Design and Development of Scaled Value Criterion for Video Compression using Block Matching Techniques

Awanish Kr. Mishra
Assistant Professor
PSIT, Kanpur

Vinod Kumar, Kailash N Tripathi
Assistant Professor
MIT, Meerut

Amod Tiwari, PhD.
Associate Professor
PSIT, Kanpur

ABSTRACT

Video Compression has played an important role in Multimedia data storage and transmission. Video compression techniques removes spatial as well as temporal redundancy using intra-frame and inter-frame coding respectively. A large level of compression can be achieved through inter-frame coding. In this paper, a new matching criteria is proposed for the video compression using block matching techniques. Along with this, performance of four matching criterion in the temporal coding of video signal, which are Minimum Mean Absolute Error, Vector Matching Criterion, Smooth Constrained - Mean Absolute Error, and proposed algorithm using new pixel values are compared. Three step search algorithm has been used for searching the matching block as a block matching technique, because three step search algorithm is very simple and efficient search algorithm, which provides near optimum results only in three steps. For the various videos it has been experimentally observed that the minimum average error per pixel, and minimum search points per block reduces upto 0 per pixel and 13.31 per block respectively using proposed criteria and the values for the same parameters and same set of frames are very high using other criterion.

Keywords: Motion Estimation, Search Parameter, Motion Compensation, MME, Motion Vector

1. INTRODUCTION

Video data has spatial as well as temporal redundancy and are removed using Video Compression Techniques [1]. Inter-frame predictive coding is used to eliminate the large amount of temporal and spatial redundancy that exists in video sequences and helps in compressing them. In conventional predictive coding the difference between the current frame and the predicted frame (based on the previous frame) is coded and transmitted.

The encoding side estimates the motion in the current frame with respect to a previous frame. A motion compensated image for the current frame is then created that is built of blocks of image from the previous frame. The motion vectors for blocks used for motion estimation are transmitted, as well as the difference of the compensated image with the current frame is also JPEG encoded and sent. The encoded image that is sent is then decoded at the encoder and used as a reference frame for the subsequent frames. The decoder reverses the process and creates a full frame. The whole idea behind motion estimation based video compression is to save on bits by sending JPEG encoded difference images which inherently have less energy and can be highly compressed as compared to sending a full frame that is JPEG encoded. It should be noted that the first frame is always sent full, and so are some other frames that might occur at some regular interval (like every 6th frame). The standards do not specify this and this might change with every video being sent based on the dynamics of the video. The most computationally expensive and resource hungry operation in the entire compression process is motion estimation. Hence, this field has seen the highest activity and research interest in the past two decades. This paper implements and evaluates the fundamental block matching algorithm that is Three Step Search[3] (TSS) using different search criterion.

The better the prediction, the smaller the error and hence the transmission bit rate. If a scene is still, then a good prediction for a particular pixel in the current frame is the same pixel in the previous frame and the error is zero. However, when there is motion in a sequence, then a pixel on the same part of the moving object is a better prediction for the current pixel. The use of the knowledge of the displacement of an object in successive frames is called Motion Compensation. There are a large number of motion compensation algorithms for inter-frame predictive coding. In this study basic focus is only on one class of such algorithms, called the Block Matching Algorithms. These algorithms estimate the amount of motion on a block by block basis, i.e. for each block in the current frame, a block from the previous frame is found, that is said to match this block based on a certain criterion.

One of the first algorithms to be used for block based motion compensation is what is called the Full Search or the Exhaustive Search. In this, each block within a given search window is compared to the current block and the best match is obtained (based on one of the comparison criterion). Although, this algorithm is the best one in terms of the quality of the predicted image and the simplicity of the algorithm, it is very computationally intensive. Some of the efficient block-based search algorithms are Full Search (FS), Three Step Search[3] (TSS), New TSS[4], Two Dimensional Logarithmic Search[2] (TDL), Four Step Search[5] (FSS), Gradient Descent Search[6], Diamond Search[7,8] Hexagon based Search [9,10].

2. BLOCK MATCHING ALGORITHM

The underlying supposition behind motion estimation is that the patterns corresponding to objects and background in a frame of video sequence move within the frame to form corresponding objects on the subsequent frame. The idea behind block matching is to divide the current frame into a matrix of ‘macro blocks’ that are then compared with corresponding block and its adjacent neighbors in the previous frame to create a vector that stipulates the movement of a macro block from one location to another in the previous frame. This movement calculated for all the macro blocks comprising a frame, constitutes the motion estimated in the current frame. The search area for a good macro block match is constrained up to p pixels on all fours sides of the corresponding macro block in previous frame. This 'p' is called as the search parameter. Larger motions require a larger p, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter p is 7 pixels. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block.

3. MATCHING CRITERION

This gives the criteria on basis of that the matching blocks are searched using matching algorithm

3.1. Mean Absolute Error Criterion*:*

The matching criterion mostly used in the literature is minimum mean absolute error, which at point (i, j) for an NxN block and search window of size ±p, is defined as –

$$MAE(i,j) = \frac{1}{N^2}\sum_{x,y}|c(x,y) - r(x+i, y+j)| \qquad (1)$$

where, −p<= i , j<= +p and c(x,y) and r(x,y) are pixel values at position (x,y) in the current and reference frame respectively. Motion vector is defined as the value of (i, j) for which MAE(i, j) is minimum. Obviously, the residue error between the predicted and actual block in the current frame should be minimum for good matching.

3.2 Vector Matching Criterion:

In MAE based criterion, the average error value is considered while ignoring the individual error term.

S. Wang and H. Chen proposed vector matching criterion for block matching to overcome this drawback. In this approach, each NxN block is represented by a vector. Further, each block is subdivided into smaller blocks of size like 2x2, which is represented by a component of the corresponding vector and MAE is calculated between each temporally adjacent subblock in the current and reference frame.

A threshold value is chosen by exhaustive search and vector components (out of N2 / 4 ,assuming the subblock size as 2x2) having value smaller than the threshold value are counted for a given block. Finally, the block having maximum number of such vector components within the defined search area is declared to be the best matching block.

3.3 Smooth Constrained – Mean Absolute Error Criterion:

In video data compression, the residue frame which is calculated by taking the difference of the current and the predicted frame, is coded using transform coding technique, called Discreet Cosine Transform (DCT). According to the characteristic of this transform , the number of bits required to code a smooth residue frame will be smaller than the non smooth residue frame. Therefore, X. Jing, C. Zhu and L. Chau , proposed a smooth constrained based MAE as block matching criterion for motion compensation to reduce the required number of bits for coding besides minimizing the total distortion. In this method, not only the MAE over the residue block is taken into consideration but also the maximum and minimum residue value error, denoted as MME, is taken care of as well. Since DCT is applied over 8x8 block, each residue block (16x16) is divided into four equal size subblocks (8x8) and MME is calculated for each subblock as

$$MME_i = r^i_{max} - r^i_{min} \qquad (2)$$

$$SC - MAE = MAE + \alpha \sum_{i=1}^{4} MME_i \qquad (3)$$

where alpha is a weighing factor. The block which has minimum SC-MAE value in the search area, is declared as the best matched block.

3.4. Proposed Scaled Value Criterion:

Though VMC and SC-MAE based methods have partially reduced the drawbacks of MAE criterion, but they are not suitable for input video data specially with rotation and zoom effect. Further, the similarity measurement of blocks in VMC is dependent on the input threshold value. In this section, a new criterion for block matching is being proposed which not only removes all the shortfalls of MAE but also gives better results than VMC and SC-MAE based techniques.

Let R and C are two frames of equal size (NxN) in the reference frame and current frame respectively. Further, let R = [R_1, R_2, ... R_{N_2}] and C = [C_1, C_2, ... C_{N_2}] be the pixel values in these blocks. Since the image block may have different range of pixel values along each dimension, the pixel values are redefined on the basis of the higher range of intensities in the frame. If the minimum and maximum intensity values in reference block R are R_{min} and R_{max}, and same for the current block are C_{min} and C_{max} , then the new intensity values of the reference block R_{new} and current block C_{new} are defined as given below:

If $(C_{max} - C_{min}) \leq (R_{max} - R_{min})$ then,

$$R_{new} = (R_{old} - R_{min}) \qquad (4)$$

And

$$C_{new} = (C_{old} - C_{min})\, round\{\frac{(R_{max} - R_{min})}{(C_{max} - C_{min})}\} \qquad (5)$$

Otherwise,

$$C_{new} = (C_{old} - C_{min}) \qquad (6)$$

And

$$R_{new} = (R_{old} - R_{min})\, round\{\frac{(C_{max} - C_{min})}{(R_{max} - R_{min})}\} \qquad (7)$$

This gives new rescaled intensity values for all the pixels. The matching function M(R,C) between block R and block C, is defined as -

$$M(R,C) = \frac{1}{N^2}\sum_{k=1}^{N^2} f(|(R_{new}) - (C_{new})|, \tau) \qquad (8)$$

where f(d, $\boldsymbol{\tau}$) is,

$$f(d,\tau) = \begin{cases} d & if\ d \leq \tau \\ max(R_{max}\, , C_{max}\,) & otherwise \end{cases} \qquad (9)$$

The function f(|(R_{new})–(C_{new})|,τ) measures the degree of matching between R_{new} and C_{new} and the positive threshold parameter τ = max(($C_{max} - C_{min}$) , ($R_{max} - R_{min}$)), determines the selection of pixels for matching purposes, i.e. for a value of τ only those pixels will contribute in matching for whom $|(R_{new}) - (C_{new})| \le \tau$. Finally, the location of any such block R in the reference frame in a given search window for which the value of M(R,C) is minimum, gives motion vector.

4. EXPERIMENTAL RESULTS:

In finding the results from the hardware design for the different Block matching criterions using three step search method for the block based search, three sample videos viptraffic.avi, Kamin2.avi and Susie.avi have been used for comparison. Results from all the predefined three criterions along with the proposed one are given in tabular form and also are shown using graphs for the comparison. These experiments have been performed in terms of two parameters-average error per pixel and average search points per block on three videos. When the changes in adjacent frames are nominal, it has been observed that the proposed criterion gives the better results in comparison to that of MAD and SC-MAD. If the degree of variation in intensities of the adjacent frames is high, the proposed algorithm gives the best results, when compared with other three criterions. As for the videos Kamin2 and Susie.avi, variation in the intensities of adjacent frames are very high, experimental results are very much in favor of proposed criterion. Proposed criterion improves the results in the sequence of increasing frames distance i.e. for more frame distance proposed criterion gives better results when compared with mean absolute difference, vector matching criterion and smooth constrained-mean absolute difference. It has been observed experimentally by taking first 30 thirty frames form these mentioned videos.

Detailed results are given in the tables and corresponding graphs.

5. CONCLUSIONS:

A new block matching criterion for motion compensation has been proposed and experimentally examined with three other existing methods in terms of average search points per block, average error per pixel for three videos (viptraffic.avi, kamin2.avi and Susie.avi) inputs with different size and varying degree of motion. The proposed algorithm gives much better results in the case when video quality fades i.e. same pels in

different frames have the different intensity and this difference in intensity for the redundant pels increases by the time.

6. REFERENCES

[1] T.Sikora, MPEG Digital Video Coding Standards, IEEE Signal Processing Magazine, pp 82-100, (September 1997)

[2] J.R.Jain and A.K.Jain, Displacement Measurement and its application in Interframe Coding, IEEE Transactions on Communications, Volume 29, No 12, (December 1981)

[3] T.Koga, K.Iinuma, A.Hirano and Y.Ishiguro, Motion Compensated Interframe Coding for Video Conferencing, Proc NTC81, New Orleans, LA, (November 1981)

[4] R.Li, B.Zeng and M.L.Liou, A New Three Step Search Algorithm for Block Motion Estimation, IEEE Transactions on Circuits and Systems for Video Technology, Volume 4, No 4, (August 1994)

[5] L.Man Po and W.C.Ma, A Novel Four Step Search Algorithm for Fast Block Motion Estimation, IEEE transactions on Circuits and Systems for Video Technology, Volume 6, No 3, (June 1996)

[6] L.Kuo and E. Feig, A Block based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding, IEEE transactions on Circuits and Systems for Video Technology, Volume 6, No 4, (August 1996)

[7] S.Zhu and K.K.Ma, A New Diamond Search Algorithm for Block Matching Motion Estimation, IEEE transactions on Image Processing, Volume 9, No 2, (February 2000)

[8] C.H.Cheung and L.M.Po, A Novel Cross Diamond Search Algorithm for Fast Block Motion Estimation, IEEE transactions on Circuits and Systems for Video Technology, Volume 12, No 12, (December 2002)

[9] C.Zhu, X.Lin and L.P.Chau, Hexagone based Search Pattern for fast Block Motion Estimation, IEEE transactions on Circuits and Systems for Video Technology, Volume 12, No 5, (May 2002)

[10] C.Zhu, X.Lin, L. Chau and L.M.Po, Enhanced Hexagonal Search for Fast Block Motion Estimation, IEEE transactions on Circuits and Systems for Video Technology, Volume 14, No 10, (October 2004)

Table 1 Average error per pixel for viptraffic.avi

Average error per pixel			
MAD	**SC-MAD**	**VMC**	**PROPOSED**
5	3	2	3
5	3	2	3
5	3	2	3
5	3	2	2
5	3	2	2
5	3	2	3
5	3	2	3
5	3	2	3
5	3	2	3
6	4	2	4
6	4	3	4
2	1	0	1
6	5	3	3
6	4	2	4
5	3	2	3
5	3	2	3
5	3	2	3
5	3	2	2
5	4	2	2
15	16	29	23
26	26	40	38
16	16	14	15
2	1	0	1
19	18	16	16
17	16	13	13
15	14	12	11
17	16	32	25
10	10	6	7
14	14	30	23

Table 2 Average search points per block for viptraffic.avi

Average search points per block			
MAD	**SC-MAD**	**VMC**	**PROPOSED**
13.48	13.52	13.39	13.56
13.48	13.48	13.39	13.56
13.48	13.48	13.39	13.56
13.48	13.48	13.39	13.52
13.48	13.48	13.39	13.52
13.48	13.52	13.39	13.52
13.48	13.52	13.39	13.52
13.80	13.73	13.39	13.52
13.80	13.69	13.39	13.63
13.80	13.84	13.39	13.63
13.80	13.73	13.39	13.73
13.44	13.91	13.31	13.44
13.87	13.59	13.35	13.48
13.57	13.52	13.35	13.48
13.48	13.52	13.39	13.56
13.48	13.48	13.39	13.56
13.48	13.48	13.39	13.56
13.48	13.48	13.39	13.52
13.48	13.48	13.39	13.52
14.75	14.85	19.69	17.37
15.63	15.73	20.28	19.56
14.80	14.69	14.17	13.84
13.44	13.44	13.31	13.44
14.87	14.76	13.95	13.97
14.47	14.47	13.64	13.72
14.48	14.37	13.77	13.84
14.55	14.37	19.44	17.51
14.25	14.25	13.41	13.76
15.16	15.16	20.03	17.80

Table 3 Average error per pixel for Kamin2.avi

Average error per pixel			
MAD	SC-MAD	VMC	PROPOSED
27	28	31	28
26	27	35	30
32	28	31	28
16	17	16	16
15	15	16	19
27	25	26	26
14	15	20	16
14	14	17	15
24	23	30	26
13	13	20	14
16	14	36	15
31	33	26	31
27	26	27	22
27	27	16	25
13	12	18	14
13	13	20	16
13	12	20	16
14	14	21	20
16	15	26	19
18	20	19	20
17	17	24	18
16	15	24	17
26	26	38	27
33	30	24	33
23	23	28	26
26	27	29	28
27	27	29	25
29	30	34	34
17	16	22	16

Table 4 Average search points per block for Kamin2.avi

Average search points per block			
MAD	SC-MAD	VMC	PROPOSED
14.45	14.56	14.15	14.71
14.15	14.23	14.89	14.53
14.85	14.85	14.05	14.32
13.75	13.75	13.51	13.81
13.71	13.71	13.79	13.85
14.21	14.00	13.76	14.15
13.64	13.61	13.55	13.75
13.75	13.61	13.68	13.85
14.49	14.35	14.76	14.25
13.64	13.64	13.72	13.81
13.64	13.65	14.87	13.92
14.69	14.69	13.91	14.49
14.39	14.39	14.40	13.81
14.20	14.20	13.65	14.17
13.71	13.64	14.15	13.79
13.68	13.68	13.80	13.89
13.64	13.57	13.76	13.89
13.61	13.61	14.23	14.11
13.87	13.80	14.31	13.87
13.87	13.87	13.65	13.93
13.76	13.76	14.23	13.87
13.68	13.68	13.91	14.29
14.13	14.24	14.71	13.93
15.17	15.17	14.41	14.45
14.15	14.00	14.29	14.07
14.23	14.33	13.80	14.68
14.21	14.19	13.80	13.93
13.96	14.07	14.59	14.21
13.57	13.57	13.79	13.81

Table 5 Average error per pixel for Susie.avi

Average error per pixel			
MAD	SC-MAD	VMC	PROPOSED
0	0	0	0
5	2	2	5
13	18	16	15
13	15	18	13
14	9	8	14
2	1	1	2
5	1	2	5
11	4	6	10
9	3	4	9
3	2	2	4
19	6	11	20
16	10	12	16
11	9	7	12
23	19	34	23
22	21	29	24
0	0	0	0
39	46	38	39
20	13	9	22
31	34	24	37
37	40	38	26
14	7	18	16
4	2	1	4
25	22	22	27
20	8	13	20
18	16	13	18
36	33	28	35
16	11	12	17
17	7	4	18
26	19	25	26

Table 6 Average search points per block for Susie.avi

Average search points per block			
MAD	SC-MAD	VMC	PROPOSED
14.50	14.50	14.43	14.50
14.53	14.52	14.48	14.53
14.54	14.56	14.53	14.52
14.57	14.62	14.62	14.57
14.59	14.55	14.48	14.59
14.50	14.50	14.45	14.50
14.51	14.52	14.45	14.50
14.62	14.53	14.45	14.58
14.55	14.55	14.48	14.55
14.50	14.52	14.48	14.50
14.65	14.53	14.50	14.69
14.62	14.59	14.55	14.62
14.58	14.55	14.48	14.58
14.69	14.69	14.73	14.69
14.76	14.71	14.82	14.84
14.50	14.50	14.43	14.50
14.78	14.79	14.77	14.80
14.64	14.57	14.52	14.69
14.72	14.67	14.65	14.33
14.79	14.76	14.72	14.48
14.65	14.58	14.58	14.65
14.51	14.50	14.43	14.51
14.72	14.65	14.60	14.69
14.65	14.57	14.50	14.65
14.58	14.59	14.53	14.58
14.76	14.71	14.66	14.76
14.58	14.59	14.52	14.58
14.59	14.62	14.49	14.66
14.69	14.62	14.64	14.67

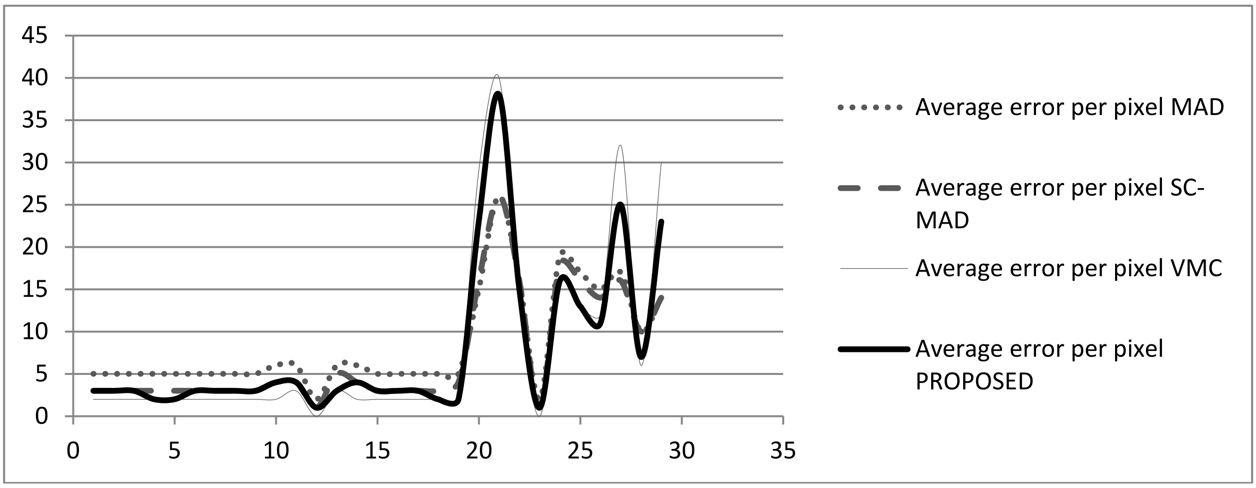

Figure 1 Comparison of Average error per pixel for viptraffic.avi

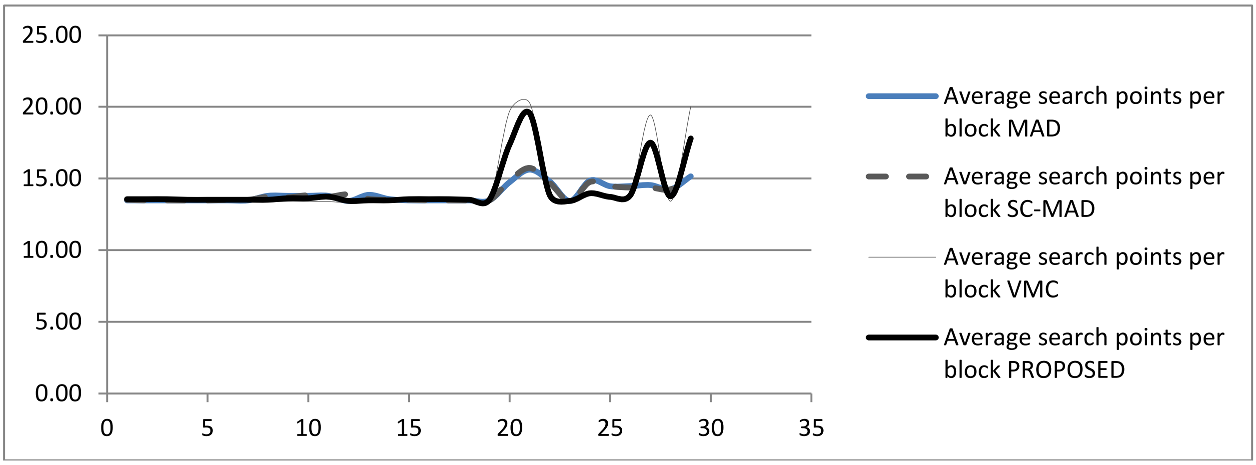

Figure 2 Comparison of Average search points per block for viptraffic.avi

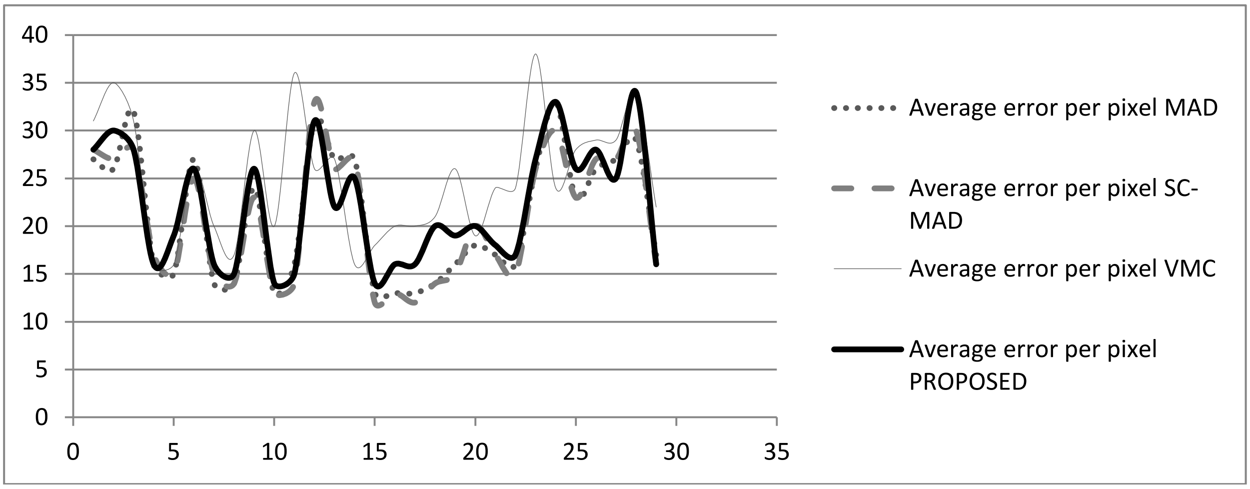

Figure 3 Comparison of Average error per pixel Kamin2.avi

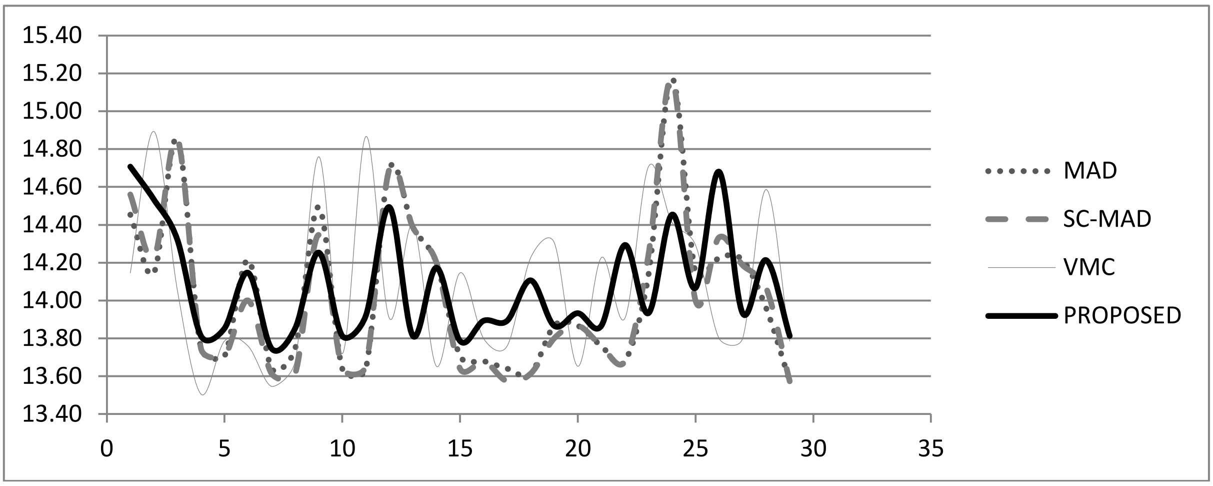

Figure 4 Comparison of Average search points per block for Kamin2.avi

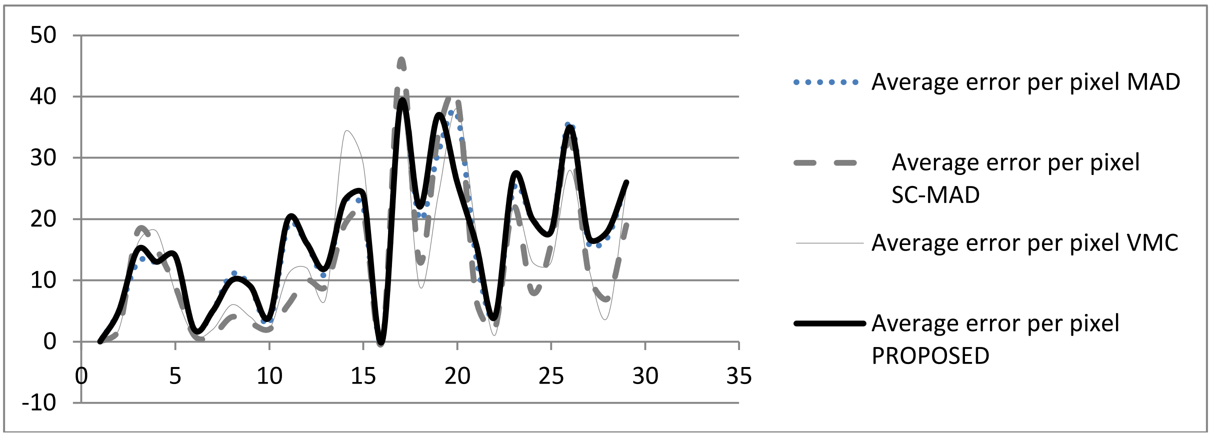

Figure 5 Comparison of Average error per pixel for Susie.avi

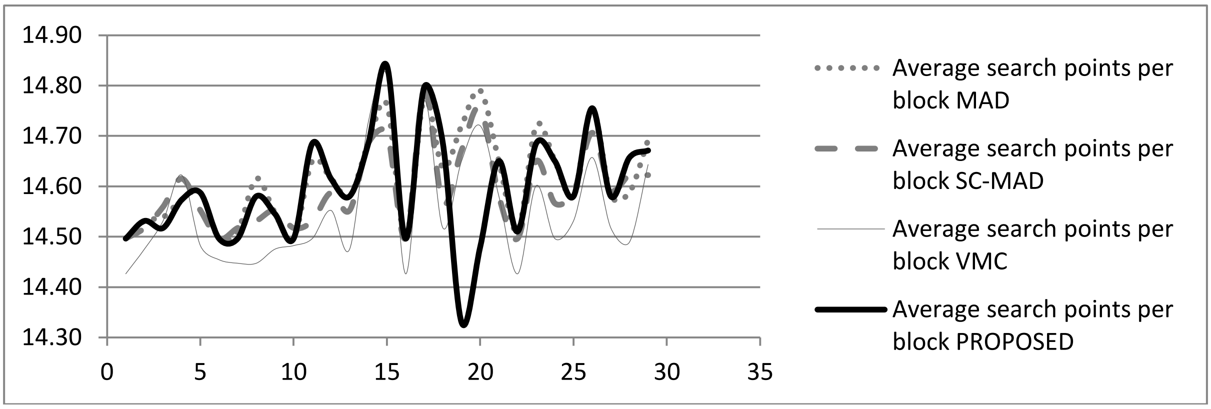

Figure 6 Comparison of Average search points per block for Susie.avi