Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Two Fast Motion Estimation Algorithms by Using Spatial & Temporal Correlation

Shen-Chuan Tai , Wen-Jan Chen and Gem-Cher Lin
Institute of Electrical Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.
e-mail:sctai@mail.ncku.edu.tw

Abstract

A good motion estimation algorithm shall use the historic information to provide hints on where to begin the search process. To obtain high performance the properties of motion vectors are studied. From these studies of motion vectors, it is found that the motion vector lengths are short for most cases.

Based on the studies, two algorithms for fast motion estimation are proposed. Experimental results show that they achieve significant saving on computation time. To particular, are as fast as the 3-steps search method the reconstructed results are nearly the same as that by the full search method.

1 Introduction

Recently the digital technology have led to new communication media in which visual plays the key role. Digital TV, high definition TV, video conferencing, and multimedia are some examples of emerging applications. As compared with audio or text information, video signals require a huge amount of bandwidth. Despite the increase of storage capacity and the development of broadband networks, compression techniques are still needed.

Data compression [1-4] plays the key role in many applications of digital video. The audio bandwidth, after all, is about 20 kHz, which translates into a digital data of about 1.4 megabits per second for high -quality stereo sound. Sampled video source signals, on the other hand, require much higher bit rates, ranging from 10 Mb/s for broadcast-quality video to more than 100 Mb/s for HDTV signals.

Even when still pictures are involved, as in image archival systems, a mountain of data is needed to represent them. For example, a color image with resolution of 1000 by 1000 picture elements (pixels) at 24 bits each will occupy 3 megabytes of storage in an uncompressed form. This will not fit onto a high-density floppy diskette, which can hold just 1.2 Mbytes.

In order to utilize digital images effectively, specific techniques are needed to reduce the number of bits required for their representations. The branch of digital image processing that deals with this problem is called image data compression. Image data compression, which is the art of efficient coding of image data, aims at taking advantage of the redundancy to reduce the number of bits required to represent an image. This can result in significant saving in the memory needed for image storage or in the channel capacity required for image transmission.

For video signals, the process of estimating motion vectors during the encoding process is called motion estimation. The process of reconstructing an image using image blocks from the previous image along with information about motion is called motion compensation. The motion estimation and compensation processes are conceptually quite simple but they may be very expensive in terms of computation time.

The block must be matched from the previous image to various positions on the new image and calculate the summation of difference at each position's point. It is possible to determine mathematically whether the difference at a given location is low enough to constitute a match. If a match is found , the new block can be represented as a displacement of previous block rather than actually transmitting the entire block of pixels.

The displacement of motion vectors is usually limited to less than or equal to the block size. One reason for this is it would be very time consuming to search a large area of the image for each motion vector calculated. The use of motion estimation and compensation has two requirements on the encode and decode system. One is the decoder must store the previous image while reconstructing the next image. The other is the encoder must reconstruct each image after encoding it to predict how the decoder will reconstruct the image.

Each block in the current frame estimates its motion vector by evaluating some matching criteria over the blocks in the previous frame and selecting the block which yields the closest matching. There are many choices for the matching criterion, e.g., mean square error (MSE), mean absolute difference (MAD), etc. The MAD is adopted in this paper because it has better performance in matching.

In this paper, we describe the characters of the motion analysis in the next section. Two fast motion estimation algorithms are presented in Section 3. In Section 4, experimental results are compared with the full search algorithm and three-steps algorithm.

2. Motion analysis

The most accurate but most expensive approach is the exhaustive search. It calculates the pixel matching error at every possible position within a search region. If it is desired to improve the performance of search, the search range size should be adjusted. On this condition we will achieve less expensive than exhaustive search but more prone to errors to local minimum.

A better motion estimation algorithm [5-9] can use the historic information to provide clues on where to begin the search process. In other words , the motion vectors in the current frame can be predicted from those neighboring blocks in the temporal or spatial direction. Most frame objects are often larger than the blocks of individual pixel. It is indicated that once the displacement vector has been determined for a given pixel block, it is of a high probability that other pixel blocks in the same vicinity will have a similar or even identical motion vector.

Motion analysis [10-13] studies provide us with efficient estimate schemes to estimate dense optical flow fields. However, for image sequence coding where motion features have to be transmitted, an adaptive compact motion representation is required to extract temporal redundancy from an image sequence with a minimum amount of side information and, never with a high quality of reconstruction.

Consider the main properties of video sequence. It is found that many video sequences have the property that moving objects only exist in some small fixed area and the background remains static. This phenomenon is specially suitable for video conference signals.

We have achieved some statistical results for four typical motion sequences show in Figure 1 by the full search method. Referring to Figure 2, it is found that 3% of the lengths of the motion vectors in susic video sequence are zero. And 53% of the lengths of the motion vectors in susic video sequence are one. And 29% of the lengths of the motion vectors in susic video sequence are two. Other lengths of motion vectors occupy 15 % only. According to the distribution of the motion vector lengths, it can be said that the video sequence of susic is of slow motion.

As shown in Figure 3, it is found that 42% of the lengths of the motion vectors in mobil video sequence are zero. And 38% of the lengths of the motion vectors in mobil video sequence are one. And 4% of the lengths of the motion vectors in mobil video sequence are two. Other lengths of motion vectors occupy 16% only. According to the distribution of the motion vector lengths, it can be said that the video sequence of mobil is of slow motion, too.

From Figure 4, it is found that 6% of the lengths of the motion vectors in windmill video sequence are zero. And 53% of the lengths of the motion vectors in windmill video sequence are one. And 21% of the lengths of the motion vectors in windmill video sequence are two. Other lengths of motion vectors occupy 20% only. According to the distribution of the motion vector lengths, it can be said that the video sequence of windmill is of fast motion.

As can be seen from Fig. 5, it is found that 2% of the lengths of the motion vectors in table tennis video sequence are zero. And the 24% of lengths of the motion vectors in table tennis video sequence are one. And 25% of the lengths of the motion vectors in table tennis video sequence are two. Other lengths of motion vectors occupy 49%. According to the distribution of the motion vector length it can be said that the video sequence of table tennis is of fast motion.

For video sequence compression, the

properties of motion as described above should be considered and utilized. In this paper, two new efficient motion estimation schemes are developed which utilize these properties. By these two algorithms, we can achieve significant saving on computation time.

3.Two fast algorithms

In this section, two algorithms for fast motion estimation are proposed. They are named as Algorithm1 and Algorithm2, respectively.

Refer to Algorithm1 shown below. Assume that the motion is linear. By using the results in section 2, a displaced, reduced search area is selected for each block.

Step1 :Use the fullsearch method to find the motion vector of each block for each refreshed frame, save motion vector information for each image block.
Step2 :Use previous frame's motion vector information to decide a displaced , reduced search area for each block in the current frame.
Step3 :Search through the reduced range found in step2 for each block to find the motion vectors information and save the information for next
Step4 :If the next frame is a refreshed frame then go to step 1,otherwiese go to step2.

Fast Motion Estimation Algorithm 1

Figure 6 shows the displaced, reduced search area used by Algorithm1. It uses the motion vector length of block(i,j,t-1) to decide the searching range of block(i,j,t) and uses the position of reference block which is the best match block of block(i,j,t-1) as the new searching center of block(i,j,t).

Refer to Algorithm2 show below. Suppose that the motion is nonlinear. In Algorithm2, the motion vector information of block(i,j,t-1) is only used to bound the search range of block(i,j,t). As shown in Figure 7, Algorithm2 uses the motion vector length of block(i,j,t-1) to bound the search range of block(i,j,t).

Consider the size of search range for Algorithm1 and Algorithm2. Since most motion vector lengths are short as indicated by the results in Section 2 and the inter-relation between the motion vectors of the corresponding blocks of two consecutive frames is strong, it is quite natural to adaptively set the size of the search range according to the motion vector length of the corresponding block of the previous frame, as shown in Figure 6 and Figure 7.

Step1 :Use the fullsearch method to find the motion vector of each block for each refreshed frame, save motion vector information for each image block.
Step2 :Use previous frame's motion vector information to decide a reduced search area for each block in the current frame.
Step3 :Search through the reduced range found in step2 for each block to find the motion vectors information and save the information for next
Step4 :If the next frame is a refreshed frame then go to step 1,otherwiese go to step2.

Fast Motion Estimation Algorithm 2

In Algorithm1 and Algorithm2, the size of search range is set to be a square of (2 x motion vector length + 1) by (2 x motion vector length + 1), unless the motion vector length is equal to 0 or larger than the width of a block. If the motion vector length is 0 for block(i,j,t-1), the search range is set to be 3 x 3. And if the motion vector length is larger than the width of a block, the search range is set to be a square of (2 x block width + 1) by (2 x block width + 1). In all cases, the block to be encoded is located at the center of the search range.

For example, if the length of the motion vector of block(i,j,t-1) is 1, the number of searching points for block(i,j,t) is equal to $(2 \times 1 + 1)^2$. If the length of the motion vector of block(i,j,t-1) is 2, the number of search points for block(i,j,t) is equal to $(2 \times 2 + 1)^2$ and so on.

Figure 8 shows that an extra component, which is a buffer for saving the motion vectors of previous frame, to support the work of motion estimation is needed by Algorithm1 and Algorithm2. This overhead, however, can be ignored in motion estimation encoder because of its simplicity.

4. Simulation results

To evaluate the efficiency and performance of the proposed algorithms, Algorithm1, Algorithm2, three steps , new three steps(NTSS)[14] and the full search methods have been coded in C language and ran on SUN SPARC 10. Four image sequence frames "SUSIC", "MOBIL", "WINDMILL", "TABLE TENNIS" of size 360 x 240 with 8-bit gray scales, are used. Every image sequence contains 140 frames and the encoding block size is 8 x 8.

The results for the "**SUSIC**" video image frames are shown in Figure 9. It is found that both Algorithm1 and Algorithm2 achieve results near that by the full search method, where Y axis is for the peak signal to noise ratio (PSNR) and X axis is for the frame number. For most of the reconstructed video image frames, it is obvious that Algorithm1 and Algorithm2 have better PSNR values than TSS and NTSS methods. The results for the "**MOBIL**" video image frames are shown in Figure 10, the results for the "**WINDMILL**" video image frames are shown in Figure 11 and the results for the "**TABLE TENNIS**" video frames are shown in Figure 12. The results are the same as the case of "**SUSIC**" video image frames, i.e., both Algorithm1 and Algorithm2 achieve better results than the TSS and NTSS methods.

Refer to Figure 13 which shows the average search points needed for each method. As can be seen, the number of points searched by Algorithm1 and Algorithm2 are about the same as that needed by TSS method, and are much fewer than that needed by the full search method. Despite of this, Algorithm1 and Algorithm2 still achieve results near that by the full search method and achieve 3 to 10 dB improvement on that by the TSS method, as can be seen from Figure 14. Table 1 and Table 2 summarize the results.

5. Conclusions

In this paper, two fast motion estimation algorithms for the image sequence coding are proposed. It uses the temporal motion vector information to reduce the search range for motion estimation. By this, a significant improvement on saving the computation time is achieved.

According to the motion analysis, there is a strong interframe correlation. The statistics information of motion vector length for typical image sequences are obtained. These information help us in deciding the average displacement of motion vector.

The first algorithm uses the motion vector information of the previous block to reduce the current encoding block's search range and assumes that the motion is linear. The second algorithm also uses the information of previous motion vectors to reduce the motion estimation search range. But it assumes that the motion is nonlinear. Experimental results show that both algorithms achieve PSNR near that by the full search method while the number of points searched by them are only about the same as that by the TSS method.

Our experiment results also show that the proposed algorithms are especially suitable for slow motion signals. They takes only 1/20 of the computation time need by the full search method but the reconstructed image quality is very close that by the full search method, e.g., the SUSIC and WINDMILL sequences as shown in Table 1 and Table 2.

6. References

[1] Lee D. Davission and Robert M. Gray, Data Compression, Dowden, Hutchison & Ross, Inc. 1976.

[2] Majid Rabbani and Paul W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, 1991.

[3] Lynch, Data Compression Techniques and Applications, A Division of Wadsworth, Inc. 1985.

[4] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Addison Wesley, 1992.

[5] S. Kim and C.-C. J.Kuo, "Adaptive motion estimation in video coding with a stochastic model," Proceeding of the IEEE First International Conference on Image Processing, Austin, Texas, November 13 - 16 , 1994.

[6] S. Kim and C.-C. Kuo, "Fast motion vector estimation with a Markov model for MPEG," Proceeding of ISET/SPIE Symposium on Electronic Imaging: Science & Technology, San Jose. California, February 5 - 10, 1995.

[7] MPEG, "Generic coding of moving pictures and associated audio," tech. rep., ISO/IEC 13818-2, Recommendation H.262, Committee Draft, March, 1994.

[8] Mei-Juan Chen, Liang-Gee Chen and Tzi-Dar Chiueh,"One-Dimensional Full Search Motion Estimation Algorithms for Video Coding", IEEE Trans. on Circuits and Systems for Video Technology, VOL. 4, NO. 5 OCTOBER 1994.

[9] H. G. Musmann, P. Pirsch, and H. J. Gralleer, "Advances in picture coding, " Proc. IEEE, vol. 73, no. 4, pp. 523-548, Apr.1985.

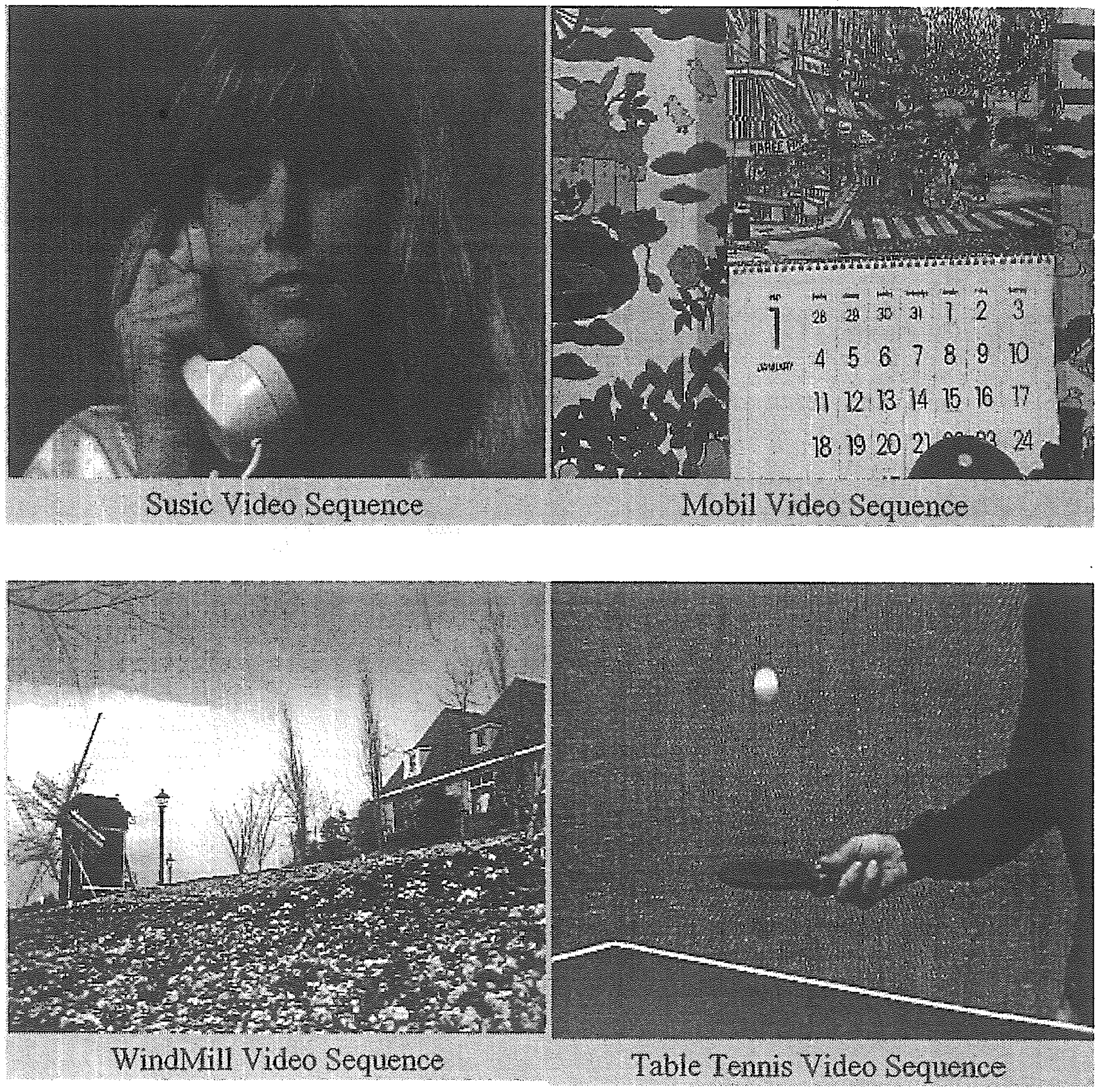
[10] Y. T. Tse And R. L. Baker, "Global zoom/pan estimation and compensation for video compression," in *IEEE Proc. ICASSP'91*, Toronto, Canada, Vol. 4, pp. 2725 - 2728, May 1991.

[11] M. Allmen and C.R. Dyer. Cylic motion detection using spatiotemporal surface and curves. In Proc. Int. Conf. on Pattern Recognition, pp. 365-370, 1990.

[12] G. Johansson. Visual perception of biological motion and a model for its analysis. Perception and Psychophysics, pp. 14:201-211, 1973.

[13]M. I. Sezan and R. L. Lagendijk, Motion analysis and image sequence processing, Kluwer Academic Publishers, 1993.

[14] R. Li, B. Zeng and M. L. Liou,"A new Three-Step search Algorithm for block Motion Estimation", IEEE Trans. on Circuits and Systems for Video Technology, VOL. 4, NO. 4 pp. 438-442 August 1994.



Susic Video Sequence

Mobil Video Sequence

WindMill Video Sequence

Table Tennis Video Sequence

Fig. 1 The four tested video sequence frames

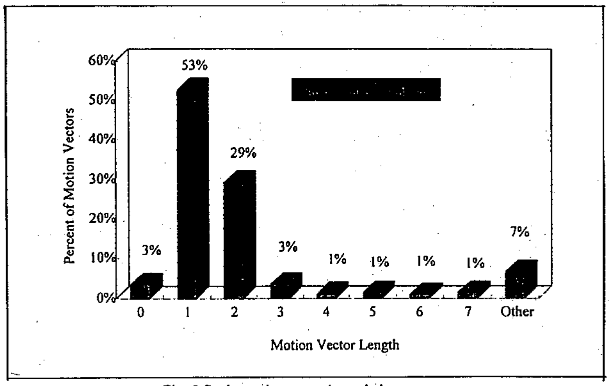


Fig. 2 Susic motion vector's statistics.

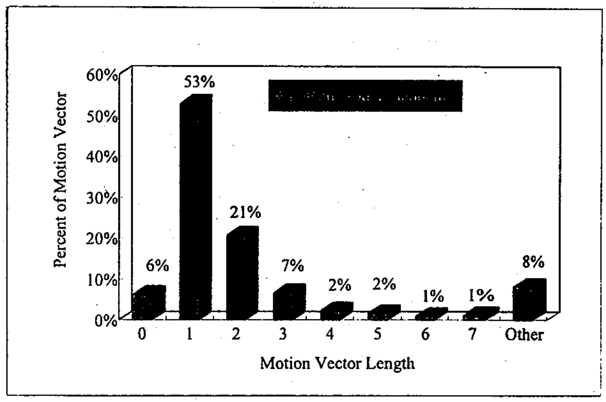


Fig. 4 WindMilll motion vector's statistics.

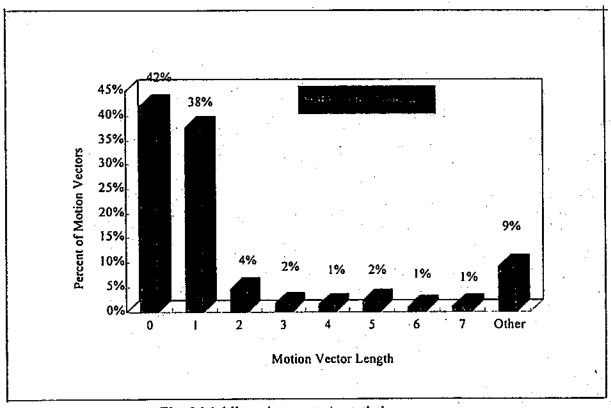


Fig. 3 Mobil motion vector's statistics.

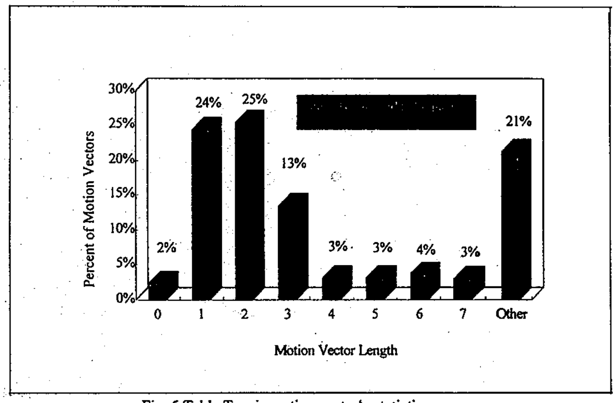


Fig. 5 Table Tennis motion vector's statistics.

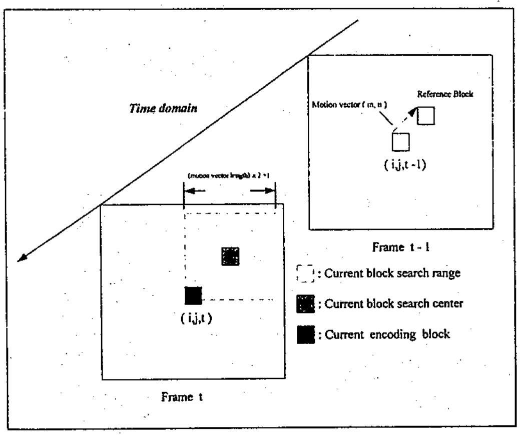


Fig. 6 Motion Estimation Block Diagram for Algorithm1.

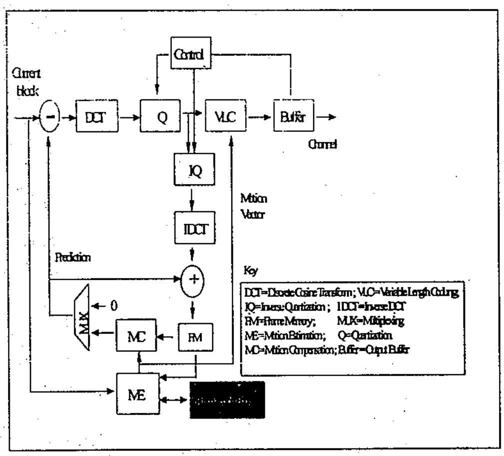


Fig. 8 A buffer is added to the MPEG encoder.

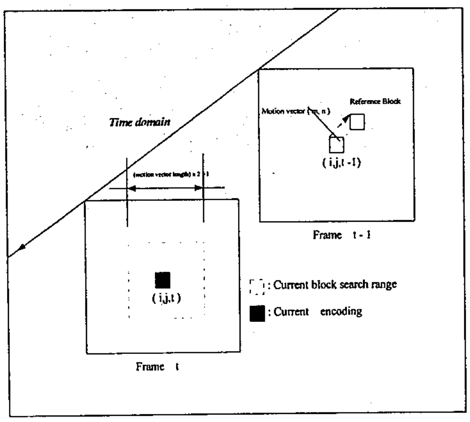


Fig. 7 Motion Estimation Block Diagram for Algorithm2.

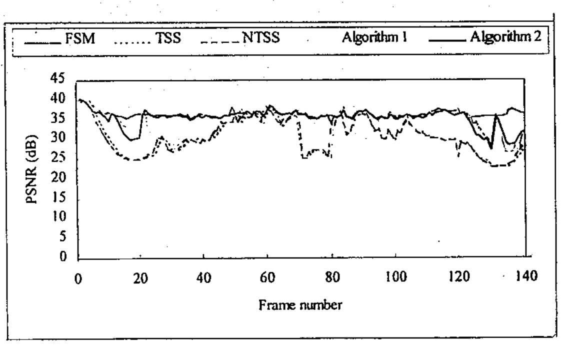


Fig. 9 Results of ***SUSIC*** video image sequences.

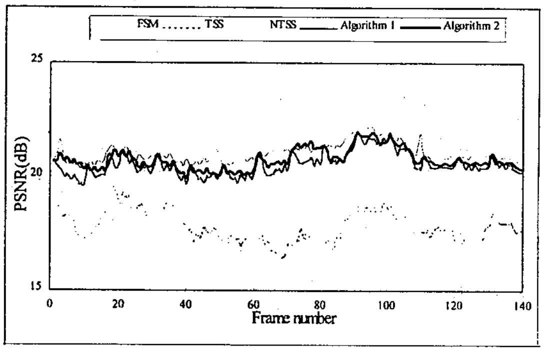


Fig. 10 Results of ***MOBIL*** video image sequence.

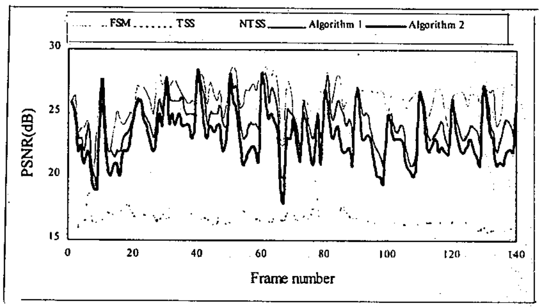


Fig. 11 Result of ***WINDMILL*** video image sequence.

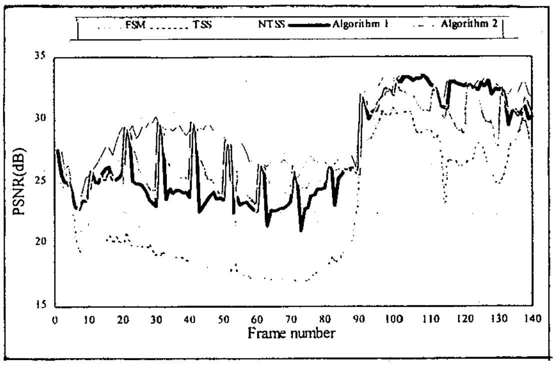


Fig. 12 Results of ***TABLE TENNIS*** video image sequence.

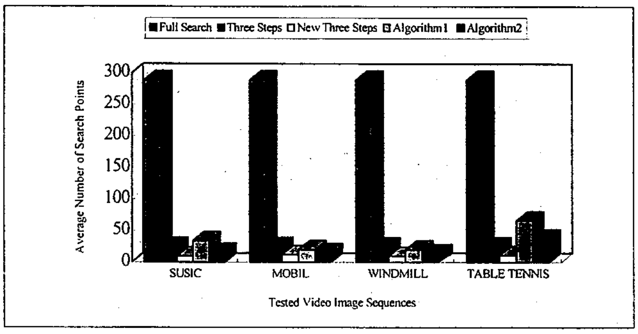


Fig. 13 ***Average Search Points*** needed for each method.

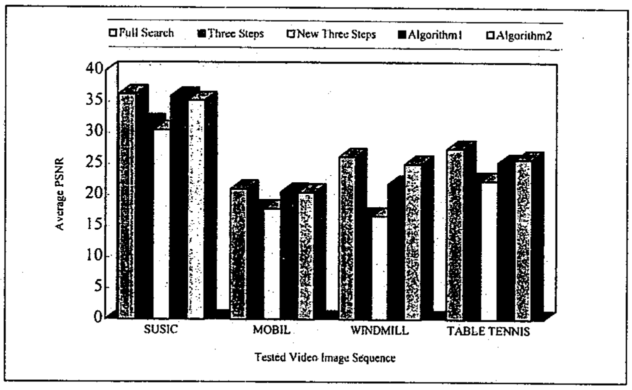


Fig. 14 ***Average PSNR achieved for each method.***

Table.1 Comparisons of the ***average search points*** needed by various algorithms.

	Full Search	Three Steps	New Three Steps	Algorithm1	Algorithm2
SUSIC	225	25	20	34	16
MOBIL	225	25	22	20	15
WINDMILL	225	25	21	20	14
TABLE TENNIS	225	25	22	66	38

Table.2 Comparisons of the ***Average PSNR*** achieved by various algorithms.

	Full Search	Three Steps	New Three Steps	Algorithm1	Algorithm2
SUSIC	36.23	31.58	30.48	35.88	35.22
MOBIL	21.01	17.84	17.86	20.54	20.46
WINDMILL	26.20	16.62	16.68	21.80	25.02
TABLE TENNIS	27.50	22.35	23.35	25.32	25.82