CHAPTER 3

BLOCK MATCHING CRITERION

Summary

In this chapter, a similarity measure for block matching has been proposed which is robust to contrast variations in input video sequences as compared to conventional mean absolute error (MAE) criterion. The algorithm maps pixel intensities of video frames using minimum and maximum block intensities in the reference and current frames and matching is performed over these mapped pixel intensities. To increase the speed and reduce the possibility spurious block matching, the proposed method is further modified which requires only integer arithmetic. A modified block matching criterion using integral frame to minimize the possibility of false block matching in conventional block sum based integral frame method has also been proposed.

3.1 Introduction

A video frame may contain various independently moving objects. Therefore, it is better to first divide a frame into small sized macroblocks (also has been referred as block interchangeably in this thesis) to study the motion of an individual object more precisely. The basic idea behind a block based motion estimation algorithm (BMEA) is to divide a frame into different macroblocks and for each macroblock in the current frame, a matching macroblock in the reference frame is searched in a given search window. The direction of the reference frame macroblock with respect to the current frame macroblock which has maximum matching, is declared as the motion vector for the corresponding macroblock in the current frame. A number of similarity measure criterions for macroblock matching have been proposed in literature Chan et al. (1994), Kappagantula and Rao (1985), T.Koga et al. (1981), Chen et al. (1995), Kim and Park (1992). MAE based technique proposed by T.Koga et al. (1981) is extensively used
because of its low cost and simplicity. MAE is defined at point \((i, j)\) for a macroblock of size \(N \times N\) and search window size of \(\pm p\) as

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

(3.1)

where, \(-p \leq i, j \leq +p\) and \(c(x, y)\) and \(r(x, y)\) are pixel values at position \((x, y)\) in the current and reference frame respectively.

As shown in eq. 3.1, in this criterion, corresponding pixels from each macroblock are compared and their absolute difference is summed up. Motion vector is defined as the value of \((i, j)\) for which \(MAE(i, j)\) is minimum.

Wang and Chen (1999) and Jing et al. (2003), proposed variants of conventional MAE called vector matching criterion (VMC) and smooth constrained MAE (SCMAE) which have been discussed in section 3.2 and section 3.3 respectively of this chapter. In section 3.4, normalized pixel mapping based block matching criterion has been proposed (R.K.Purwar et al. (2008), R.K.Purwar et al. (2009b), R.K.Purwar et al. (2009a)) which is followed by the proposed multilevel block matching criterion in section 3.5. In the last, section 3.6 concludes this chapter.

### 3.2 Vector Matching Criterion (VMC)

Although conventional MAE is simple and easy for hardware implementation, but accuracy of block matching is compromised as it covers the average error over the entire block ignoring the error at individual pixel. It may not produce good results if the assumption of uniform motion of macroblocks in BMEAs becomes invalid. Wang and Chen (1999) modified conventional MAE and suggested a new distortion measure using vector matching concept which works at the fine level in a macroblock and improves accuracy of matching.

In this approach, as shown in flow chart of fig. 3.1, each \(N \times N\) macroblock is represented by a vector. Further, each macroblock is subdivided into \(\frac{N^2}{4}\) smaller sub blocks of size \(2 \times 2\) each. Each macroblock in the current frame is searched in the reference
Divide each macro-block of size 16x16 into 64 sub-blocks of size 2x2

Represent each macro-block using a vector with 64 components

Current macro-block

Candidate macro-block

Compute MAE at sub-block levels and count no. of corresponding vector components having value lesser than the threshold

Candidate macro-block with maximum count of vector components is selected as the best match block

Figure 3.1: Flow Chart of VMC
frame by examining values of sum of absolute differences of their corresponding $\frac{N^2}{4}$ subblocks with respect to a threshold. This threshold is chosen using exhaustive search technique for motion estimation based on conventional MAE as similarity measure. For each current frame macroblock, its best match reference frame macroblock is found and the value of MAE between these macroblocks is stored. This procedure is repeated for all current frame macroblocks. Finally, average value of these MAEs is used as threshold of all macroblocks of the current frame in VMC. Similarity of macroblocks is decided by counting the number of corresponding vector components with sum of absolute difference less than threshold. Finally, the macroblock having maximum number of such count within the defined search area is declared to be the best matching block.

3.3 Smooth Constrained Mean Absolute Error (SC–MAE)

As discussed in chapter 1, video encoder uses interframe coding for P frames where they are differentially coded i.e. residue frame is constructed by taking absolute difference between the desired frame and motion compensated frame and this residue frame is coded using intraframe coding. In intraframe coding, number of bits assigned to a smoother residue block are smaller than to a non smoother block. Conventional MAE considers only the average pixel error as distortion measure and it does not give any weightage to the overall smoothness of a residue block. As a result, BMEAs using conventional MAE as similarity measure criterion may fail to produce efficient bits/pixel. Therefore, it is desired that match between the candidate block and the current block requires not only to minimize the average distortion but also it should make the residue block as smooth as possible.

Jing et al. (2003) proposed a novel block matching criterion, called smooth constrained mean absolute error (SC–MAE), which aims to eliminate this drawback of conventional MAE. In the proposed algorithm, as shown in the flow chart of fig. 3.2, a residue block $R$ (16x16) is formed by taking the absolute difference between candidate block and current block and conventional MAE is computed. After that, $R$ is subdivided into four small blocks ($R_1$ to $R_4$) with size 8x8 and the difference between maximum and minimum residue, denoted by $MMR_m$, is calculated for each sub block $R_m$ using
eq. 3.2. This procedure is depicted in fig. 3.3.

\[
MME_m = r_m^{max} - r_m^{min}
\]  

(3.2)

where \( r_m^{max} \) and \( r_m^{min} \) are the maximum and the minimum residue values within the residue block \( R_m \), respectively. The reason for dividing the 16x16 residue block into four 8x8 sub blocks is that motion estimation algorithms work on 16x16 size whereas DCT used in intraframe coding works on 8x8 size. Therefore, it is better to consider smoothness at the sub block levels separately and take the sum of their MMEs as the overall smoothness for the resulting residue block.

Finally, proposed similarity measure (SC–MAE) is computed as:

\[
SC-MAE = MAE + \alpha \sum_{i=1}^{4} MME_i
\]  

(3.3)

where \( \alpha \) is the weighing factor and \( MAE \) is computed using eq.3.1. Macroblock which has smallest SC–MAE value is declared as the best match block.

The steps used in SC–MAE are summarized below –

1. Subtract the candidate macroblock from the current macroblock and form the residue macroblock.
2. Compute conventional MAE for the residue macroblock.
3. Divide the residue macroblock into four smaller sub blocks and determine \( MME \) for each such sub block using eq. 3.2.
4. Compute the final distortion measure SC–MAE using eq. 3.3.
5. Select the candidate block which has minimum SC–MAE value.
Figure 3.2: Flow Chart of SC-MAE Criterion
3.4 Normalized Pixel Mapping based Block Matching Criterion

VMC (Wang and Chen (1999)) and SC–MAE (Jing et al. (2003)) have reduced the drawbacks of conventional MAE criterion by considering individual error term and smoothness of the residue block respectively but these distortion measures are not sufficient enough to take care of the contrast variations in input video sequences. Further, a single value of threshold in VMC and weighting factor $\alpha$ in SC–MAE does not give good results for all type of video inputs. In this section, a new block matching criterion is being proposed which takes care of contrast variations in input videos. In this technique, both current frame block and reference frame block are normalized before matching. Two types of normalization criterions have been proposed – one is using block based floating point arithmetic (R.K.Purwar et al. (2008) and R.K.Purwar et al. (2009b)) and other is frame based integer arithmetic (R.K.Purwar et al. (2009a)) as discussed below.
3.4.1 Block based Normalization using Floating Point Arithmetic

Normalization criterion uses the minimum and maximum pixel values of the block (current as well as reference) as suggested by Tseng et al. (1996). Detailed algorithm is described below.

Let \( R \) and \( C \) are the reference and the current frame blocks of size \( N \times N \) respectively. Further, let \( R = [R_1, R_2, \ldots, R_{N^2}] \) and \( C = [C_1, C_2, \ldots, C_{N^2}] \) be the pixel intensities in these blocks. The matching function \( M(R, C) \), with \( 0 \leq M(R, C) \leq 1 \), is used to define the degree of similarity between the current block \( C \) and the reference block \( R \). Assume \( R_{\text{low}} \) and \( R_{\text{up}} \) are the lower and upper bounds of all \( R_k \)'s, pixel intensities in the reference frame block \( R \) are scaled as

\[
S(R_k) = \frac{R_k - R_{\text{low}}}{(R_{\text{up}} - R_{\text{low}})} \tag{3.4}
\]

Similarly scaled pixel values of current frame block are defined as

\[
S(C_k) = \frac{C_k - C_{\text{low}}}{(C_{\text{up}} - C_{\text{low}})} \tag{3.5}
\]

where \( C_{\text{low}} \) and \( C_{\text{up}} \) are lower and upper bounds of all \( C_k \)'s in the current frame block.

In denominators of these equations, 1 may be added to avoid the divide by zero error. The matching function \( M(R, C) \), between the reference block \( R \) and the current block \( C \) of size \( (N \times N) \), is defined as

\[
M(R, C) = \frac{1}{N^2} \sum_{k=1}^{N^2} f(|S(R_k) - S(C_k)|, \tau) \tag{3.6}
\]

where \( f(d, \tau) \) is

\[
f(d, \tau) = \begin{cases} 
1 - d\tau & \text{if } d\tau \leq 1 \\
0 & \text{otherwise}
\end{cases} \tag{3.7}
\]

The function \( f(|S(R_k) - S(C_k)|, \tau) \) measures the degree of matching between \( S(R_k) \) and \( S(C_k) \) and the positive parameter \( \tau \) is a threshold for selection of pixels.
for matching purposes. For a given value of $\tau$, only those pixels will contribute in matching for whom $d\tau \leq 1$ or $d \leq \frac{1}{\tau}$.

Finally, the location of the reference frame block for which the value of $M(R, C)$ is maximum in the search area, gives motion vector.

### 3.4.2 Frame based Normalization using Integer Arithmetic

Proposed floating point arithmetic based block matching criterion gives better results than conventional MAE and its variants – VMC and SC–MAE but it has two major drawbacks—

1. Since the matching criterion is heavily using floating point operations, it is not practical for real time applications.

2. As can be seen from eq. 3.4 and eq. 3.5, pixels within macroblocks are scaled using minimum and maximum pixel intensities locally within the block, there is possibility of spurious block matching.

Keeping these points in mind, previous criterion has been modified and a new one using integer arithmetic has been suggested to reduce computation time (R.K.Purwar et al. (2009a)). Further, it does not suffer from drawback of spurious block matching possibility. In this new approach, pixel intensities are mapped using global minimum and maximum intensities within the frame. Proposed frame based normalization using integer arithmetic is explained below as well as summarized in the flow chart, shown in fig. 3.4.

Let $R_{\text{min}}$, $R_{\text{max}}$ and $C_{\text{min}}$, $C_{\text{max}}$ are minimal and maximal pixel values in the reference and current frame respectively. Further, let $R'_k$s and $C'_k$s are pixel values in these frames respectively. If $(C_{\text{max}} - C_{\text{min}}) \leq (R_{\text{max}} - R_{\text{min}})$, then $R_k$ and $C_k$ are scaled as

$$R_{k,\text{new}} = R_k - R_{\text{min}}$$  \hspace{1cm} (3.8)
Find $R_{\text{min}}, R_{\text{max}}, C_{\text{min}}$ and $C_{\text{max}}$ in the reference frame and current frame respectively.

Is $(C_{\text{max}} - C_{\text{min}})$ is less than or equal to $(R_{\text{max}} - R_{\text{min}})$?

Yes:

\[
\begin{align*}
R_{k\text{new}} &= R_k - R_{\text{min}} \\
C_{k\text{new}} &= (C_k - C_{\text{min}})(R_{\text{max}} - R_{\text{min}}) \\
&\quad \div (C_{\text{max}} - C_{\text{min}}) \\
&\quad \text{(rounded to nearest integer)}
\end{align*}
\]

No:

\[
\begin{align*}
C_{k\text{new}} &= C_k - C_{\text{min}} \\
R_{k\text{new}} &= (R_k - R_{\text{min}})(C_{\text{max}} - C_{\text{min}}) \\
&\quad \div (R_{\text{max}} - R_{\text{min}}) \\
&\quad \text{(rounded to nearest integer)}
\end{align*}
\]

Current frame macroblock

Reference frame macroblock

\[
\text{Computer} \ M(R, C)
\]

Select block with minimum $M(R, C)$

Figure 3.4: Flow Chart of Proposed Integer Arithmetic based Criterion
\[ C_{k\text{ (new)}} = \frac{(C_k - C_{\text{min}})(R_{\text{max}} - R_{\text{min}})}{(C'_{\text{max}} - C'_{\text{min}})} \text{ (rounded to the nearest integer)} \quad (3.9) \]

otherwise

\[ C_{k\text{ (new)}} = C_k - C_{\text{min}} \quad (3.10) \]

\[ R_{k\text{ (new)}} = \frac{(R_k - R_{\text{min}})(C'_{\text{max}} - C'_{\text{min}})}{(R'_{\text{max}} - R'_{\text{min}})} \text{ (rounded to the nearest integer)} \quad (3.11) \]

so that both frames have same minimum and maximum pixel intensity values. In denominators of eq. 3.9 and eq. 3.11, 1 may be added to avoid the divide by zero error. The matching function \( M(R,C) \), between the reference block \( R \) and the current block \( C \) of size \( (N \times N) \) after scaling, is defined as -

\[ M(R,C) = \frac{1}{N^2} \sum_{k=1}^{N^2} f(|R_{k\text{ (new)}} - C_{k\text{ (new)}}|, \tau) \quad (3.12) \]

where \( f(d, \tau) \) is -

\[ f(d, \tau) = \begin{cases} 
  d & \text{if } d \leq \tau \\
  \max(R_{\text{max}}, C_{\text{max}}) & \text{otherwise}
\end{cases} \quad (3.13) \]

The function \( f(|R_{k\text{ (new)}} - C_{k\text{ (new)}}|, \tau) \) measures the degree of matching between \( R_{k\text{ (new)}} \) and \( C_{k\text{ (new)}} \) and the positive parameter \( \tau \) is a threshold for selection of pixels for matching purposes. Value of \( \tau \) is bounded by \([0, \max(R_{\text{max}} - R_{\text{min}}, C_{\text{max}} - C_{\text{min}})]\). 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. Proposed criterion gives better results than existing methods. Required justification is given as follows.

Let us assume an 8 bit input sequence for simplicity so the minimum and maximum intensity values in all frames will be lower and upper bounded by 0 and 255 respectively. For a particular block matching, let \( C_{\text{min}} = 0, C_{\text{max}} = 255 \) and \( R_{\text{min}} = 5, R_{\text{max}} = 230 \)
for the current and reference frame before scaling, as intensity may change from frame to frame. Consider a pixel with intensity 100 in the current frame block. This pixel will have best match with a pixel in the reference frame block whose intensity is 100 (assuming this value is present in the search window). Using MAE or other matching criterion, this current block pixel with intensity 100 will have closest match with pixel value 100 in the reference frame though that particular pixel is not meant to be the actual corresponding pixel. In proposed criterion, after scaling, current block pixel will have intensity 100 using eq. 3.10 and it will best match with the scaled reference block pixel intensity 100 which is 93 before scaling (see eq. 3.11), the actual corresponding pixel in the reference frame. Similarly, there is better matching for other block pixels in the current and reference frames which can be proved for generalized values of $C_{\text{min}}, C_{\text{max}}, R_{\text{min}}$ and $R_{\text{max}}$ also.

In chapter 5, a number of experimental results have been performed using various parameters like – average search points/block, average PSNR (dB), average bits/pixel value, execution time (sec.), $\frac{\text{quality}}{\text{compression}}$ ratio etc. to compare the performance of the proposed criterion with conventional MAE and its variants – VMC and SC–MAE and we have achieved better results for the proposed criterion.

### 3.5 Multilevel Block Matching Criterion using Integral Frames

In block matching criterions discussed in section 3.2 to 3.4, each pixel of the current block is processed to find the best match candidate block. For example, in conventional MAE, absolute difference of each pixel in the current block is taken with the corresponding pixel in the candidate block and it is summed to produce MAE for similarity measure. For an NxN block, it requires $N^2$ subtraction operations, $N^2$ absolute differences and finally $N^2 - 1$ addition operations. To overcome this problem, in this section, a multilevel block matching criterion using integral frame has been proposed. In subsection 3.5.1, concept of integral frame has been explained. Integral frame based block matching criterions suggested by Nguyen and Tan (2004) and Nguyen and Tan (2006)
is explained in subsection 3.5.2. A new technique based on multilevel block matching is proposed in subsection 3.5.3.

### 3.5.1 Integral Frame

For a given video frame \( f \), its integral frame feature (Viola and Jones (2001)) at point \((i, j)\), denoted by \( I_f(i, j) \), is defined as the sum of all pixel values that are above and to the left of pixel \((i, j)\) including itself, i.e.

\[
I_f(i, j) = \sum_{x=0}^{i} \sum_{y=0}^{j} f(x, y)
\]  

(3.14)

Let \( R_f(i, j) \) denotes cumulative row sum of pixel values in frame \( f \), defined as

\[
R_f(i, j) = \sum_{x=0}^{i} f(x, j)
\]  

(3.15)

Assuming \( R_f(-1, j) = 0 \) and \( I_f(i, -1) = 0 \), integral frame \( I_f \) can be computed by using recursive formulas

\[
R_f(i, j) = R_f(i-1, j) + f(i, j)
\]

\[
I_f(i, j) = I_f(i, j-1) + R_f(i, j)
\]  

(3.16)

It is clear from eq. 3.16 that for frame of size \( M \times N \), its integral frame feature can be computed using only 2MN additions.

With the help of integral frame feature, sum of all pixel values in any block of frame \( f \), (referred as block sum \((BS_f)\)) can be computed using only 1 addition and 2 subtraction operations. It can be seen from fig. 3.5 that the block sum of block \( S \) in frame \( f \) can be computed as
Figure 3.5: Computation of block sum for block S using four integral frame values

\[ BS_f(S) = \sum_{x=r+1}^{i} \sum_{y=s+1}^{j} f(x, y) \]
\[ = I_f(i, j) - I_f(r, j) - I_f(i, s) + I_f(r, s) \]  

(3.17)

using four corresponding values of the integral frame.

### 3.5.2 Integral Frame based Block Matching Criteria

Nguyen and Tan (2004), suggested motion estimation algorithm using integral frames to minimize the number of computations. For matching, each macroblock is partitioned in a number of sub blocks and block sum of each sub block is then computed using eq.3.17. To find the best matching block in the reference frame, block sums of all sub blocks in the current block are compared with those in each candidate block in the reference frame. Specifically, the sum of absolute differences between the corresponding
block sums (referred as SAD\_BS) is used for block matching, which is given by –

\[
SAD\_BS = \sum_i |BS_{fc}(S_i) - BS_{fr}(S_i)|
\] (3.18)

where \(BS_{fc}(S_i)\) and \(BS_{fr}(S_i)\) denote the block sum of \(i^{th}\) sub block from the current and reference block respectively.

It can be seen from eq. 3.18 that the performance of SAD\_BS based block matching algorithm depends on how the block is partitioned into sub blocks. Partitioning a block into too many sub blocks will increase accuracy as well as computational cost while too few sub blocks will compromise accuracy. Authors have carried out a number of simulations and finally opted for 4 symmetrical sub block division as a tradeoff.

Nguyen and Tan (2006), extended their previous work and proposed another motion estimation algorithm using integral frames to increase quality of reproduced video. In this algorithm, authors used sum of absolute difference of block variance (referred as SAD\_VR) to find the best match block.

Like block sum, squared block sum of block \(S\), denoted as \(BS_{f2}(S)\), shown in fig. 3.5, can be obtained from integral frame as –

\[
I_{f2}(i, j) = \sum_{x=0}^{i} \sum_{y=0}^{j} f^2(x, y)
\]

\[
BS_{f2}(S) = \sum_{x=r+1}^{\text{frame}} \sum_{y=s+1}^{\text{frame}} f^2(x, y)
= I_{f2}(i, j) - I_{f2}(r, j) - I_{f2}(i, s) + I_{f2}(r, s)
\] (3.19)

Using block sum and block square sum features, authors defined variance of all pixel in block \(S\) of size \(N\times N\) as –

\[
\delta^2(S) = \frac{1}{N^2} BS_{f2}(S) - (\frac{1}{N^2} BS_f(S))^2
\] (3.20)

The sum of absolute differences of block variances between the current block and
candidate block is given as –

$$SAD_{VR} = \sum_{i} |\delta_C^2(S_i) - \delta_R^2(S_i)|$$  \hspace{1cm} (3.21)

where $\delta_C(S_i)$ and $\delta_R(S_i)$ denote the variances of $i^{th}$ sub blocks from the current block and the candidate block respectively. Further, authors have also explored the results after combining both features SAD_BS and SAD_VR to find matching block.

### 3.5.3 Multilevel Block Matching Criterion using Integral Frames

Block sum (Nguyen and Tan (2004)) and block variance (Nguyen and Tan (2006)) based matching algorithms using integral frame drastically reduces computational cost with respect to conventional SAD but it has possibility of being trapped in spurious block matching as different blocks may have same representation using integral frame features. To minimize this drawback, a multilevel matching criterion has been proposed which uses multiple minimum points obtained using SAD_BS and SAD_VR algorithms to further explore them with the help of conventional MAE at each iteration of used BMEA to find out the final minimum point for next iteration of BMEA. Since computation cost of SAD_VR is more than that of SAD_BS, proposed criterion has been compared with later one for simplicity.

As shown in the flowchart of fig.3.6, proposed matching criterion has multiple levels depending on how many minimum points (obtained using SAD_BS criterion) are processed at each iteration of BMEA. It is described below.

1\textsuperscript{st} level refinement –

Let locations of 1\textsuperscript{st} and 2\textsuperscript{nd} minimum points obtained using eq. (3.18) are $(i_1, j_1)$ and $(i_2, j_2)$ respectively at a particular iteration of BMEA. These points are further explored using conventional MAE criterion given in eq. 3.1 to find $MAE(i_1, j_1)$ and $MAE(i_2, j_2)$ respectively.

If $MAE(i_1, j_1) \leq MAE(i_2, j_2)$, then $(i_1, j_1)$ otherwise it is $(i_2, j_2)$ that works as final minimum point for that iteration of BMEA. This process is repeated for each iteration of used BMEA until motion vector of each block is found.
Find integral frame representation of input video

Current frame macroblock

Select a candidate reference frame macroblock within the search window for current iteration of used BMEA

Compute SAD_BS and store distortion value

All candidate blocks in the reference frame for current iteration of BMEA have been processed?

No

Select k minimum distortion points in the order of their values and explore these points with the help of conventional MAE to find final minimum distortion point to be used for next iteration of BMEA

Yes

Last iteration of BMEA for current frame macroblock?

No

Location of minimum point represents motion vector for the current frame macroblock

Yes

Figure 3.6: Flow Chart of Proposed Multilevel Block Matching Criterion using k levels
2nd level refinement –

Unlike 1st level matching, in this level three minimum points – 1st, 2nd and 3rd obtained using eq. 3.18 are used at each iteration of BMEA. These points are explored further using conventional MAE (eq. 3.1) to find final minimum point for the next iteration of BMEA.

Similarly higher level refinements may be opted for better quality with increased computation cost. In this work, matching up to 2nd level has been done and quality results as comparable to conventional SAD has been obtained in terms of PSNR (dB) in chapter 5. Further, detailed comparison of proposed criterion with conventional SAD and SAD_BS in terms of other parameters like – execution time (sec.), \( \frac{\text{quality}}{\text{computation}} \) ratio have also been performed and very favorable results have been obtained for the proposed technique.

3.6 Conclusion

This chapter is focussed around block matching criterions used in fast BMEAs for finding motion vectors. Two block matching criterions have been proposed – first uses normalized pixel intensities for block matching whereas second is an enhancement of block sum based sum of absolute difference criterion over integral frames. Experimental results showing significant improvements using proposed techniques over conventional/existing techniques are shown in chapter 5.