

Multiplication-Free One-Bit Transform and Diamond Search Combination for Fast Binary Block Motion Estimation

Eun-Seo Lee¹, Oğuzhan Urhan^{1,2}, Tae-Gyu Chang¹

¹School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, South Korea.

²Kocaeli University Laboratory of Image and Signal Processing (KULIS), Electronics and Telecom. Eng. Dept., Kocaeli University, TR.

¹ikart@dmc.cau.ac.kr , ²urhano@kou.edu.tr , ³tgchang@cau.ac.kr

Abstract

Combination of multiplication-free one-bit transform based block estimation with the diamond sparse search pattern is presented in this paper. Unification of sparse search patterns with low-bit representation motion estimation methods are rarely investigated and therefore one-bit transform based approaches are executed using full-search approach mostly. Our experiments show that such a combination can be carried out with acceptable performance fall.

1. Introduction

Usage of multimedia information has been increasing every day due to increasing computational resources and effective compression methods. Generally, video data constitutes most of the multimedia data. Efficient coding of video is important for effectual usage of limited bandwidth and storage medium. Temporal correlation between successive image frames enables high amount of compression. Motion estimation is an important tool for exploiting temporal correlation. Block based motion estimation with non-overlapping rectangular blocks is used many video coding standards. In this case, image frames are divided into non-overlapping blocks and the best match is searched around a pre-defined search range using all possible positions for each block. Though this full search method provides optimal quality it significantly suffers from computational load. Approximately more than half of the video compression time is consumed in motion estimation for many video coding methods [1]. Therefore, lots of works have been carried out to reduce computational burden of motion estimation.

Reduction of number of search points using various patterns has been investigated for more than 25 years. Some of the commonly known approaches are three-step search (3SS) [2], new three-step search (NTSS) [3], four-step search (4SS) [4], 2D-logarithm search (2DLOG) [5], and diamond search (DS) [6]. Among them DS generally provides better result using fewer search points.

Another way to reduce computational load is to use lower bit representations. One-bit transform (1BT) based motion estimation method proposed in [7] firstly converts 8-bit image frames into one-bit binary images then computes the matching criteria using Boolean exclusive OR (XOR) operation. Two-bit transform (2BT) based motion estimation in [8] forms two bit-plane using block mean and variances and aging uses XOR based operations in the computation of matching criteria. This approach provides better performance at the expense of increased computational load. Recently proposed

multiplication-free one-bit transform (MF-1BT) based motion estimation approach in [9] aim to obtain binary image frames using only integer arithmetic simply changing number of non-zero in filtering kernel to enable faster implementations. Another recently proposed approach called constrained one-bit transform (C-1BT) motion estimation approach presented in [10] forms a constrain mask to further improve performance introducing a slight computational cost. 1BT, 2BT, MF-1BT, and C-1BT methods compute matching criteria using only XOR based operations which is suitable for hardware implementations. Among them, MF-1BT aims to reduce computational load keeping performance similar level, whereas 2BT and C-1BT methods target to improve performance at the expense of some more computational load. Modified one-bit transform (M1BT) in [11] and modified two-bit transform (M2BT) in [12] uses an additional two-step search approach in 8-bit domain to further improve the performance. However, these additional searches in 8-bit domain prevent full binary implementations.

Despite all this effort in fast binary motion estimation, combination of sparse search point procedure with lower-bit representation approaches rarely investigated in the literature up to now. Only unification with 2DLOG search is presented in original 1BT work [7]. We aim to further reduce computational load of binary motion estimation approaches using sparse search point approaches in this work. We prefer to use MF-1BT based approach since it has the lowest total computational load among low-bit representation approaches. Our experiments exhibit that combination of DS approach with MF-1BT provides reasonable results.

2. 1BT and MF-1BT Based Motion Estimation

Binary image frames used in 1BT are obtained filtering images frames with a multi band-pass filter and then comparing them with filtered images. The multi band-pass filter in 1BT defined as in (1).

$$K(i, j) = \begin{cases} 1/25, & \text{if } i, j \in [1, 4, 8, 12, 16] \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Let I and I_F represent original and filtered image frames respectively. Then, we can obtain the one-bit images, which will be used matching process, as in (2).

$$B(i, j) = \begin{cases} 1, & \text{if } I(i, j) \geq I_F(i, j) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Since the kernel used in 1BT based motion estimation has 25 non-zero component, required normalization can be performed using floating point arithmetic which relatively slower then integer arithmetic. MF-1BT approach presented in [9] uses a kernel that contain 16 non-zero component which enable integer normalization operation using shifting operation. The kernel used in [9] is defined as in (3).

$$K = \frac{1}{16} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

An example image frame and its one-bit form obtained using MF-1BT kernel is given in Fig 1.

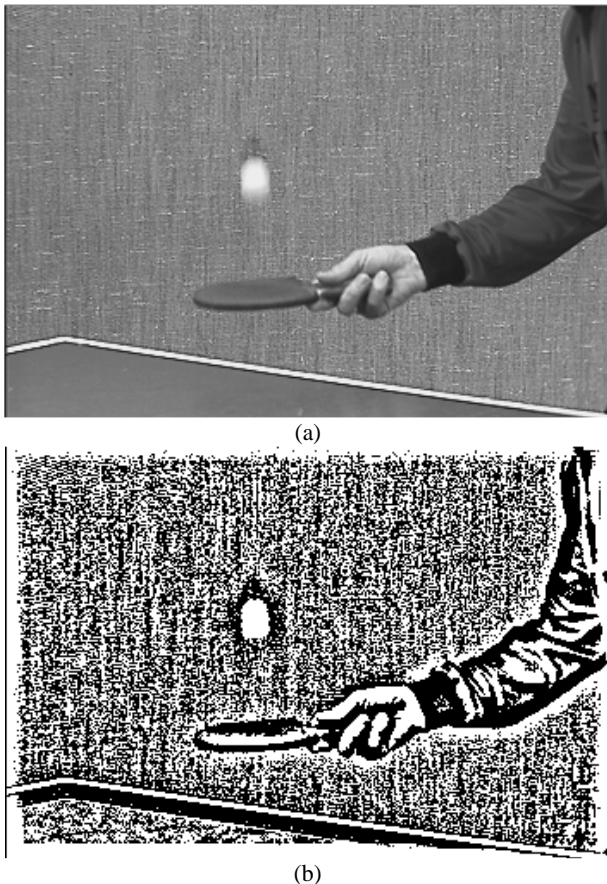


Figure 1: (a) An example image frame and (b) its corresponding one-bit image obtained using kernel proposed in MF-1BT based approach.

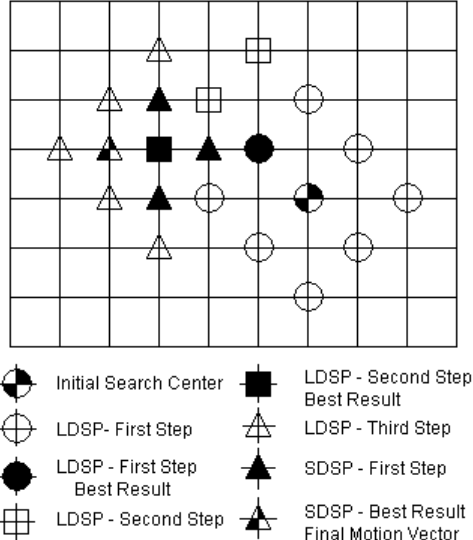


Figure 2: DS procedure.

The matching criteria, number of non-matching points (NNMP), used in MF-1BT method is defined as in (4).

$$NNMP(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B'(i, j) \oplus B'^{-1}(i+m, j+n)\} \quad (4)$$

$$-s \leq m, n \leq s-1$$

where (m,n) shows the candidate displacement, s determines the search range, and \oplus shows the exclusive OR (EX-OR) operation.

3. Diamond Search Method

Proposed method in this work targets combination of MF-1BT motion estimation approach with DS sparse search pattern. We have investigated unification with various sparse search approaches such as 3SS, N3SS, 4SS, 2DLOGS, and DS and found that DS provides better results.

DS approach uses two patterns for block motion estimation. Firstly, Large Diamond Search Pattern (LDSP) is used around a search center. The search center is initially determined as zero motion position. The matching error is computed for 9 initial search points. The point giving minimum matching error is defined as new search center. If this point is the previous search center then Small Diamond Search Pattern (SDSP) is used. The point giving minimum error at this stage is accepted as final motion vector. Otherwise, LDSP is used until the search center gives minimum error. DS procedure is depicted in Fig. 2.

4. Experimental Results

Proposed MF-1BT and DS combination is compared against various sparse search point combinations. Performance of motion estimation methods is evaluated using Peak Signal to Noise Ratio (PSNR) and number of search point (NSP) metrics. Fig. 3, 4 and 5 show PSNR and NSP results for 30 frames of Caltrain, Tennis, and Garden sequences respectively. It is clearly seen from these figures that the combination of the

MF-1BT with 4SS and DS provides similar PSNR performance with respect to full search based MF-1BT approach. However, when we examine NSP results it is shown that the DS achieve this performance using fewer search point. The full search approach requires more than 200 search point for this case. This means that proposed approach can be executed at least 10 times faster with respect plain MF-1BT having similar performance. It is noted in [7] that 1BT based motion estimation can be executed 15 times faster than classical SAD based approach. Taking into account this assumption we may conclude that the proposed approach can be executed more than 150 times faster with respect to SAD based full search approach.

5. Conclusions

MF-1BT based motion estimation approach is combined with the DS approach to further reduce computational load of MF-1BT approach. We have examined various combinations of MF-1BT and sparse search patterns. Our experiments show that DS combination can be executed having acceptable PSNR performance and number of search point.

6. Acknowledgements

This work is supported by the professor invitation program of IITA, Korea.

7. References

- [1] He, Z.L., Tsui, C.Y., Chan, K.K., and Liou, M.L., "Low-power VLSI design for motion estimation using adaptive pixel truncation", *IEEE Trans. Circuit Syst. Video Technol.* 10(5):669-678, 2000.
- [2] Koga, T., Iinuma, K., Hirano, A., Iijima, Y., and Ishiguro, T., "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, Nov./Dec. 1981, pp. G5.3.1-G5.3.5.
- [3] Li, R., Zeng, B., and Liou, M.L. "A New Three-Step Search Algorithm for Block Motion Estimation", *IEEE Trans. Circuits Syst. Video Technol.*, 4(4):438-442, 1994.
- [4] Po, L.M. and Ma, W.C., "A novel four-step search algorithm for fast block motion estimation", *IEEE Trans. Circuits Syst. Video Technol.*, 6(3):313-317, 1996.
- [5] Jain, J.R. and Jain, A.K., "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29(12):1799-1808, 1981.
- [6] Zhu, S. and Ma, K-K., "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", *IEEE Trans. Image Process.*, 9(2):287-290, 2000.
- [7] Natarajan, B., Bhaskaran, V., and Konstantinides, K., "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. Circuit Syst. Video Technol.*, 7(4):702 - 706, 1997.
- [8] Ertürk, A. and Ertürk, S., "Two-Bit Transform for Binary Block Motion Estimation," *IEEE Trans. Circuit Syst. Video Technol.*, 15(7):938- 946, 2005.
- [9] Ertürk, S., "Multiplication-free one-bit transform for low-complexity block-based motion estimation", *IEEE Signal Process. Lett.*, 14(2):109-112, 2007.

- [10] Urhan, O. and Ertürk, S., "Constrained One-Bit Transform for Low Complexity Block Motion Estimation", *IEEE Trans. Circuit Syst. Video Technol.*, 17(4): 478-482, 2007.
- [11] Wong, P.H.W. and Au, O.C., "Modified One-Bit Transform for Motion Estimation," *IEEE Trans. Circuit Syst. Video Technol.*, 9(7):1020 - 1024, 1999.
- [12] Demir, B. and Ertürk, S., "Block Motion Estimation using Modified Two-Bit Transform", *Lecture Notes in Computer Sciences*, 4263, 522-531, 2006.

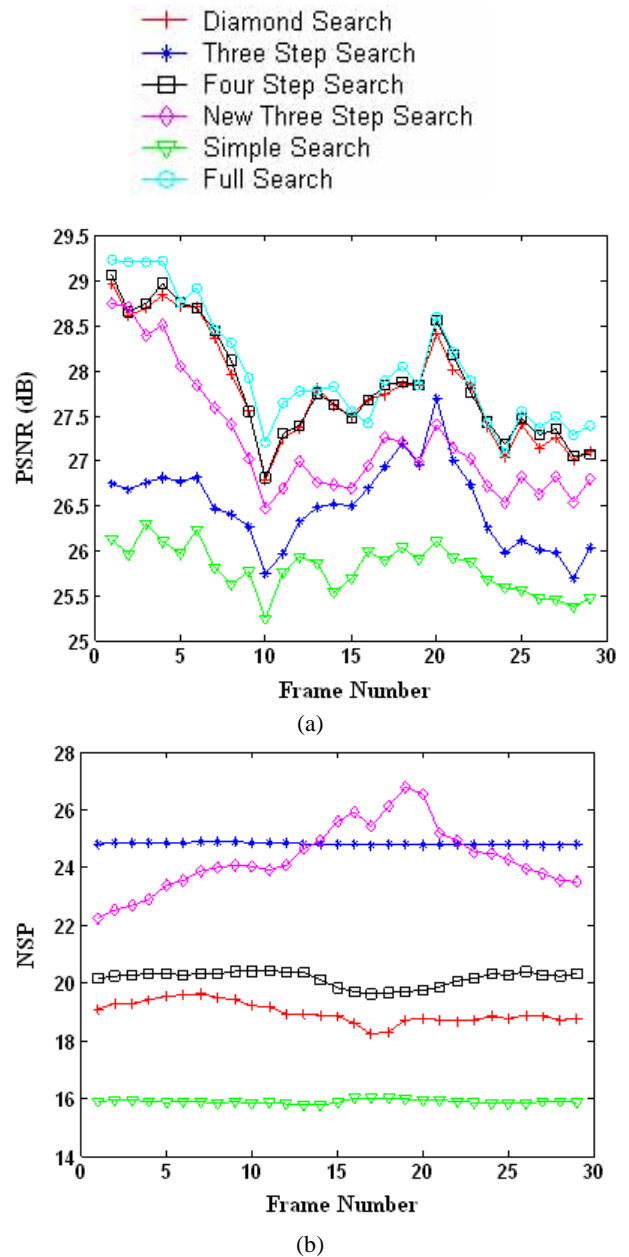


Figure 3: PSNR performance and NSP of various combinations for Caltrain sequence.

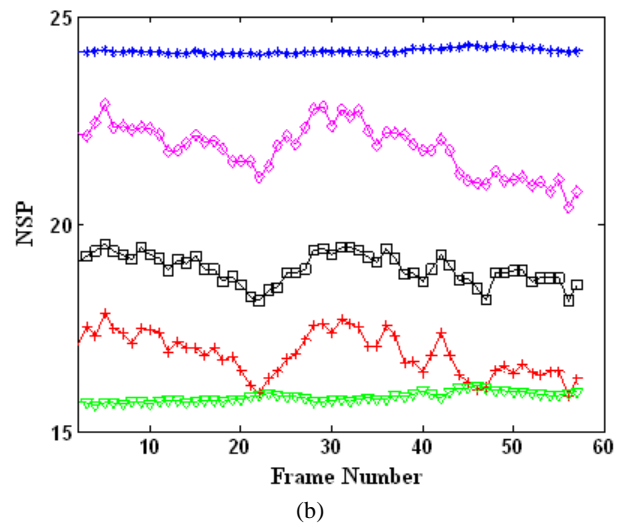
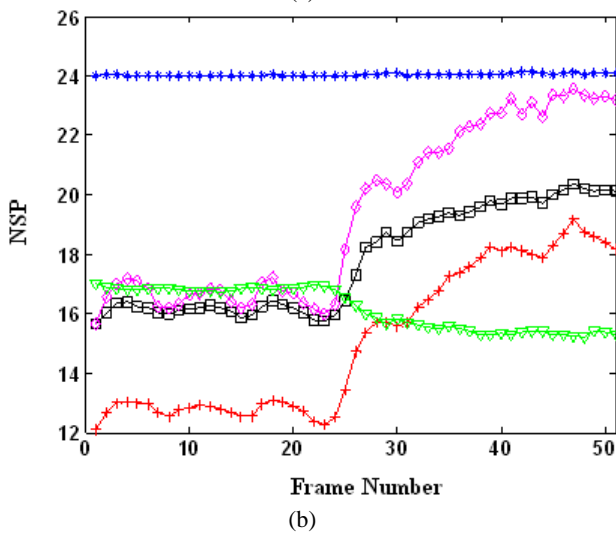
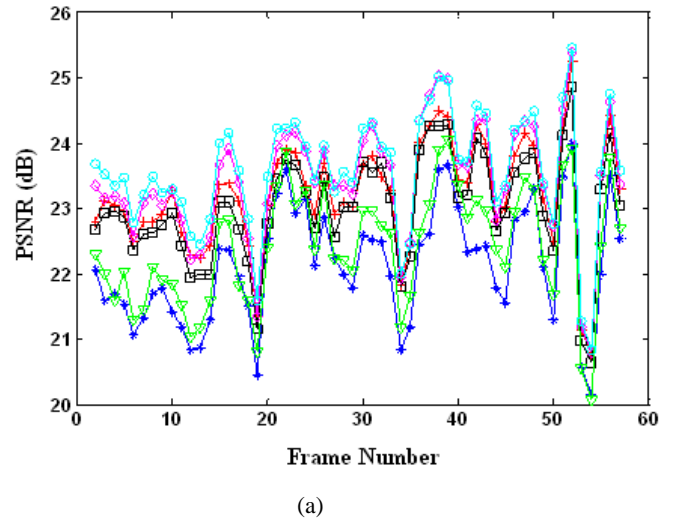
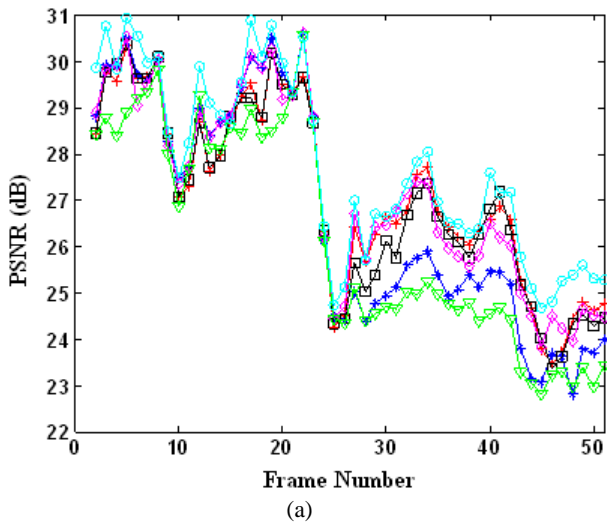


Figure 4: PSNR performance and NSP of various combinations for Tennis sequence.

Figure 5: PSNR performance and NSP of various combinations for Garden sequence.