *Football*, and *Rallycross* video sequence are used. Analytically, we used

two fixed size VOs of the *News Quarter Common Intermediate Format (QCIF)* video sequence, *News 0* (slow motion), and *News 1* (faster motion and zooming), two variable size video sequences *Coastguard 0* and *Coastguard 1* that have

resolutions between Common Intermediate Format (CIF) and QCIF (variable size medium motion video objects), and the Football and Rallycross which are QCIF medium-fast motion video sequences. The frame rate for all the sequences is 25 frames/second. We use the peak signal-to-noise ratio (PSNR) as the measure of performance. The PSNR is an image quality metric where larger values of it translated to better quality,

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right). \quad (2)$$

(MSE is the mean squared error.) The required average number of search points for each block is used as the measure of computational complexity. Each video sequence is processed by five algorithms: FS, DS, 4SS, NTSS, and the proposed HS. The degree of computational complexity of each algorithm with respect to full search algorithm is calculated. The simulation results are shown in Tables 1, 2, and Figure 5.

The simulation shows that HS performs better than DS, 4SS, and NTSS for all the video sequence with similar performance to FS in terms of PSNR, Figures 5, 6 and Tables 1, 2.

5. CONCLUSIONS

In this paper, the HS algorithm is proposed to perform block motion estimation in video coding. Based on the observations that global minimum distribution is centralized in real-world video sequences and that the shape and size of the search patterns determine not only the performance but also the computational complexity of fast BMMs, the HS employs a center-biased hexagon search pattern. Experimental results show that the proposed HS outperforms DS, 4SS, and NTSS search algorithms, having always better average computational complexity (slow, medium, fast motion) with similar performance to FS in terms of PSNR.

REFERENCES

- [1] International Telecommunication Union, "Video codec for audiovisual services at px64 kbits," ITU-T Report/Recommendation H.261, March 1993.
- [2] ISO/IEC JTC1/SC29/WG11, "ISO IEC CD 11172: Information Technology," Committee draft of the MPEG-1 standardization body, December 1991.
- [3] ISO/IEC JTC1/SC29/WG11, "ISO IEC CD 13818: Information Technology," MPEG-2 Committee Draft, December 1993.
- [4] "Special Issue on MPEG-4," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, February 1997.
- [5] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. National Telecommunications Conference*, pp. C9.6.1–9.6.5, New Orleans, La, USA, November 1981.
- [6] R. Srinivasan and K. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Communications*, vol. 33, no. 8, pp. 888–896, 1985.
- [7] L. Po and W. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313–317, 1996.

- [8] R. Li, B. Zeng, and L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, 1994.
- [9] S. Zhu and K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *International Conference on Information, Communications, and Signal Processing*, pp. 292–296, Singapore, September 1997.
- [10] T. Ebrahimi, "MPEG-4 video verification model version 8.0," ISO/IEC JTC/SC29/WG11 N1796, Stockholm MPEG-4 meeting, Sweden, July 1997.
- [11] MPEG-4 simulation software, "ISO/IEC N2205," Tokyo MPEG-4 meeting, Japan, March 1998.

Anastasios Hamosfakidis received the diploma of computer engineering from the Technical University of Patras, Greece, in 1995 and his Ph.D. degree in computer science from the Queen Mary, University of London in 2001. His research interests include image and video compression, low bit-rate video encoding, motion estimation, and parallel architectures.



Yakup Paker is an Emeritus Professor at Queen Mary, University of London. His research interests have included parallel computing, real-time architectures for video processing, virtual studios, and digital broadcasting convergence. He has published extensively and is the author/editor of a number of books. He holds a Ph.D. and an MS degrees from Columbia University, New York, USA, and a degree in electrical engineering from Istanbul Technical University, Turkey.

