Multimed Tools Appl (2007) 34:221–237
DOI 10.1007/s11042-006-0091-6

An improved variable-size block-matching algorithm

Haifeng Wang · Qingshan Liu · Hanqing Lu

Published online: 27 January 2007

Abstract In this paper, we proposed an improved “bottom–up” variable-size block matching method. Different from previous work, the proposed method does not need any threshold during the matching, and we just keep all the motion vectors leading to the minimum matching error. A Marco-block mode prediction method is put forward to speed up the motion estimation procedure without introducing any loss to the prediction precision. The improved variable-size block matching algorithm can achieve exactly the same prediction precision as full-search based fixed-size block matching algorithm. In order to reduce the effect of illumination change on mode selection, we proposed an illumination removal method, which acts as a post-processing step to prevent the macro-blocks from over-splitting. Experiments show its encouraging performance.

Keywords Variable-size block matching · Threshold · Illumination removal · Macro-mode prediction · Motion estimation

1 Introduction

A key issue of video compression is to remove the redundancy between consecutive frames as much as possible to achieve a high compression rate. The basic idea is to predict the current frame from one or more previous frames, and the difference between the current frame and the predicted version together with the displacement vector are coded for subsequent storage or transmission [11]. Motion Estimation (ME) is the process to get the prediction information, and Motion Compensation (MC) process uses this prediction

H. Wang (✉) · Q. Liu · H. Lu
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
e-mail: hfwang@nlpr.ia.ac.cn

Q. Liu
e-mail: qsliu@nlpr.ia.ac.cn

H. Lu
e-mail: luhq@nlpr.ia.ac.cn

information and frame difference to reconstruct the image. Thus, ME and MC are two critical components for video compression. They are also the most time consuming parts in video coding systems, especially when multi prediction modes, multi reference frames and higher motion vector resolution are adopted. For example, in H.264, the ME component can consume up to 60% of the total encoding time with one reference frame, and up to 80% with five reference frames [2]. Block-based motion estimation and compensation are the most popular approaches [14] and are adopted in most standards, such as ISO MPEG series [5–7] and the ITU-T H.26× series [3, 17, 18].

1.1 FSBM vs. VSBM

In the traditional fixed-size Block Matching (FSBM) algorithm, the size of block is predefined. The ME process is only to find a block with the minimum matching error from the reference frame(s). Often the sizes of block are set as 16×16, 8×8, or 4×4. As for a large size block, such as 16×16, there may be different motions in the same block, which will increases the prediction error. This phenomenon is so-called "majority effect." Figure 1a and b show an example. A person and a car move toward opposite directions. If the region marked with dashed square is motion estimated as a 16×16 block, whatever the matching block we get, the prediction error is relatively large. If the block size is 8×16, its MV can easily point to the best match [22]. As we motioned before, there are two kinds of information need to code. One is the prediction error (or frame difference, residue), and the other is motion information. To achieve a high compression ratio, a trade-off must be made between these two kinds of information. The displacement vectors got by block matching algorithm only mean that the MSE (Mean Squared Error) between the current block and the corresponding block in the reference frame is minimum, which has nothing to do with the semantic meaning. This makes it very difficult to capture the structured moving objects of the scene. Several approaches have been proposed to overcome these shortcomings. Among them, variable-size block matching (VSBM) based motion estimation and compensation may improve the performance of the traditional FSBM method, because to a certain extent, it can approximate the distribution of the variable-size blocks to the different motion parts of the scene [4]. In Fig. 1d, we split the block in a into two 8×16 blocks, and

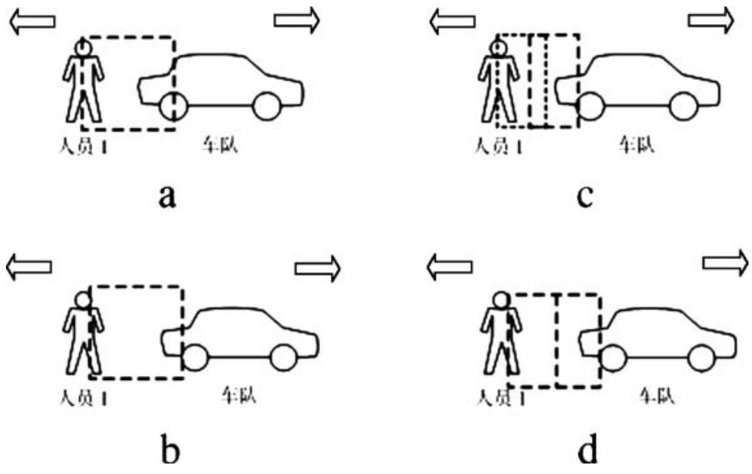


Fig. 1 FSBM vs. VSBM **a** reference frame with FSBM **b** current frame with FSBM **c** reference frame with VSBM **d** current frame with VSBM

do ME for each block, respectively, then a better matching result can be obtained, as shown in Fig. 1c.

The VSBM algorithm is first adopted in H.263 [8], in which smaller blocks are used to describe detailed or complex motion, while larger blocks are used for regions with stationary or undergoing uniform motion [12]. Figure 2 gives a comparison of the block distribution using FSBM and VSBM. It can be seen that the FSBM algorithm uses one size block for all regions (Fig. 2a), while the VSBM algorithm uses large-size blocks to represent the stationary regions and small-size blocks to represent regions with detailed motion (Fig. 2b). In this paper, the sizes of blocks used in our experiments are the seven block modes adopted in H.264, as shown in Fig. 3.

In H.264, the exhaustive-search method is performed for ME with all the block-sizes defined in the standard, and the one that minimizes the cost function is chosen [16]. Though it can find the minimum matching error, this exhaustive method slows down the encoding procedure.

1.2 Fast VSBM algorithms

In order to reduce the complexity of VSBM, Rhee et al. [12] proposed a "bottom–up" VSBM algorithm, in which a full-search is first done on the minimum blocks. If the SAD of a block is smaller than a threshold, its corresponding motion vector (MV) is reserved in a candidate MV set. That the neighbor blocks are merged or not depends on whether they have the same MVs in their candidate MV sets. So the threshold is a key parameter to this method. It determines which vectors should be included in the initial sets, which further determines the prediction precision. In [12], the threshold is calculated by using an iterative procedure on a frame by frame base, which increases the computational complexity. Moreover, in real video sequences, the motion details of different parts in a frame are different, so if we use the same threshold calculated this way to determine whether the MVs of all blocks should be included in the initial sets or not, the prediction precision will be reduced because of the "majority effect." Tu et al. [16] presented to calculate the threshold adaptively according to the information of the motion estimation, quantization parameter and rate distortion cost, but this method is also frame-based and its ME process is complicated too.

Chan et al. [1] proposed a "top–down" approach, in which large blocks are matched first. If the SAD (Sum of Absolute Difference) of the large blocks is higher than a

a b

Fig. 2 A comparison of FSBM and VSBM **a** block distribution of FSBM algorithm; **b** block distribution of VSBM algorithm

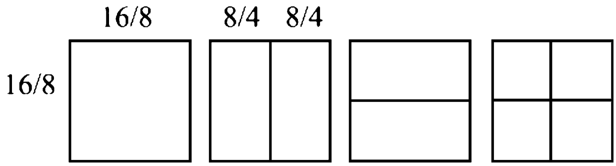


Fig. 3 Block sizes adopted in H.264

predefined threshold, these blocks are split into two or four small blocks. This process is repeated until the maximum number of blocks, or minimum errors are obtained. Finally, a remerging process is conducted to remerge those blocks that cannot reduce SAD or improve image quality. In this method, ME is re-done for sub-blocks when a larger block was split into sub-blocks, which makes the search procedure more complicated, even though some predict methods can accelerate the procedure.

Zhou et al. [22] proposed fast variable block-size motion estimation algorithms based on merge and split procedures. ME is first conducted on 8×8 size blocks and MVs of these initial blocks are used to decide whether to undergo a merge procedure or a split procedure. As shown in Fig. 4, if the neighbor 8×8 blocks have common MVs, then a merge procedure is conducted and these neighbor blocks are merged into 16×8, 8×16 or 16×16 size blocks; if not, 8×8 blocks will be split into 4×8, 8×4 or 4×4 blocks. In both cases, the MVs have to be refined to find a better match. In [22], ADSS (Adaptive Diversity Search Strategy) [20] is used to get MVs of 8×8 blocks. These MVs are used for prediction in the merge and split process and the overall results are highly dependent on the accuracy of these MVs. The final MVs for merged or split blocks are the results of a Small Diamond Search (SDS) plus the prediction as the start point, that is:

$$\mathrm{MV} = \mathrm{MV}_p + \mathrm{MV}_{\mathrm{sds}},$$

where MV is the final MV for the merged or split blocks, MV_p is the prediction MV, which is used as the start search point of the SDS pattern and $\mathrm{MV}_{\mathrm{sds}}$ is the result of SDS.

There is a problem about this Split and Merge method. If we choose small-size blocks to do motion estimation, the MV we get may not reflect the true motion of the scene. Figure 5 shows this phenomenon. Figure 5a is the 25th frame of mother-CIF.yuv, and figure b illustrates the motion vectors of the red squares after motion estimation, and c shows the corresponding matching error. Compared Fig. 6a with b, we can see that only the head of the mother moved. There is no movement with the child and the background. But when we predict frame 27 from 25 using Full-Search (FS) algorithm, we find the MV of the background is not what we expected (0,0), as shown in Fig. 5b. This is the effect of the illumination change. When we

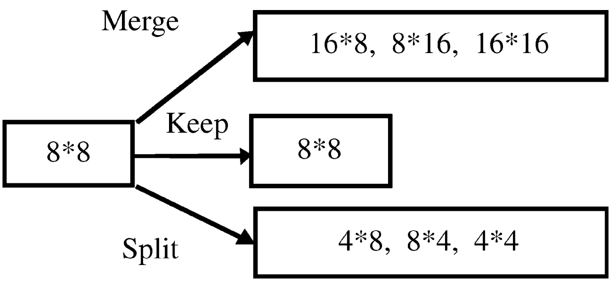


Fig. 4 Split and merge

a

(-1,0)	(0,-3)	(-2,6)	(2,-5)
(0,2)	(0,3)	(-1,5)	(-1,2)
(0,-2)	(-5,3)	(2,3)	(7,-3)
(7,-2)	(4,-4)	(-2,1)	(2,-4)

b

49	57	56	48
54	45	47	46
46	46	52	52
47	48	41	50

c

Fig. 5 Illumination effect on ME **a** 25th frame of mother-CIF.yuv; **b** motion vectors corresponding to the *red square*; **c** matching error

use these MVs as the measurement to decide split and merge, the background will be further sub-split into 8×4, 4×8 or 4×4 blocks, which is not what we expected.

In this paper, an improved "bottom–up" VSBM method is proposed for motion estimation. Different from previous work, we do not need a threshold to determine which MV should stay in the candidate set. For each minimum block, we only keep MVs which result in the minimum SAD, so we can keep the best predict precision with less number of MVs. To speed up the procedure, we use a macro-mode prediction (MMP) method to predict the mode of current block from that of the corresponding reference block. This MMP method does not introduce any loss to the ME precision. In order to reduce the illumination effect on mode selection, we add a post-processing step to deal with those macro-blocks split into 4×4 size blocks.

The paper is organized as follows: In Section 2, we first introduce the theory of the "bottom–up" VSBM algorithm, and then propose some modifications to improve the

a b

Fig. 6 25th and 27th frame of mother-CIF.yuv **a** 25th frame; **b** 27th frame performance of the VSBM. In Section 3, the illumination effect on the selection of block mode is further discussed. In Section 4, experimental comparison between the proposed method and Rhee's method are reported. Finally, we conclude the paper in Section 5.

2 Improved VSBM with prediction

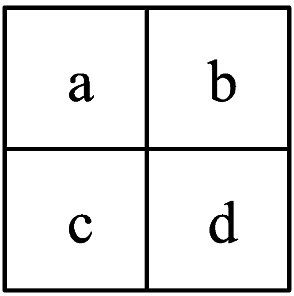
2.1 General procedure

The heuristic algorithm proposed in [12] is based on the observation that the true motion cannot be obtained just using local information for a moving object. Especially when motion estimation is performed by using small size block, such as 4×4, the matching results may be very close to the global minimum, but not to the true motion of the object. The true motion of the moving object can be partially obtained by choosing the common "candidate" motion vectors in the neighboring blocks. A threshold, which is calculated on a frame by frame basis, is used to determine which vectors should be included in the candidate sets. As we know, the motion levels of different parts in a frame are different. If the whole scene of one frame seems relatively stationary, the threshold will be large for the stationary regions, because it is proportional to the minimum mean absolute matched error of the entire frame [12]. Thus, the number of candidate MVs is increased not only for the stationary blocks, but also for the blocks with detail motions, so blocks with detail motion information are more likely to be merged, which will decrease the prediction precision. Some researchers [13, 21] proposed to use the Rate-Distortion optimization methods to get a tradeoff between rate and distortion, that is, making distortion minimum under a certain rate and vice versa.

In this paper, we propose a new method of VSBM. We seek for the minimum error based on the idea of VSBM. We do not use a threshold to determine which motion vector should be included in the initial set, but keep all the motion vectors resulting in the minimum prediction error. This method is equivalent to setting the minimum prediction error of each block as the threshold. After we get the initial sets for all mini-blocks, a merging procedure begins.

The merge procedure performed is the same as in [1]. As shown in Fig. 7, if block "a" can merge with b, and block c can merge with d or if block a can merge with c, and block b can merge with d, then rectangle block mode is used. If the four blocks can be merged together, then a larger square block mode is used. Different from [1], we use the minimum 4×4 size blocks as the initial block size, not 8×8 size blocks. Because we have seven block modes to select, we store the final block mode in a tree structure, as shown in Fig. 8. The tree structure of Fig. 8a represents the block structure in b. In this tree structure, the root node represents macro-block mode. Each node (including root node) can have two or four children, and it can be also a leaf node without children.

Fig. 7 The partition of macro-block (or sub-block)



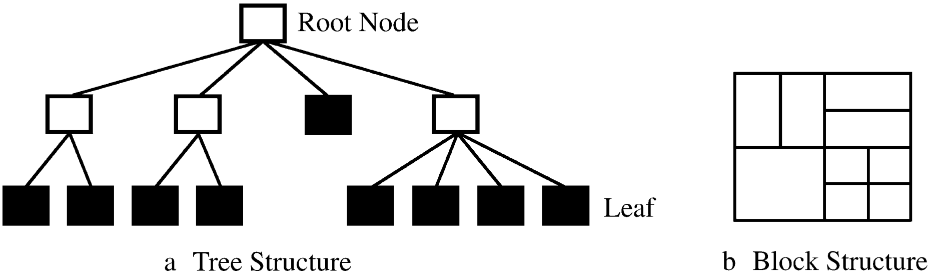


Fig. 8 Tree structure

2.2 Macro-mode Prediction (MMP)

Most of the VSBM algorithms suffer from intensive computation load [19]. In this paper, we propose a macro-mode prediction method to reduce the computational complexity. In VSBM, large or macro blocks are used to represent those stationary parts or parts with uniform motion, while small size blocks are used to represent parts with detailed motion. According to the observation, the distributions of macro-blocks between consecutive frames show great correspondence, while the distributions of small or minimum blocks may change a lot according to the content of the video. Figure 9 shows an example.

It is reasonable to only predict those macro-blocks, that is, 16×16 size blocks, from the previous frame. For each macro-block in the current frame, if the mode of the corresponding block in the previous frame is of macro-block mode, we use (MV_x, MV_y) as the prediction MV and (0,0) as a candidate MV to predict the mode of current macro-block, where (MV_x, MV_y) is the motion vector of the macro block in the previous frame. The average of SAD of that block is used as the threshold to determine whether the current block should use the predict MV or not. If the prediction error is larger than the threshold and (MV_x, MV_y) is not (0,0), (MV_x, MV_y) is set to (0,0) and the procedure above is repeated. If the SAD got using the predict MV is still larger than the threshold, this block has to undergo the whole VSBM process. Otherwise, the ME process is over, and the mode of this block is macro-block mode and (MV_x, MV_y) is its MV.

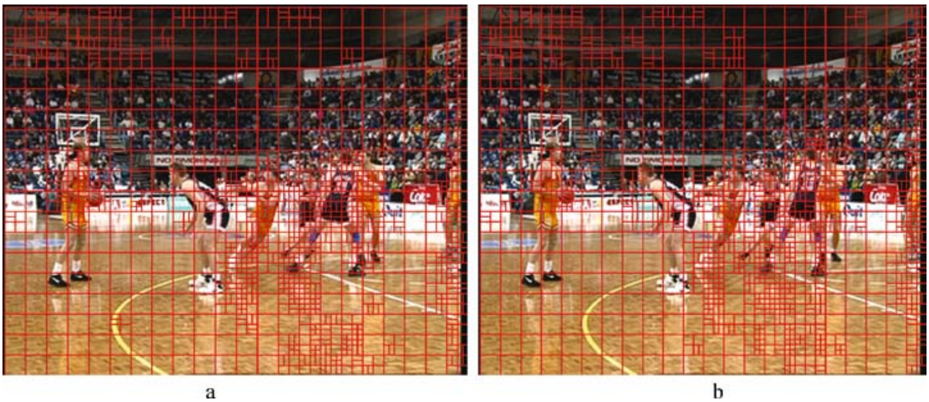


Fig. 9 The mode selection results of applying "bottom–up" VSBM algorithm on basketball-CIF.yuv (CIF) **a** the result of predicting the seventh frame from the fifth frame; **b** the result of predicting the ninth frame from the seventh frame

When dealing with macro-blocks, there is a problem of "majority effect," that is, the local information is obscured by the average effect of the large size block. On seeing this, we split the macro-block into four sub-blocks when applying this MMP method. If the average prediction error of every sub-block is smaller than that of the corresponding macro-block in the previous frame, then prediction mode is selected, otherwise the VSBM process is directly performed.

The flowchart of the prediction method is shown in Fig. 10, and can be summarized as follows:

1. If the mode of the corresponding macro-block in the previous frame is macro-block mode, go to Step 2; otherwise go to Step 3.
2. Split the current macro-block into four sub-blocks, and get the average of the SAD calculated with motion vector (MV_x, MV_y) for each sub-block. Denote the current average of the SAD as SAD_ac and the average of SAD of the macro-block in the previous frame as SAD_pre. For each sub-block, if SAD_ac < SAD_pre, (MV_x, MV_y) is set as the MV of current block and go to Step 4. Otherwise, use the data predicted with (0,0) to update SAD_ac. And if SAD_ac is smaller than SAD_pre, (0,0) is set as the MV of the current block and go to Step 4, otherwise go to Step 3.
3. Perform our improved VSBM algorithm.
4. Deal with the next macro-block.

Whether this MMP method can accelerate the whole ME procedure depends on the content of the video. The larger the amount of macro-block mode, the more this prediction

Fig. 10 Prediction flowchart

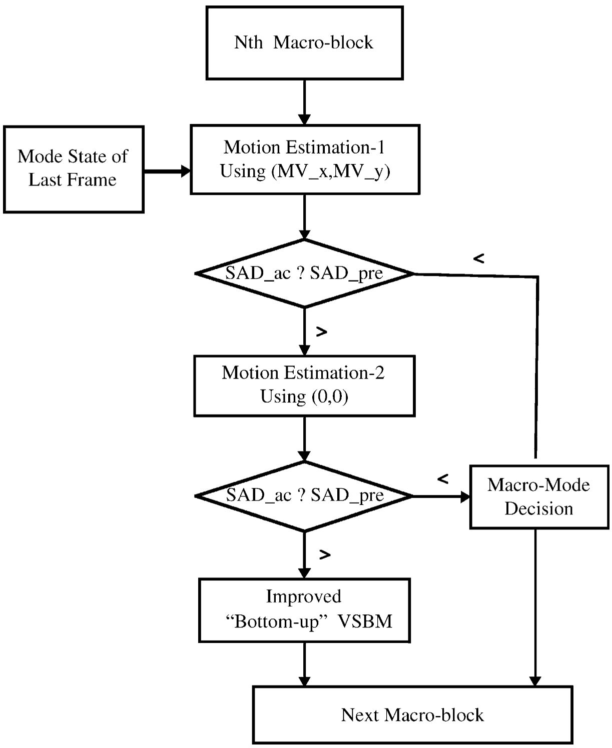


Table 1 The block statistics of akiyo-CIF.yuv (frame interval: 2)

Number Frame no	Total blocks	MBs	MBPs	MB/MBP (%)	PSNR
5	1,507	290	261	90.00	45.820
7	1,582	285	254	89.12	44.594
9	1,643	282	259	91.84	45.836
11	1,746	271	268	98.89	43.823
13	2,067	251	233	92.83	42.081

MBP: Macro Block with Prediction

method can accelerate the ME procedure. If MMP is used, we only need to perform 4–8 times SAD calculation comparing with 225 times when using full search method (when the search window is set as 15×15 pixels). Tables 1 and 2 show the number of macro blocks after mode selection (MBs column) and the number of macro blocks taking advantage of our prediction method (MBPs column), respectively. The background of akiyo-CIF is steady and the motion parts mainly lie in the head of the woman (Fig. 2b), the distribution of macro-blocks shows great correspondence, which can be seen from the "MB/MBP (%)" column. The macro-blocks undergoing our prediction procedure is above 89.12%, even as high as 98.89%. From Table 2 we can see that when predict frame 7 using the block distribution of 5, there are 234 macro-blocks and 101 macro-blocks can fulfill the requirement to use our prediction method. The corresponding data between frame 7 and frame 9 are 250, among which 154 macro-blocks are predicted using our method. The MB/MBP rate is 61.60%. Since the sequence type of state-of-the-art coding standard such as H.264, is usually IBPBP or IBBPBBP [15]. That is to say that the reference frame interval is 2 or 3, so above 43.16% macro-blocks in scenes affluent with large motion and 89.12% in scenes with large steady background can take advantage of our MMP method.

3 Eliminating the illumination effect

In Section 1.2, we have mentioned the illumination effect on mode selection. The common phenomenon is that scene (background and foreground) does not move, while the illumination undergoing changes. In this case, the value of Y component varies stochastically, which does not mean the motion of corresponding object but only reflects the illumination change. This kind of phenomena has direct impact on the selection of MVs in the candidate sets, which in turn influences the merge procedure. As shown in Fig. 5, the region marked with grid does not move between frame 25 and frame 27, but the MVs obtained by our improved VSBM (IVSBM) method and Full Search (FS) algorithm are the

Table 2 The block statistics of basketball-CIF.yuv (frame interval: 2)

Number Frame no	Total blocks	MBs	MBPs	MB/MBP (%)	PSNR
5	2,018	223	152	68.16	31.509
7	1,968	234	101	43.16	31.440
9	1,879	250	154	61.60	31.823
11	1,784	258	162	62.79	32.092
13	1,712	262	159	60.69	32.645

Fig. 11 Motion vector obtained by IVSBM and FS

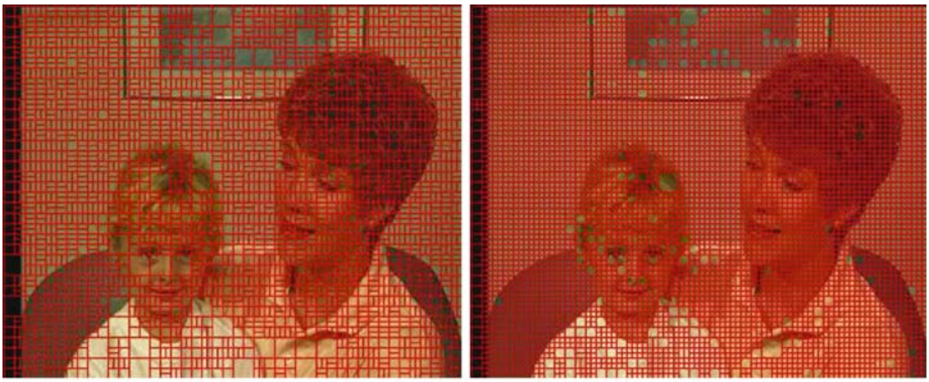
(-1,0)	(0,-3)	(-2,6)	(2,-5)
(0,2)	(0,3)	(-1,5)	(-1,2)
(0,-2)	(-5,3)	(2,3)	(7,-3)
(7,-2)	(4,-4)	(-2,1)	(2,-4)

same, as shown in Fig. 11. As we mentioned above, it is impossible to merge these mini-blocks basing on these MVs. The block mode distribution after motion estimation is shown in Fig. 12. Figure 12a is the result of Rhee's "bottom–up" VSBM algorithm. We keep 15 MVs in candidate sets for each 4×4 blocks. Figure 12b is the result of our proposed IVSBM algorithm. From Fig. 12, we can see that even though the perceived motion lies in the head of the mother, under the effect of illumination change, the mode selection is not reasonable because mini-blocks are used to represent regions without any motion.

In order to reduce the effect of illumination change on mode selection, we use a post-processing step to deal with stationary regions represented with mini-block mode. As shown in Fig. 13, after mode decision step, if the macro-block is found to be the mini-mode, we use an illumination removal step to re-select its mode. We first confirm the matching error of all the mini-blocks is smaller than 32, otherwise the region corresponding to this macro-block is considered to be of detailed motion, therefore should be represented with mini-blocks.

The reason why we select 32 as the threshold is because human eyes are not very sensitive to residual error under 32. Figure 14 shows several degree of residual error. The top–left block represents residual error 4, the residual error of the right one is always 4 larger than its left neighbor, and the bottom–right block turns out to be 64. From these blocks, we find that difference under 32 is hardly noticed with human eyes. So we select 32 as the threshold sounds more reasonable.

For all the mini-blocks, we first calculate the number of MVs whose corresponding SAD are smaller than 32. MV with the largest number is taken as the MV of this macro-block. A matching procedure is redone for those mini-blocks which do not contain this MV in their candidate MV sets. If the SAD of all the mini-blocks is smaller than 32, the mode of this macro-block is changed to macro-block mode and the MV is taken as its motion vector.



a b

Fig. 12 Block-mode distribution with illumination change **a** block-mode distribution using VSBM algorithm; **b** block-mode distribution using IVSBM. (Sequence: mother-CIF.yuv, use frame 25 to predict frame 27)

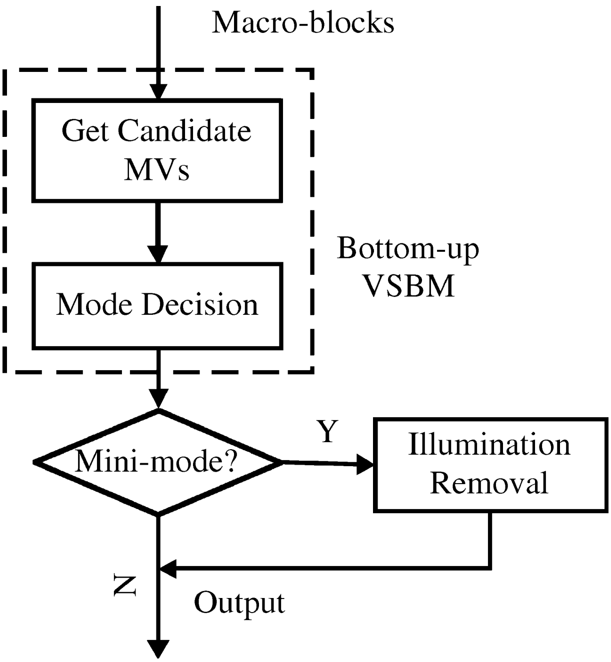


Fig. 13 Flowchart of illumination removal

However, the MVs in the candidate sets may not reflect the true motion of each block under the effect of illumination change, so we take (0,0) as the backup. The procedure is repeated for (0,0) if the MV with the largest amount failed to change mode to the macro-block mode.

Figure 15 shows the result using our illumination removal method. Compare Fig. 15 with Fig. 12, we can find that most background is represented with macro-blocks and foreground is represented with sub-blocks, which is the essence of VSBM algorithm. Table 3 illustrates the performance comparison among VSBM, IVSBM and IVSBMwIR (IVSBM with Illumination Removal), from which we can see that the block number of IVSBMwIR is 1861 less than that of IVSBM with 0.275 PSNR losses. Figure 16 shows another example. Figure 16a illustrates the mode selection result using IVSBM algorithm

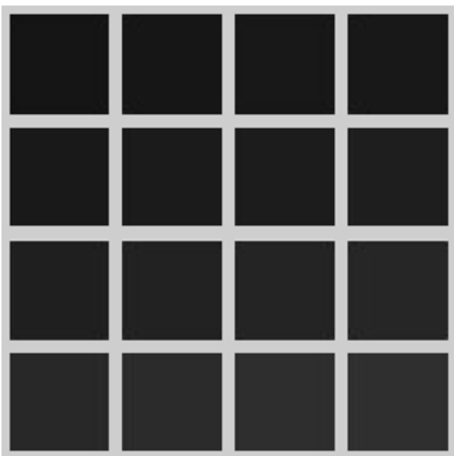


Fig. 14 Residual error

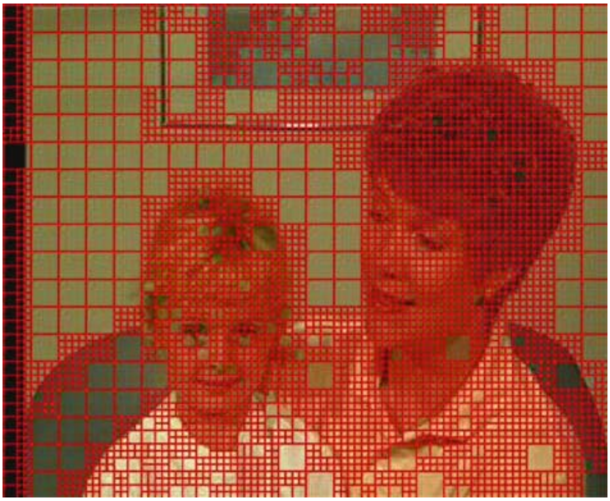


Fig. 15 Block-mode distribution after illumination removal (mother-CIF.yuv)

when predicting frame 4 from frame 2; b is the result using IVSBMwIR method. The final block number of IVSBMwIR is 3,044, which is 604 smaller than that of IVSBM with 0.01 PSNR losses, as shown in Table 4. Some of the blocks whose mode changed are demonstrated with yellow circles in dashed lines. From Figs. 15 and 16, we can see that our illumination change removal method mainly works on areas where Y component value varies smoothly.

4 Experiment results and analysis

The program is written in C++ and the source code is complied on Visual C++ 6.0 platform. Three CIF format sequences, that is, basketball-CIF.yuv, mother-CIF.yuv and akiyo-CIF.yuv are used to compare the proposed method with FSM (Fixed-Size Matching) with block size 4×4, VSBM [12], the improved VSBM and VSBM with illuminant change removal proposed in this paper. The basketball sequence represents those sequences with complex background and foreground motion. The akiyo sequence is a representative of sequences with stationary background. The mother sequence is mainly used to demonstrate the effect of our illumination removal method. In the following, we denote the proposed VSBM method as IVSBM and IVSBM with illuminant removal as IVSBMwIR for simplicity. We set the search range as seven only for speeding up the test period.

We use the original $n-2$th frame as the reference frame of the nth frame, not the frame reconstructed after transformation, quantization and inverse transformation and quantization. The PSNR results are shown in Fig. 17. We see that our IVSBM algorithm can achieve

Table 3 The performance of different algorithms under illumination change (mother-CIF.yuv)

Number Algorithm	Total blocks	MBs	MBPs	MB/MBP (%)	PSNR
VSBM	2,665	29	0	0.0	39.875
IVSBM	5,733	3	2	66.67	41.007
IVSBMwIR	3,872	128	70	54.69	40.732

IVSBMwIR: IVSBM with Illumination Removal

Fig. 16 IVSBM vs. IVSBMwIR (paris-CIF.yuv) **a** basketball-CIF.yuv, **b** akiyo-CIF.yuv, **c** mother-CIF.yuv

almost the same PSNR performance as FSM, which is 0.5 db above Rhee's "bottom–up" VSBM algorithm. For frames without the effect of illumination, the performance of IVSBMwIR is almost the same as that of IVSBM. The splitting macro-block into four sub-blocks mode prediction method prevents the displacement error from increasing.

Figure 18 shows the corresponding block amount relationship between these four methods. We can see that for basketball-CIF.yuv, the number of blocks needed in IVSBM and IVSBMwIR algorithms is almost the same, which is about 600–2,000 more than that of VSBM. As for akiyo-CIF.yuv, the block amount difference between VSBM and IVSBM and IVSBMwIR is about 1,000.

Figure 18c shows the block amount relationship of mother-CIF sequence, the PSNR performance is showed in Fig. 17c and the corresponding data is listed in Table 5. We can see that the block number of IVSBMwIR is about 2,000 less than that of IVSBM with 0.45 PSNR losses. But from Fig. 15, we can find out that the mode selection is much more reasonable compared to VSBM and IVSBM algorithms.

If the number of blocks increases, the direct effect on compression rate is that there are more MVs needed to be coded. Nowadays, the state-of-the-art coding standards use prediction method to encode MVs, that is, only the difference between current MVs and reference MVs need to be coded. This difference is usually very small, so the number of MVs won't have much effect on the compression rate.

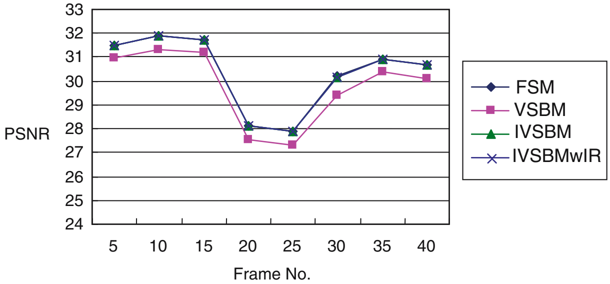
5 Conclusion

In this paper, we proposed an improved VSBM algorithm. For each block, we only keep those motion vectors resulting in minimum prediction error, and we do not need a threshold

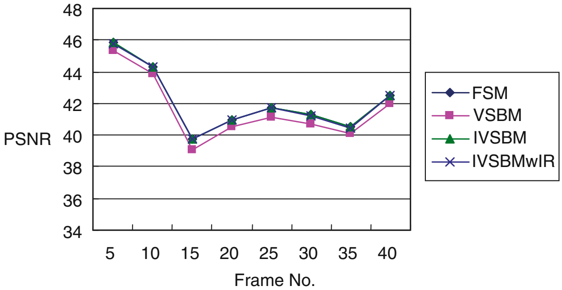
Table 4 The performance of different algorithms under illumination change (paris-CIF.yuv)

Number Algorithm	Total blocks	MBs	MBPs	MB/MBP (%)	PSNR
VSBM	1,814	163	0	0.0	32.397
IVSBM	3,648	121	80	66.12	33.181
IVSBMwIR	3,044	162	100	61.73	33.171

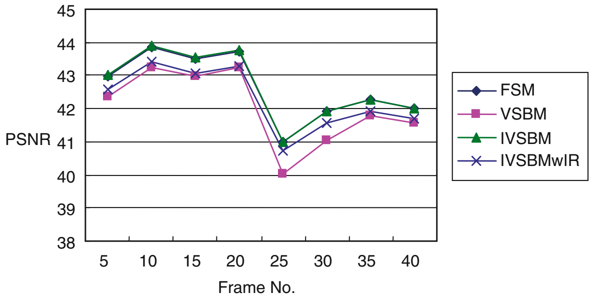
Fig. 17 PSNR comparison: FSM, VSBM, IVSBM and IVSBMwIR **a** basketball-CIF.yuv, **b** akiyo-CIF.yuv, **c** mother-CIF.yuv



a basketball-CIF.yuv



b akiyo-CIF.yuv



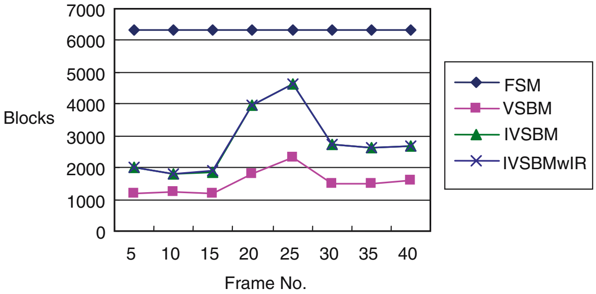
c mother-CIF.yuv

to determine which vector should be contained in the candidate set. In a sense, this method is equivalent to use the minimum matching error of each block as the threshold. To speed up the motion estimation procedure, we proposed a mode prediction method for macro-blocks, which expedites the process without introducing any loss of prediction precision.

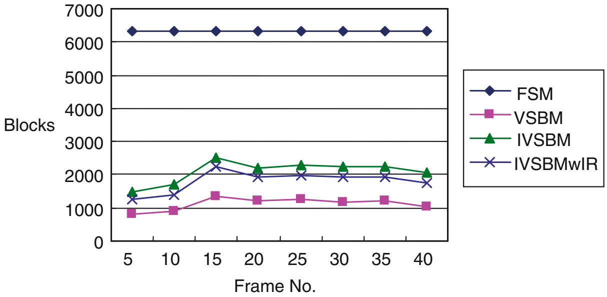
In order to reduce the effect of illumination change on mode selection, we use a post-processing step to re-decide the mode of macro-blocks assigned mini-block mode. The experiment results illustrate the effectiveness of our method. But it seems that the ideal way should detect the illumination change beforehand, and then perform the motion estimation and mode determination procedures.

In "bottom–up" VSBM algorithm, all the state-of-art algorithms use Full Search (FS) algorithm to get the candidate MVs. There are many fast algorithms to improve the performance of FS algorithm, such as Three-Step Search (TSS) and New TSS (NTSS) [9], block-based gradient decent search (BBGDS) [10], hexagon-based search (HEXBS) [23] and Enhanced HEXBS [24], etc. Zhou et al. [22] proposed a variable block-size MV prediction method, which can speed up the ME process. But all these improvements mainly

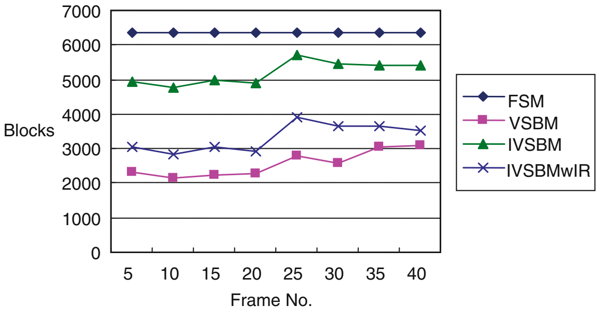
Fig. 18 Blocks amount relationship



a basketball-CIF.yuv



b akiyo-CIF.yuv



c mother-CIF.yuv

Table 5 PSNR values corresponding to Fig. 17c

	FSM	VSBM	IVSBM	IVSBMwIR
5	42.99	42.35	43.00	42.60
10	43.87	43.26	43.89	43.42
15	43.52	42.99	43.53	43.08
20	43.74	43.23	43.75	43.30
25	41.01	40.03	41.01	40.71
30	41.90	41.03	41.92	41.57
35	42.29	41.77	42.29	41.94
40	42.01	41.56	42.01	41.68

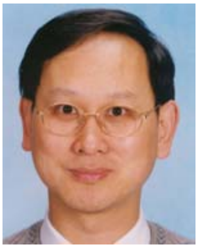
concern the start point and the selection of matching points. Our future work will focus on designing a fast algorithm to find candidate MVs for guiding the merge procedure in the "bottom–up" VSBM algorithm.

References

1. Chan M, Yu Y, Constantinides AG (1990) Variable size block matching motion compensation with applications to video coding. Proc Inst Elect Eng, 1 137(4):205–212 (Aug)
2. Chen ZB, Zhou P, He Y (2002) Fast integer pel and fractional pel motion estimation for JVT. JVT-F017r1.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, Island, JP, 5–13 December
3. Draft ITU-T recommendation and final draft international standard of joint video specification, ITU-T Recommendation H.264 and ISO/IEC 14 496–10 AVC, Joint Video Team, May 2003.
4. Ferri FJ, Malo J, Albert J, Soret J (1998) Variable-size block matching algorithm for motion estimation using a perceptual-based splitting criterion, Pattern recognition, 1998. Proceedings. Fourteenth International Conference on Volume 1,16–20 Aug. p 286–288 vol 1.
5. Information technology-coding of audio-visual objects-part 2: visual, ISO/IEC 14 496-2, 1999.
6. Information technology-coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-Part II:Video, ISO/IEC 11 172-2, 1993
7. Information technology-generic coding of moving pictures and associated audio information: video, ISO/IEC 13 818-2 and ITU-T Recommendation H.262, 1996
8. ITU-T, Video coding for low bitrate communications, ITU-T Recommendation H.263, version 1, Nov. 1995, version 2, Jan. 1998.
9. Li R, Zeng B, Liou ML (1994) A new three-step search algorithms for block motion estimation. IEEE Trans Circuits Syst Video Technol 4:438–442, Aug
10. Liu LK, Feig E (1996) A block-based gradient descent search algorithm for block motion estimation in video coding. IEEE Trans Circuits Syst Video Technol 6:419–423, Aug
11. Martin GR, Packwood RA, Rhee I (1996) Variable size block matching motion estimation with minimal error," Proceedings of IS&T/SPIE Symposium on Electronic Imaging: Science and Technology (Digital Video Compression: Algorithms and Technologies), San Jose, CA, USA, (Jan)
12. Rhee I, Marlin G, Muthukrishnan S, Packwood R (2000) Quadtree-structured variable-size block-matching motion estimation with minimal error. IEEE Trans Circuits Syst Video Technol 10:42–50, Feb
13. Sullivan GJ, Baker RL (1991) Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks. Global telecommunications conference, 1991. GLOBECOM '91. Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services 1 (2–5), pp 85–90, Dec.
14. Tekalp AM (1995) Digital video processing. Prentice Hall, New York, p 95
15. Tourapis AM, Sühring K, Sullivan G (2004) Proposed H.264/MPEG-4 AVC reference software manual. JVT-M012, Palma de Mallorca, Spain 18–22 Oct
16. Tu YK, Yang JF, Shen YN, Ming-Ting Sun (2003) Fast variable-size block motion estimation using merging procedure with an adaptive threshold. Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on, Volume: 2, 6–9 July, p II 789–92 vol 2
17. Video Codec for Audiovisual Services at p_64 Kbit/s, ITU-T Recommendation H.261, Mar. 1993.
18. Video Coding for Low Bit Rate Communication, ITU-T Recommendation H.263, Feb. 1998.
19. Wang Z, Na Lin, Wang GX (2004) Constrained growing based variable size block motion estimation. Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on 2(27–29):910–913, June
20. Xin J, Sun MT, HV (2003) Diversity-based fast block motion estimation. Proc. IEEE International Conference on Multimedia & Expo (ICME) 3:525–528, July
21. Yang Y, Hemami SS (1998) Rate-distortion-based combined motion estimation and segmentation. ICIP 98. Proceedings. 1998 International Conference on, 4–7 Oct
22. Zhou Z, Sun MT, Hsu YF (2004) Fast variable block-size motion estimation algorithms based on merge and split procedures for H.264/MPEG-4 AVC. ISCAS2004, May
23. Zhu C, Lin X, Chau LP (2002) Hexagon-based search pattern for fast block motion estimation. IEEE Trans Circuits Syst Video Technol 12:349–355, May
24. Zhu C, Lin X, Chau LP, Po LM (2004) Enhanced hexagonal search for fast block motion estimation. IEEE Trans Circuits Syst Video Technol 14(10) (Oct)

Haifeng Wang received his B.S. degree in computer science from Beijing Institute of Technology, China in 2003. He is currently pursuing his Ph.D. degree in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, multimedia processing, video compression and object segmentation. The current emphasis of his work is on the global motion analysis for MPEG-4.

Qingshan Liu received his B.E. degree from Chinese Textile University (now Dong Hua University) in 1997 and his M.S. degree from South East University in 2000. He received his Ph.D. degree from the Institute of Automation, Chinese Academy of Science in 2003. Now he is an associate professor in the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Science. His current research interests are image and video analysis, machine learning, and pattern recognition.



Hanqing Lu received his B.E. and M.S. degrees from Harbin Institute of Technology in 1982 and 1985 respectively. He received his Ph.D. degree from Huazhong University of Sciences and Technology. Now, he is a professor at the Institute of Automation, CAS. His current research interests are image the video analysis, image and at the video retrieval, pattern recognition, and multimedia technology and system.