BITCEM:An Adaptive Fast Block Motion Estimation via Binary Transform Center of Mass Object Tracking

Kuo-Liang Yeh*
Wen-Hsien Fang**
Mon-Chau Shie***
Fei-pei Lai****
Senior Member, IEEE

Abstract

In this paper, a fast algorithm for block motion estimation is presented. BInary Transform and CEnter of Mass (BITCEM) object tracking techniques are adopted in the proposed algorithm. In this algorithm, the block difference is used to find the maximum pixel value regarded as the reference to define the moving object area in the block. Via binary transform, we greatly reduce the computation in this step. Sub-sampling method is used here to further achieve the computation reduction. Then, we calculate the center of mass of the moving object to quickly estimate the direction of the moving object. According to this direction, we adaptively arrange a different search point pattern to each block.

* Lecturer, Dept. of Information Management, Kang-Ning Junior College of Medical Care and Management
** Professor, Dept. of Electronic Engineering, National Taiwan University of Science and Technology
*** Assistant professor, Dept. of Electrical Engineering, National Taiwan University of Science and Technology
**** Professor, Dept. of Electrical Engineering & Dept. of Computer Science and Information Engineering,National Taiwan University

Different video motion types are classified by the percentage of (0,0) BITCEM motion vectors. we classify the image sequences into three categories: quasi-still, slow movement and fast movement, based on the movement of the center of mass. Furthermore, the BITCEM motion vector and the temporal-spatial correlations are used to estimate the initial search point based on the video motion type.

The furnished simulation results show that the BITCEM can achieve a speed-up ratio from 14.0 to 21.8 with only 0.7 %-9.6 % increase of the mean square errors (MSE) as compared to the full search algorithm in various scenarios.

Keywords：adaptive, block-based motion estimation, center of mass estimation, object tracking, spatio-tempral correlations.

BITCEM:以二元值轉換質心爲基礎的調適性物件循跡快速區塊移動估測法

葉 國 良*

方 文 賢**

許 孟 超***

賴 飛 羆****

Senior Member, IEEE

摘 要

由於提高視訊壓縮效能的關鍵在於有效消除時間域的冗餘度。因此，移動估測和補償技術在視訊壓縮領域中扮演著重要的角色。其中，區塊比對法由於複雜度低，且容易設計成硬體架構的優點，而受到廣泛地注意。基於壓縮速度和電力消耗的考量，在不嚴重影響影像品質的前提之下，已有許多方法被提出以企圖降低搜尋複雜度。爲了克服大多數區塊比對技術的缺點，本文提出了一個不錯的以二元值轉換質心（BInary Transform CEnter Of Mass, BITCEM）爲基礎的調適性（Adaptive）快速區塊運動估測技巧。

首先，本文修正了傳統的質心定義，加入粗-微調（Coarse-fine）的觀念，利用此質心在相鄰圖框間的運動方向來粗略估測運動區塊的運動方向。接著利

* 康寧醫護暨管理專科學校 資訊管理科講師

** 國立台灣科技大學 電子系教授

*** 國立台灣科技大學 電子系助理教授

****國立台灣大學 電機系/資工系教授

用質心移動量將視訊分類爲近似靜止、慢速運動及快速運動視訊，針對不同種類的視訊及運動方向採用了不同的搜尋點配置，以精確地決定區塊的運動向量。此外，基於相鄰區塊運動向量之時間-空間相關性，本文亦利用線性預測模型（Linear Prediction Model）及質心運動量，估測運動向量的初始值，以加快搜尋速度。

由實驗結果可證明：與傳統方法比較起來，本文提出的方法不但可以大量降低複雜度（視測試視訊的不同，比全域搜尋法快了 14.0~21.8 倍），同時也維持了極佳的影像品質（只比全域搜尋法增加了 0.7%~9.6% 誤差）。

關鍵詞：時間－空間相關性、粗-微調、區塊運動估測、調適性、質心。

I、Introduction

Several fast algorithms, such as the three-step search (TSS), new three step search (N3SS), and four step search (4SS) algorithms have been developed to reduce the computational complexity by reducing the number of checking points. Based on the characteristic of center-biased motion vector distribution, the N3SS and 4SS algorithms for improving the performance of the TSS on the estimation of small motions were proposed in [7][4]. They not only employed the characteristic of center-biased motion vector distribution but also used the halfway-stop technique to speed up the stationary or quasi-stationary blocks matching. Simulation results in [7][4] show that the N3SS and 4SS are much more robust, and produce smaller motion compensation errors as compared with the TSS. On the other hand, there are some researches focused on the one-bit transform techniques for motion estimation. For instance, in [1] the one bit transform (1BT) was used to indicate whether a pixel was an edge pixel or not. The advantage of this representation is that the distortion between the reference block and the search block can be computed using the exclusive-or (XOR) function very efficiently. Its performance, however, is not good enough. In this paper, the proposed BITCEM algorithm is a totally novel method to do the motion estimation. We introduce the center of mass technique and the one-bit (binary) transform concept into our algorithm. In order to further reduce the computational complexity and make the algorithm more robust, three more techniques including sub-sampling, classification of the video motion type, and the initial search point are introduced to the proposed searching scheme[5].

II、Preliminary

The idea of center of mass technique has been used in other video applications [3][6], but the calculation of center of mass requires too much computation so that a re-definition is needed to reduce the computation. Thus, we transform the gray level of image to binary level image to reduce the number of operations. From this bit transform center of mass, we can quickly compute the BITCEM of moving object in the block and roughly estimate their motion direction. Three more techniques are employed to reduce the computational complexity and increase the picture quality. In the following, we will discuss all the techniques used in the proposed searching scheme.

1、Center of Mass via Binary Transform

The extra effort in calculating the center of mass (CEM) of nonmotion area is useless, so we redefine the center of mass to be the CEM of motion area in order to save calculation effort. To define the CEM, we make the following assumptions:

- ▲ The object in block will not distort during moving, which is the assumption of rigid object motion.
- ▲ There is only one moving object in one block and the whole object has the uniform gray level.
- ▲ The object will not move outside this block.

We do the binary transformation to each block to make every pixel into a bi-level value and assume the motion part of block is P. $P(i,j) = 1$ means (i, j) pixel is moving, while $P(i,j) = 0$ means it is still. The BITCEM is defined as:

$$\bar{i} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} i * P(i,j)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} P(i,j)} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} i\big|_{P(i,j)=1}}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} P(i,j)} \tag{1}$$

$$\bar{j} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} j * P(i,j)}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} P(i,j)} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} j\big|_{P(i.j)=1}}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} P(i,j)} \tag{2}$$

where $I(i,j)$ *: gray level of (i,j),*

$(\bar{i},\bar{j})$ *: coordinate of center of mass,*

(M,N) *: dimension of block*

Obviously, we can avoid the multiplication and division in calculating BITCEM by using Equations (1) and (2). Hence, this BITCEM formula greatly reduces the computation complexity.

2、Tracking of moving object direction and the BITCEM motion vector

Before we decide the moving object of one block, we further assume that the moving object of a block is larger than 5x5 pixel area to avoid the false decision caused by noise. Because the movement of the object would cause the gray level difference, we subtract the predicting block B_k from the reference block B_{k-1} to get the block difference (BD). A moving object is supposed to locate at the position that has the largest BD value.

Furthermore, we also use the motion vector of the same block in the reference frame to help us finding the reference gray level of the moving object. According to the motion vector, we can find pixel *(i',j')* with the largest gray level difference locating at the most remote part of that direction and then subtract the above coordinate by 5 to the reverse motion direction to get $I_k(\hat{i},\hat{j})$, the reference gray level of the moving object of block *k*.

After we get the reference gray level $I_k(\hat{i},\hat{j})$ of the moving object, we can use equations (1) and (2) to find the BITCEMs of the predicting block *B k* and the referenced block B_{k-1} of the moving object. The following are our steps:

（1）Use $I_k(\hat{i},\hat{j}) \pm TH$ as the threshold, if $I_k(i,j)$ falls within this range let $P_k(i,j)=1$, else let $P_k(i,j)=0$. Hence, P_k represents the bi-level pixels of the moving object in the predicting block B_k .

（2）Use equations (1) and (2) to get $(\bar{i}_k,\bar{j}_k)$, the BITCEM of the predicting block B_k .

（3）Use $I_k(\hat{i},\hat{j}) \pm TH$ as a threshold, if $I_{k-1}(i,j)$ falls within this range let $P_{k-1}(i,j)=1$, else let $P_{k-1}(i,j)=0$. *P* is the bi-level pixel of referenced block B_{k-1} .

（4）Use Equations (1) and (2) to get $(\bar{i}_{k-1},\bar{j}_{k-1})$, the BITCEM of the predicting block B_{k-1} .

The TH value is based on human visual perception of spatial characteristic. In this experiment, we choose Th = 40 to get the BITCEM motion vector (*m*1,*m*2) by the following equations.

$$m1 = Round(\bar{i}_k - \bar{i}_{k-1}) \tag{3}$$

$$m2 = Round(\bar{j}_k - \bar{j}_{k-1}) \tag{4}$$

In the above discussion, we assume the BITCEM motion vector equal to the motion vector of the moving object. In fact, the above assumptions do not always hold so that it would affect the precision of the BITCEM motion vector, hence instead of directly adopting the motion quantity of the BITCEM, we use the BITCEM of moving object to determine the direction of the block motion. In the following subsections, we will describe three more techniques to reduce computation further and achieve better picture quality.

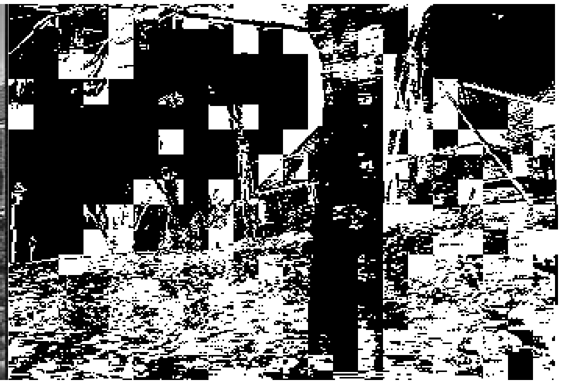
3、Sub-Sampling

In order to get the BITCEM of a 16x16block, 256x2subtractions and 256x2comparisons will be required. To reduce the searching points, the sub-samplings of the bi-level image with sub-sampling rate 1, 2, 4, 8 result in precision decreased. To trade off between computational complexity and picture quality, sub-sampling rate 4 would be an acceptable compromise.

Figure 1. shows the bi-level image of “Flower Garden” with sub-sampling rate 1, 2, 4, 8.

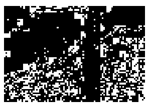
Figure 1. the bi-level image of “Flower Garden” with sub-sampling rate1, 2, 4, 8.

Original



Sub-Sampling rate = 1

Sub-Sampling rate = 2



Sub-Sampling rate = 4

Sub-Sampling rate = 8

4、Classification of Video Motion

In order to compensate the BITCEM motion error incurred by sub-sampling (especially for the fast moving frames), we use the percentage of the BITCEM with motion (0,0) to classify the video motion type. There are three types of video motion, Quasi-still, slow-movement, and fast-movement. Table 1 lists the classification of the video motion type according to the BITCEM (0,0) motion vector distribution. By using this information, we allocate different searching point patterns to different video motion type.

Table 1. Classification of video motion type Video type Still block percentage

Video type	Still block percentage
Quasi-Still	100%~93.75%(15/16)
Slow-Movement	93.75%~75%(12/16)
Fast-Movement	75%~0%(0/16)

5、Estimation of Initial Search point

From the published research [2], we find that the spatial and the temporal correlations of the moving blocks are the very important characteristics to be explored,

（1）In the consecutive frames, the moving object is almost at the same speed, so that motion vectors of the same block at consecutive frames has very high correlation.

（2）The motion vectors of the neighboring blocks in the same picture frame are almost the same.

Hence when the motion vectors of certain blocks are determined, we apply linear prediction model [8] to predict the motion vectors of the related blocks.

Let $MV(i, j, k)$ be the motion vector in the kth frame (i, j) block,

then $MV(i, j, k) = E[MV(i, j, k)] + dMV(i, j, k),$ (5)

The $E[MV(i, j, k)]$is the initial estimation of motion vector formulated as Eqn. (6).

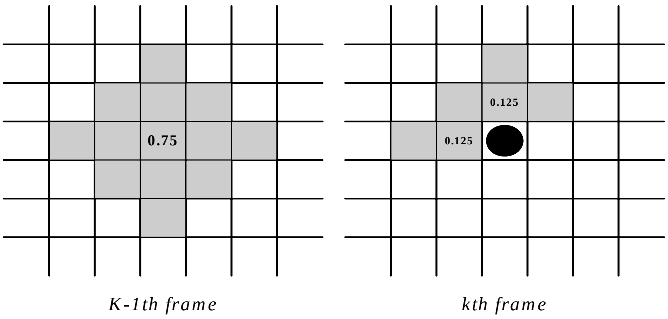
$$E[MV(i,j,k)] = \sum_{p,q\in W1}\lambda_{p,q,k} MV(i-p, j-q, k) + \sum_{p,q\in W2}\lambda_{p,q,k-1} MV(i-p, j-q, k-1) \quad (6)$$

where p, q are the coordinate difference of the neighboring blocks and the predicting block.

$\lambda_{p,q,k}$ and $\lambda_{p,q,k-1}$ are the weighting coefficients.

λ *p,q,k,* is the spatial correlation of *MV(i, j, k), and* λ *p,q,k-1is* the temporal correlation of *MV(i, j, k). W*1 and *W*2 are the selected area in the *kth and k-1th* frame, respectively. Figure 2. illustrates the concept of Linear Prediction using temporal-spatial correlations as formulated in Eqn. (6) .

Figure 2. the concept of Linear Prediction using temporal-spatial correlations.



III、The proposed fast block motion estimation scheme

In the proposed scheme, we use the percentage of (0,0) BITCEM motion vector of the previous frame to classify three video motion types. Based on the type, we can give different initial searching point. According to our experience, BITCEM motion vector is suitable to be the initial search point in quasi-still and slow-movement video motion type. As for the fast movement video motion type, initial search point estimated by linear prediction model will be more precise.

For near still and slow-movement frames, we focus the search on its surrounding points. For fast-movement frame we search more on outside points. We also introduce alternate checking point pattern into our scheme in figure 3.

According to the different situations of BITCEM motion, we propose different searching point patterns and different searching strategies for different directions of block motion and different video motion types, respectively in figure 3 to get more precise motion estimation.

As for the non near-still, whatever the block motion direction is, the first step search pattern contains 3x3 points close to the center in order to employ the characteristic of center-biased distribution and precisely estimate a slow motion sequence. As for the four types of motion direction (horizontal, vertical, sloped, inverse-sloped), in addition to the nine points close to the center, we add eight points in a circular shape around the center and four to six more points along the motion direction in order to predict a fast motion sequence and increase the precision of search along that direction. In the next step, we allocate four or eight points, alternatively around the best match candidate of the first step to get the final motion vector. The proposed algorithm is described as follows:

1、Compute the BITCEM motion vector by equation (1-4).

2、Classify the video motion types according to the percentage of (0,0) BITCEM motion vector.

3、Assign the initial motion vector according to the video motion type determined by step 2. BITCEM motion vector is suitable to be the initial search point of video types of quasi-still and slow-movement; for video types of fast movement, linear prediction model in equation (6) is used to compute the initial search point.

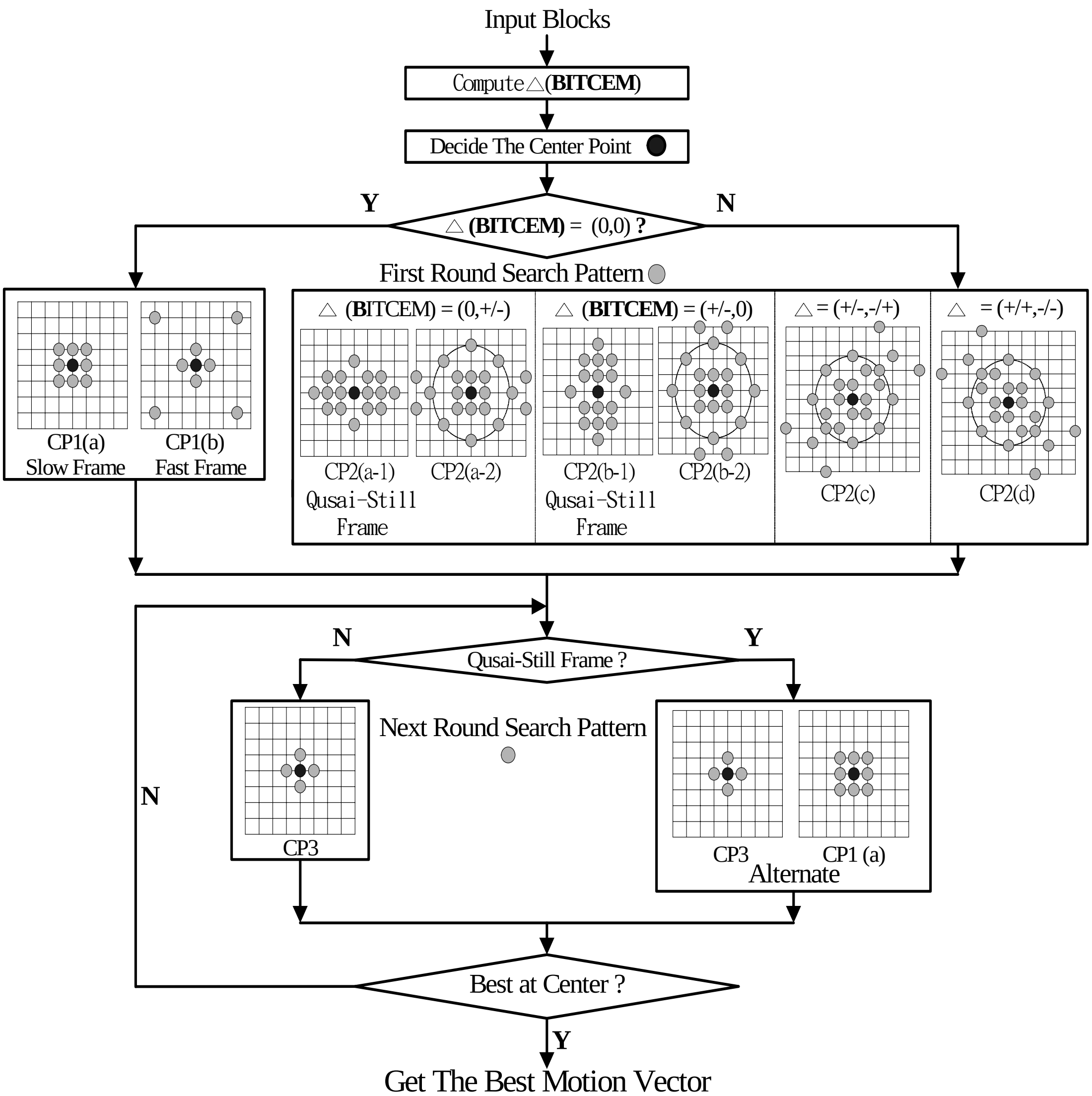
4、Refer to figure 1 to start the first round search point pattern from the initial search point.

5、Use the best matching point in step 4 as the initial search point to start the next round search pattern refers to figure 3.

6、If the best match point is (0,0), stop the search and go to step 7, else continue on searching with the next round search pattern.

7、Get the final motion vector.

Figure 3. The proposed block matching flow and the arrangement of search point patterns



IV、Experimental results

We compare the full search (FS), three step search (TSS), and four step search (4SS) with the proposed BITCEM algorithm.

We use the following criteria to evaluate the performance of each algorithm.

1、Average mean square error

Since we focus on the block matching not on the whole coding scheme, we only compare the difference between the predicted frame and the original frame. Note that we do not use the reconstructed frame for comparison to avoid the DCT quantization error.

2、Picture deterioration percentage

This term represents the MSE/pel of each algorithm divided by that of the FS algorithm.

3、Complexity/block

The complexity means the equivalent searching points of each algorithm. Take BITCEM as an example, since each search point needs 256 subtractions and 255 additions, the complexity of BITCEM is calculated as follows:

Complexity = Search Points + BITCEM Computation /511.

4、Speedup

This term represents the complexity/block of each algorithm divided by that of the FS algorithm.

From Table 2, we see that the proposed algorithm can greatly reduce the computational complexity so that it is faster than the TSS and 4SS in all sequences only except for the football case. The algorithm is faster 13 to 20 times than the FS. This result verifies the motive of our proposed scheme. That is, via the calculation of BITCEM motion vector, we can allocate the search points to the right position by giving the fast movement video the looser search point pattern and giving the quasi-still or slow-movement video the more concentrated search point pattern to avoid getting into local minimum.

In addition, the proposed algorithm still maintains a competitive picture quality although it has fewer search points. Except that the quality in Football is ranked the third among the compared algorithms, the quality in all other sequences is ranked the second just behind the FS algorithm.

V、Conclusion

In this paper, a fast block matching algorithm based on the video motion type using the technique of center of mass object tracking via binary transform is proposed. The main purpose of the proposed scheme is to predict the motion direction and motion quantity of the block for increasing the efficiency of matching process (including speed and precision). The key technology used includes the estimation of block moving direction (object moving direction), the arrangement of search point pattern, determination of the initial search point, classification of the video motion type, and the sub-sampling.

Experimental results show that the proposed algorithm is not only an efficient block matching algorithm to reduce the computational complexity, but also a pretty robust method to fit in all benchmark sequences.

References

B. Natarajan, V. Bhaskaran, and K. Konstantinides（1997，8）“Low- Complexity Block-Based Motion Estimation via One-Bit Transforms,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 7, No. 4.

J. Chalidabhongse, and C. -C. Jay Kuo（1997，6） “Fast Motion Vector Estimation Using Multiresolution-Spatio-Temporal Correlations,” IEEE Trans. On Circuits and Systems for Video Technology, Vol. 7, No. 3, pp. 477-488.

J. D. Gibson, T. Berger, T. Lookabaugh, and D. Lindbergh（1998）Digital Compression for Multimedia, Morrgan Kaufmann Publishers.

L. M. Po, and W. C. Ma（1996，1） “A Novel Four-Step Search Algorithm for Fast Block Motion Estimation,” IEEE Tran. on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 313-317.

Po-Hung Chen, Hung-Ming Chen, Kuo-Liang Yeh, Mon-Chau Shie and Feipei Lai（2001）"BITCEM: An Adaptive Block Motion Estimation Based on Center of Mass Object Tracking via Binary Transform," 2001 IEEE Int'l Symp. On Intelligent Signal Processing and Communication Systems, Nashville, Tenn., USA, No. 20-23. (Proc.)

R. Jain, R. Kasturi, and B. G. Schunck（1995）Machine Vision, McGraw-Hill.

R. Li, B. Zeng, and M. L. Liou（1994，8）“A New Three-Step Search Algorithm for Block Motion Estimation,” IEEE Tran. On Circuits and Systems for Video Technology, Vol. 4, No. 4, pp. 438-442.

S. Haykin（1996）Adaptive Filter Thoery, 3rd ed., Prentice Hall.

Table 2. Performance comparison of BITCEM, FS, TSS, and 4SS

Sequences		Search Algorithm	FS	TSS	4SS	Proposed
		MSE/pel	8.83	8.99	9.03	8.98
	CLAIRE	Deterioration %	0.0	1.8	2.3	1.7
	(352*288*91)	Complexity/block	204.28	23.28	17.37	10.00
		Speedup %	1.0	8.8	11.8	20.4
		MSE/pel	11.78	12.4	12.23	12.13
	MISS AMERICA	Deterioration %	0.0	5.3	3.8	3.0
		Complexity/block	204.28	23.5	18.65	10.59
	(352*288*91)	Speedup %	1.0	8.7	11.0	19.3
		MSE/pel	26.93	27.84	27.62	27.25
	SALESMAN	Deterioration %	0.0	3.4	2.6	1.2
	(352*288*91)	Complexity/block	204.28	23.24	16.21	9.36
		Speedup %	1.0	8.8	12.6	21.8
		MSE/pel	266.19	312.02	289.02	276.61
	FLOWER	Deterioration %	0.0	17.2	8.6	3.9
	(352*240*91)	Complexity/block	202.05	23.25	18.89	14.46
		Speedup %	1.0	8.7	10.7	14.0
		MSE/pel	380.65	411.85	424.17	417.15
	FOOTBALL	Deterioration %	0.0	8.2	11.4	9.6
	(352*240*91)	Complexity/block	202.05	23.09	18.04	14.42
		Speedup %	1.0	8.8	11.2	14.0

Sequences		Search Algorithm	FS	TSS	4SS	Proposed
	TABLE TENNIS (352*240*91)	MSE/pel	182.26	237.12	210.95	209.36
		Deterioration %	0.0	30.1	15.7	14.9
		Complexity/block	202.05	23.32	19.06	14.26
		Speedup %	1.0	8.7	10.6	14.2
	MOBIL & CALENDER (352*240*91)	MSE/pel	348.54	362.18	354.82	349.96
		Deterioration %	0.0	3.9	1.8	0.4
		Complexity/block	202.05	23.02	15.85	12.39
		Speedup %	1.0	8.8	12.7	16.3
	BIKE (352*240*91)	MSE/pel	39.65	42.77	41.85	41.73
		Deterioration %	0.0	7.9	5.5	5.2
		Complexity/block	202.05	23.18	19.22	13.76
		Speedup %	1.0	8.7	10.5	14.7
	CARPHONE (176*144*91)	MSE/pel	41.03	42.82	42.82	41.98
		Deterioration %	0.0	4.4	4.4	2.3
		Complexity/block	184.56	21.6	15.98	10.23
		Speedup %	1.0	8.5	11.5	18.0