

Fast Motion Video Estimation

Nikunj Radadia

(Student), Atharva college of engineering, Department of Electronics and Telecommunication engg, Mumbai university (UGC, NAAC, AIU)

Mayur Sawant

(Student), Atharva college of engineering, Department of Electronics and Telecommunication engg, Mumbai university (UGC, NAAC, AIU)

Abstract

In this study, we evaluate fast motion estimation (ME) techniques in the context of a JPEG 2000-based video coding system for surveillance-type videos. The authors have designed a low-complexity algorithm, called block-selective ME, which restricts block matching to certain frames or blocks containing high motion. They compare the performance of our block-selective ME algorithm to a frame-based approach and to a standard fast-motion algorithm (three-step search (TSS)). For surveillance-type videos, the authors show that the block-selective approach achieves the peak signal to noise ratio (PSNR) quality of a full ME scheme for; 70–80% of the blocks. Moreover, this approach delivers a higher visual image quality compared to TSS, if the computational load for a set number of blocks were fixed. The authors have integrated our block-selective approach into different coders (H.264 and MPEG-2) and show that our approach is an outstanding alternative to fast-ME in low-complexity environments.

1 Introduction

With the increasing general importance of video surveillance systems, video coding systems are also becoming more important. In years past the quality of the video stream as well as the frame encoding rate was unsatisfactory. Today, both quality and frame encoding rate can be significantly improved using one of the two well-established video coders, H.264 and MPEG-2. However, in low-complexity environments such as surveillance videos, hardware solutions for H.264 and MPEG-2 are significantly costlier when compared with solutions for Motion JPEG 2000. In addition, H.264 coders (in both hardware and software) are complex and require substantial resources for encoding and decoding.

We solve this problem by developing an inter-frame JPEG

2000 two-dimensional (2D) system [A significant number of wavelet-based hybrid 2D algorithms have been suggested in the literature (e.g. [1–6]); however, all of them employ a full-frame motion estimation, which does not satisfy our low-complexity constraint.] with selective motion estimation, which requires fewer resources and is much less expensive overall, because JPEG 2000 hardware components are significantly cheaper than H.264 and MPEG-2 components. The baseline system is based on the

coding of intra-frames (I-frames) and differential frames with the JPEG 2000 standard. This is advantageous, because differential frames are easy to compute and they contain the motion content of the video. It has already been shown in previous work that Motion JPEG 2000 can be significantly improved upon [7]. In order to further improve on average visual quality, we needed to include some form of motion estimation/compensation (ME/MC). Since ME/MC is responsible for

50–80% of the computational demand in current video coding algorithms, the use of ME/MC would lead to a drastic rise in complexity. Hence, we have developed a low-complexity motion estimation algorithm called block-selective motion estimation. It takes advantage of the fact that video objects, or areas exhibiting a high level of motion, are subject to motion estimation, whereas low motion areas can be excluded from the motion estimation process entirely. This leads not only to reduced storage consumption, it also speeds up the coding process. Our block-selective approach only requires a simple mathematical framework and is thus predestined for use in low-complexity environments. Owing to its simplicity our motion estimation algorithm can be easily ported to other video coding systems. Since our system is able to calculate the motion content (high or low motion) of the frame, it is possible to adapt it to the motion content by setting the number of blocks subject to motion estimation. In this paper, we also compare our results to a frame-based selective ME/MC approach [8]. In order to demonstrate that our block-based approach can improve coding time and compression efficiency of conventional motion estimation algorithms, we have chosen to work with surveillance-type videos, because they contain only object motion or a low amount of overall motion. Thus, they lend themselves perfectly to testing our block-selective approach.

In Section 2, we briefly describe the baseline system that codes I-frames and differential frames. In Section 3, we discuss both the block-selective ME/MC approach with a frame-selective approach. In all our experiments, which are presented in Section 4, we have compared the ME/MC approaches with both a full ME/MC and a TSS version. In addition, we will present the results of the integration of the block-selective approach with H.264 and MPEG-2. Section 5 offers a conclusion, which will provide readers with perspectives for the future, such as the porting whole framework to other coding environments and the inclusion of additional motion indicators.

2 Baseline system

In the low-complexity baseline software system (without ME/MC)(Fig. 1 (left)), the encoding of frames is done via the most well-known open source JPEG 2000 reference implementation, Jasper. (<http://www.ece.uvic.ca/mdadams/jasper/>). The initial raw video frame is read in and encoded as an I-frame. For subsequent frames, a differential frame is computed between the current frame to be encoded and the reconstructed reference frame. Differential frames $d_i(x, y)$ based on the simple arithmetic difference between pixels with identical positions (where i defines the number of the frame that is currently encoded) in temporally adjacent frames $f_i(x, y)$ and $f_{i-1}(x, y)$ (assuming no motion) are quick and easy to compute:

$$d_i(x, y) = |f_i(x, y) - f_{i-1}(x, y)|$$

The frames $d_i(x, y)$ can be used to determine motion present between frames by computing the following parameters (with N, M being the image dimensions): in our paper, we define indicators as mathematical values, analyzing our differential frames, indicating the motion content (high or low motion) of the differential frame. Based on those ‘indicators’ the decision is made in a dual threshold scheme whether to code the current frame as a differential frame (little or no motion present, motion-compensated blocks or not) or as an I-frame (high motion content).

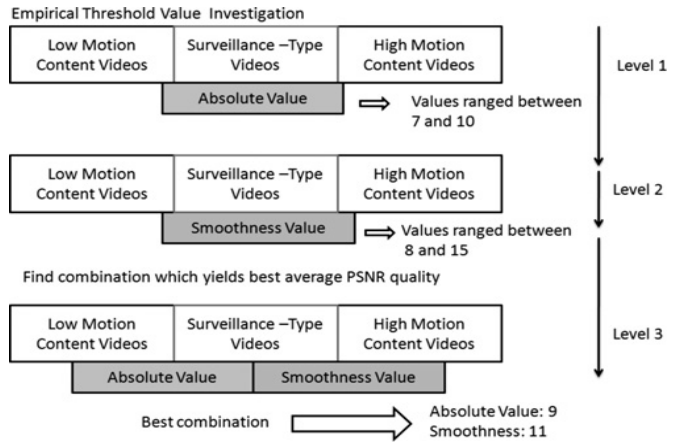


Fig. 2 Threshold investigation

The last indicator we use is the variance of a differential frame, which defines the variance on a block basis.

In order to find the appropriate set of indicators, we first computed results only with the absolute value indicator for both high and low motion content videos. Through this process we found out that by adapting our indicator to the image width and height, respectively, (mainly low motion videos, surveillance-type videos) the absolute value indicator was performing best within the range of 8–10. While high motion content videos were coded with a lower indicator value to achieve better average frame quality, which is measured in peak signal to noise ratio (PSNR) (dB), low motion content videos needed to be coded with a higher value to achieve better visual results (measured average PSNR quality). In Fig. 2, the empirical threshold investigation process is illustrated in an algorithmic way.

Second, we computed results for the smoothness indicator and found the best-performing individual indicators for smoothness as well. The variance indicator was left out at this stage because of performance constraints as the variance indicator yields to a greater computational load. Combining the absolute value indicator and the smoothness indicator posed a challenge. It was mainly done by testing different sets of these indicators.

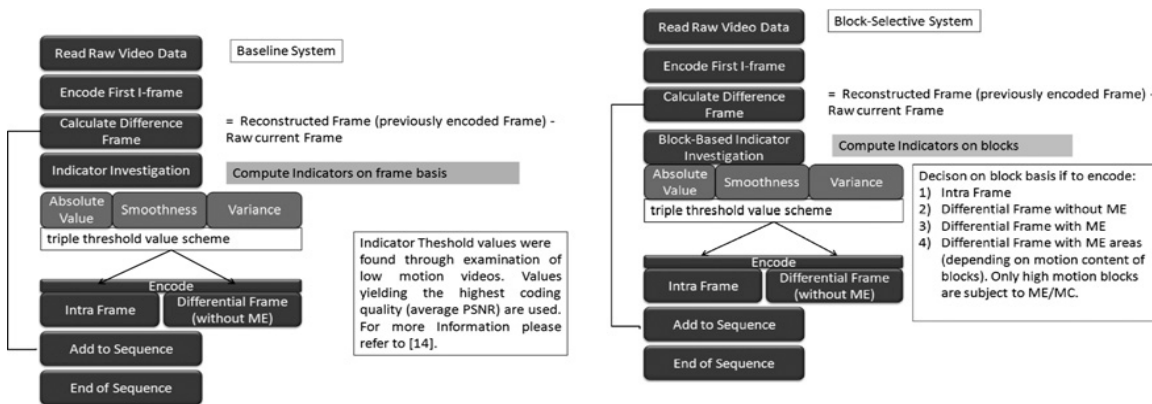


Fig. 1 Baseline system against block-based selective ME/MC scheme

best average performance in our study. Furthermore, we computed the correlation between the differential frame quality (PSNR) and the indicator to give our assumption a stronger basis. Fig. 3 below illustrates the interconnection between the smoothness inside the frame and its coding process by simply visualising the correlation between the coding PSNR and the indicator value.

We have computed these simple indicators on a block basis to determine whether to code an I-frame or a differential frame. In addition, it must be decided which blocks of the differential frame are subject to ME/MC. Coding indicators on a block basis is advantageous, because low object motion (motion apparent only inside single blocks) can be addressed on a block basis. It is easy to compute the general error distribution (standard deviation) inside a frame or an area (object) inside a frame. The motion information is used to decide whether to perform a full ME/MC on the whole frame or a block-selective ME/MC constrained by threshold values. By computing a large variety of different surveillance videos, we arrived at these threshold values, which achieved the best results in visual quality, and used them to decide how to code the frames. In an advanced development stage these threshold values can also be set adaptively. If the error (the motion itself) were distributed uniformly over the differential frame, a selective block ME/MC would lead to a drastic rise in complexity, since most of the blocks are subject to ME/MC. This complexity arises from the fact that most of the blocks would be subject to ME/MC. If this were the case, an I-frame would be coded to avoid increasing complexity. Motion vectors are coded separately using a Huffman coder and are included in the video stream. Moreover, we apply the indicators to differential frame on a block basis to address the high motion areas of a frame. High motion areas of frames are subject to ME/MC, because ME/MC is able to achieve better results for overall object motion. Based on the motion content of the differential frame, an I-frame, a motion compensated or non-motion-compensated differential frame is coded. (Within a motion-compensated differential frame, the number of blocks can vary). These three types of frames create an adaptive group of pictures (GOP) structure. [All state-of-the-art video coding systems employ a fixed GOP structure, whereby the GOP structure is defined as the repeated pattern of different frame-types in a video stream. One main advantage of our system is that the GOP structure is created based on the motion content of the video.]

Additionally, the block-selective approach creates differential frames with motion-compensated areas. An additional goal was to be able to adapt to both coding quality and coding rate dynamically. We created such a system, but it failed to satisfy our desire to maintain a low-complexity system. In particular, choosing the rate adaptively leads to a drastic rise in computational load (of up to 40%). Hence, we disregard this topic in this paper. In experiments, we have compared our baseline system to Motion JPEG 2000 and other coders with a fixed GOP structure. Our results show that our baseline system outperforms Motion JPEG 2000, but it does not achieve competitive results compared to MPEG-4 and H.264

3 Selective ME/MC

3.1 Block-selective ME/MC system: DJPEG2000

ME/MC is widely used in video coding for examining the differences between current frames and reference frames to exploit temporal redundancy and to compensate for object or camera motion. Block matching as employed in most systems does this on a per-block basis, assuming constant motion within a block by computing motion vectors for each block representing the motion of that block. Applying ME/MC to a video leads to an increase in quality. However, ME/MC causes a drastic rise in computational load. In order to reduce complexity, fast ME/MC algorithms have been developed (TSS, diamond search and so on) to optimise the process of ME/MC calculation by restricting the search space for each candidate block or by introducing more efficient matching procedures.

In this paper, we focus on selective ME/MC in the sense that not all parts of the video that correspond to residual frames are subject to ME/MC (see Fig. 1). The aim of this approach is to limit ME/MC adaptively to the parts of the video in which it is actually needed (high motion present) and not to use ME/MC for low motion areas. Note that this concept is entirely different to classical non-adaptive low-complexity ME/MC techniques (like TSS), where the search effort for all regions of the frames is reduced regardless of their motion content. In our approach, the decision to apply ME/MC is made on a block basis and ME/MC is only performed when a block is determined to

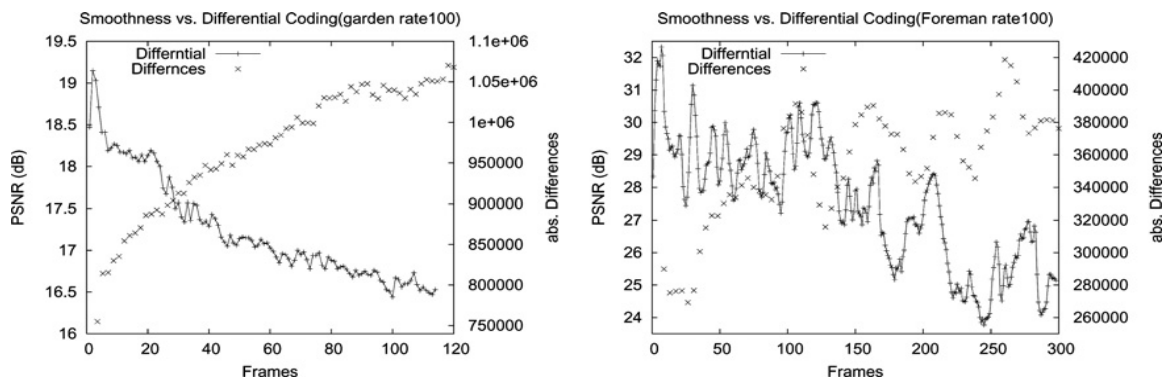


Fig. 3 Coding against smoothness (high motion content)

contain high motion.

The actual block-selective ME is conducted as follows. By analogy to the baseline approach, the current frame is subtracted from the reconstructed reference frame. Subsequently, the difference frame is tiled into blocks that are analysed for motion content by applying the motion indicators as outlined in the previous section. Note that this approach fits perfectly into the baseline system approach since the blocks of the difference frame are assessed using the motion indicators to determine the next I-frame in the adaptive GOP process. The additional computational load presents no disadvantage, because the motion indicators are further used in the coding pipeline. The actual value of the threshold on the motion indicators determines the amount of blocks subject to ME/MC. If ME/MC has been applied to a block, the new motion-compensated residual block is computed and inserted into the difference frame at the corresponding position. After this hybrid difference frame has been generated, it is JPEG 2000 compressed in the usual way. Note that blocks, which have not been subject to ME/MC, correspond exactly to blocks where the result of the ME/MC indicates a zero displacement (no motion present). We denote our coder with ME/MC as DJPEG2000.

3.2 Frame-selective ME/MC system

Frame-selective ME/MC has also been proposed. Its basic idea is to restrict ME/MC to frames exhibiting high motion; frames with low motion content are simply encoded as differential frames (zero-motion assumption). Motion activity in the current frame is estimated based on the average value of the motion vectors from the previous frame. By setting a proper threshold it can be determined whether the current frame is subject to ME/MC or encoded as simple arithmetic difference to the reconstructed reference frame. In the original version as proposed, ME/MC is skipped for a maximum of one differential frame in a row. The following frame is once again subject to ME/MC and examined with respect to motion activity. This approach thereby allows a reduction of ME/MC frames by a factor of 2. An improvement on this restriction can be achieved by adapting the skip rate threshold value. If the motion activity is found to be very low (determined by a second threshold value), more frames than one can be skipped. This leads to a higher reduction of computational load in ME/MC.

4 Experiments

We have created surveillance-type videos in their respective resolution (QCIF) (please refer to Fig. 4): Car from left to right (100 frames), Cycle (100 frames), Pillow (100 frames), Leavin Car (100 frames) and Rollin (100 frames). Three single frames are shown in the figure (frame 1, frame 50 and frame 100). These video sequences contain no camera movement, rather only object motion (mostly little objects). In order to be able to compare our results with standard video sequences, we have used following video sequences: Garden (100 frames, Camera Pan), Foreman (100 Frames, High Motion) and the Paris sequences (50 Frames, Low Motion). We used these self-created and standard video sequences for all further experiments (in their respective resolutions and frames). The sensor we use/have used is a Olympus Camedia Master SP-510 UZ, which is able to yield QCIF and CIF. The video codecs used are DJPEG2000, H.264 (Intra mode in which every frame is an I-frame

(i.e. no MC is used), and interframe mode with a constant GOP of 15 without B-frame functionality), MPEG-2 (GOP-15, with and without B-frames) and Motion JPEG 2000.

ME (i.e. block-matching) for the DJPEG2000 coder, H.264 Coder (use of the reference implementations $\times 264$ and JM 15.1), and the MPEG-2 coder is performed on 8×8 pixel blocks using single-pixel accuracy in non-overlapping mode (also because of low-complexity reasons). In DJPEG2000, we use a fixed GOP size of 15 frames instead of the adaptive GOP scheme to limit the observed effects to the selective ME process, because they would otherwise interfere with phenomena caused by varying GOP size. The motion vectors as well as the information that the blocks have been subjected to ME/MC are encoded via a Huffman Coder.

Using the open-source reference implementation Jasper presents a serious disadvantage, which has to be accepted. This source code is non-speed optimised, which makes it difficult to compare it with other codecs. We choose the free reference implementation of Jasper because we want a hardware-oriented code to build up an embedded low-complexity video codec based on JPEG 2000 hardware components. The hardware needed to implement a JPEG 2000 system is significantly less expensive than the components of other coding systems. In addition, several studies have confirmed that motion-based JPEG 2000 systems are faster than H.264 systems. Using optimised Kakadu v.4.2.1 software libraries Motion JPEG 2000, for example, the huge difference between the two reference implementations becomes apparent. To see detailed results of coding complexity and quality, please refer to Table 1, where we present averaged results for our set of videos. Motion JPEG

2000 is also easier to integrate into other video coding systems such as H.264 (low-complexity mode) and MPEG-2 (no B-frames); see Sections 4.1.2 and 4.1.3. The difference between optimised and non-optimised implementations with regard to coding performance is so massive that comparisons yield few meaningful results. Thus, comparing DJPEG2000 to H.264 and MPEG-2 under similar conditions is not feasible.

4.1 Block-selective against FULL ME/MC and TSS

4.1.1 Block-selective DJPEG2000: In general, for surveillance-type videos only area-restricted ME/MC is required to increase average visual quality, whereas the rest of the differential frame contains little or no motion. We want to show that in such an environment our selective block-based ME/MC approach saves on coding time compared with a TSS approach with similar visual quality. Once high motion areas are detected by using our set of indicators these blocks containing high motion are subject to a full-search ME/MC. It is clear that if 100% of blocks are used for coding a full-search ME/MC is applied. This will appear only in rare cases (complete scene change or overall motion). The selective approach can also be compared with a classical TSS, where less blocks have to be examined.

We investigate both approaches in Fig. 6, which shows PSNR (left y-axis label) and overall coding time (right y-axis label) for an increasing percentage of blocks (x-axis) used during the ME/MC process in the block-selective ME/MC approach. Plots for full search (FULL) and TSS ME/

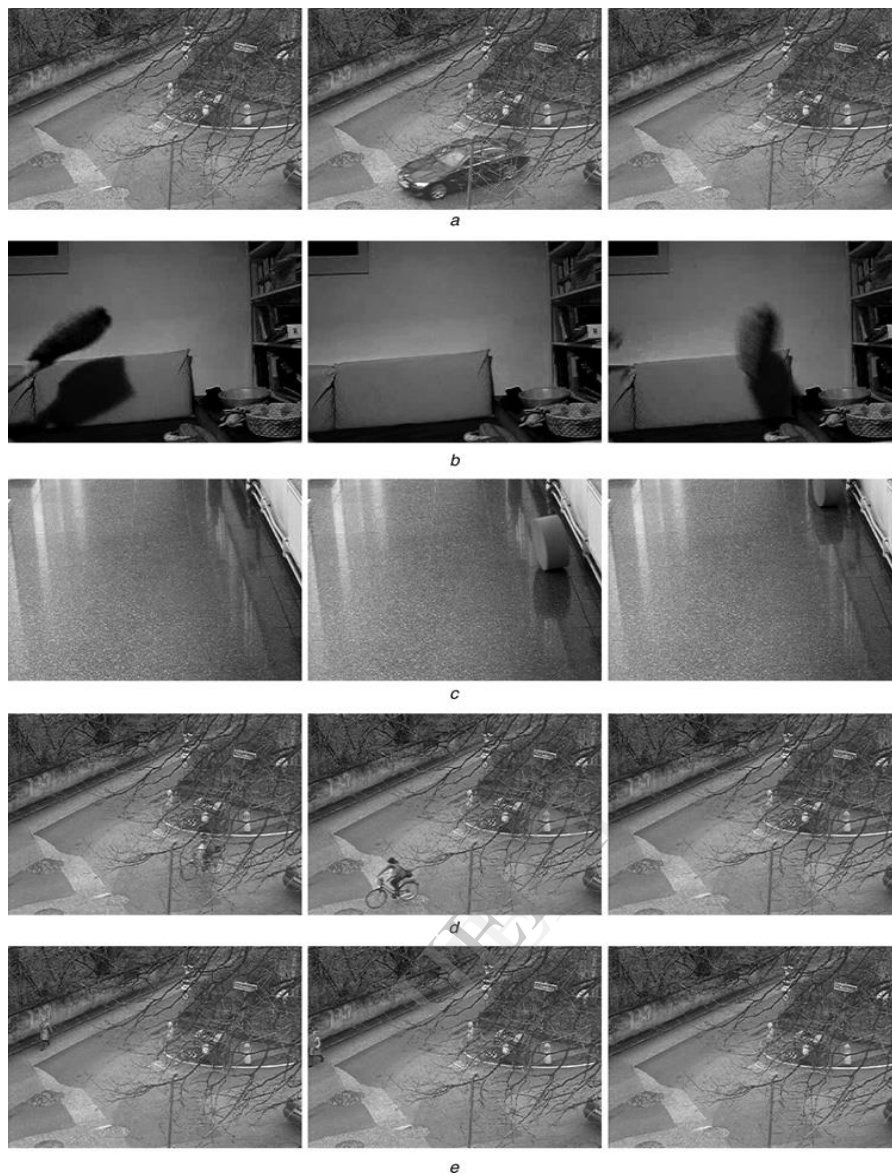


Fig. 4 Surveillance-type videos (frame extraction):

- a Car from left to right (static background, no camera pan, car is moving from the left side to the right and disappears, only object (car) is apparent)
- b Pillow (static background, no camera pan, pillow object is inserted from the left side to the scene and disappears afterwards)
- c Rollin (static background, only object motion is apparent, round plastic object (diameter $\frac{1}{4}$ 24 cm, height $\frac{1}{4}$ 15 cm))
- d Cycle (static background, no camera pan, object (cyclist) is moving from the right to the left side of the scene and disappears, no additional object motion)
- e Walker (static background, no camera pan, object (walking person) is moving from the upper right to left, additional moving objects (body parts))

of blocks used increases, as we expected. In particular, the quality of the whole video stream on average is clearly improved, when compared with an approach relying on the coding of simple differential frames. However, the shape of the PSNR and timing curves are not identical in all cases. We observed that a full search yields the maximum PSNR quality with about 70–80% of all blocks, and that quality is hardly improved when more blocks are employed. This is a favourable result since it indicates that block-selective ME/MC can be employed in the case of low-motion videos with only 70–80% of the blocks without significantly affecting PSNR quality. We note that this phenomenon is similar for TSS. For a comparison of these surveillance-type videos to standard video sequences, please refer to Table 2.

In addition, we compare our approach with the classical demand. For this purpose, we determine the PSNR and coding time of classical TSS (using 100% of all blocks) and compare the PSNR result to the PSNR of block-

selective ME/MC using a full search exhibiting the same coding time (i.e. using a lower percentage of blocks, e.g. 30–35%). This can be seen in Fig. 5: First, the percentage of blocks allowed for a full search for all graphs is determined by drawing a horizontal line from the coding time of TSS at 100% blocks and intersecting this line with the graph of the coding time of for the full search (refer to of Fig. 4 for visual explanation). (This usually leads to an intersection at 30–35% of the blocks.) Table 2 shows the resulting PSNR differences in decibel (dB), where a negative sign indicates a higher PSNR value for TSS.

Table 1 Complexity comparisons for coders: averaged results for surveillance-type and standard video sequences:

Coder's performance	PSNR, Db			Coding time, s		
	20	100	200	20	100	200
Compression rate ^a						
motion JPEG 2000 (Kakadu)	30.55	27.92	23.84	8.5	8.3	8.7
DJPEG2000 (Jasper) I	30.37	27.62	23.62	23	22	21
DJPEG2000 (Jasper) with ME/MC	32.49	29.56	25.01	64	62	61
DJPEG2000 (Jasper) without ME/MC	30.15	27.15	23.27	32	34	33
H.264 (×264) I,P	34.78	31.91	26.98	33	35	34
H264 (×264) I	31.02	28.34	24.91	11	13	12
MPEG-2:GOP15, with B-frames	33.23	31.02	26.23	54	51	54
MPEG-2:GOP15, without B-frames	32.94	30.78	26.32	34	34	32

^aCompression rate divides the original image size by the compressed image size, which is commonly used in our reference JPEG 2000 implementation (Jasper)

This is a desired result since it indicates that block-selective ME/MC can be employed with 70% of the blocks without significantly affecting PSNR quality. An additional result is shown in Fig. 6, where the upper and lower PSNR boundaries (100% of blocks used for ME/MC and 0% of blocks for ME/MC, as well as comparing our adaptive coder with ME/MC capabilities (selective ME) to the baseline system (dynamic coder). It shows that by employing ME/MC it can be significantly improved upon the baseline system, which was to be expected.

4.1.2 Block-selective H.264: We also provide individual visual results for the H.264 codec. The reference implementation of JM was used to port the block-selective scheme to H.264. We provide visual and tabular results for surveillance and standard video sequences. As stated earlier, only internal comparisons make sense because the performance differences between the coders are too massive. Therefore we present similar graphs as in the case of DJPEG2000. Please refer to Fig. 7 for surveillance-type videos as well as results for the Foreman sequence. It is clearly visible that despite a significantly higher average visual coding quality in the case of H.264 quality, the block-usage against the coding quality trend is similar in the case of DJPEG 2000.

Comparison between Selective ME/MC and THREE STEP SEARCH ME/MC

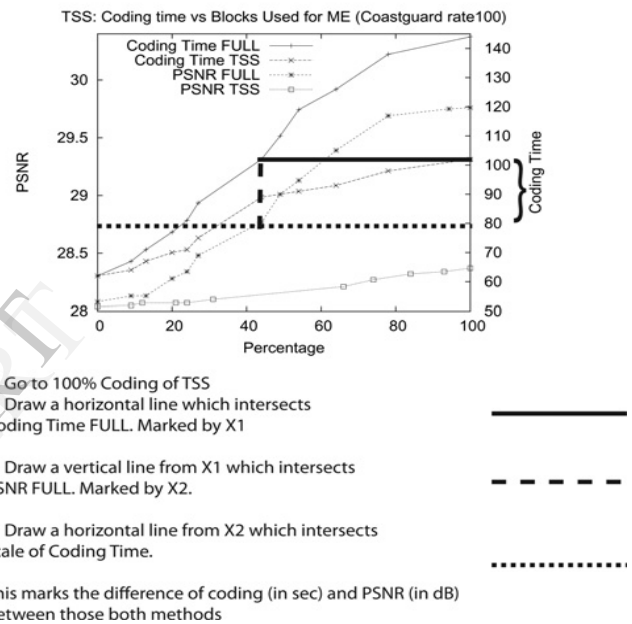


Fig. 5 Comparison between the TSS and block-selective approach

Table 2 DJPEG2000 comparisons: TSS against block-selective full-search (FS) ME/MC

Compression rate (Jasper)	DJPEG2000					
	20		100		200	
Results in PSNR (dB)	TSS	Block	TSS	Block	TSS	Block
TSS against block-selective FS ME/MC						
Carfromlefttoright	29.52	30.02	22.4	22.65	20.3	20.61
Cycle	29.6	29.85	22.4	22.2	19.9	20.2
Rollin	35.6	36.03	32.1	32.5	28.15	28.53
LeavinCar	34.52	34.63	24.52	24.63	20.91	21.08
Pillow	31.95	32.25	27.2	27.55	25.05	25.3
average of surveillance	32.2	32.5	25.7	22.9	22.8	23.1
Garden	24.1	24.3	18.7	18.8	16.7	16.6
Foreman	29.35	29.05	24.3	24.2	21.42	21.1
Paris	24.68	24.42	21.18	21.21	18.97	18.97

Fig. 6 Block-selective ME/MC PSNR boundaries for the Foreman and the Paris sequence: compression ratios 20 and 200

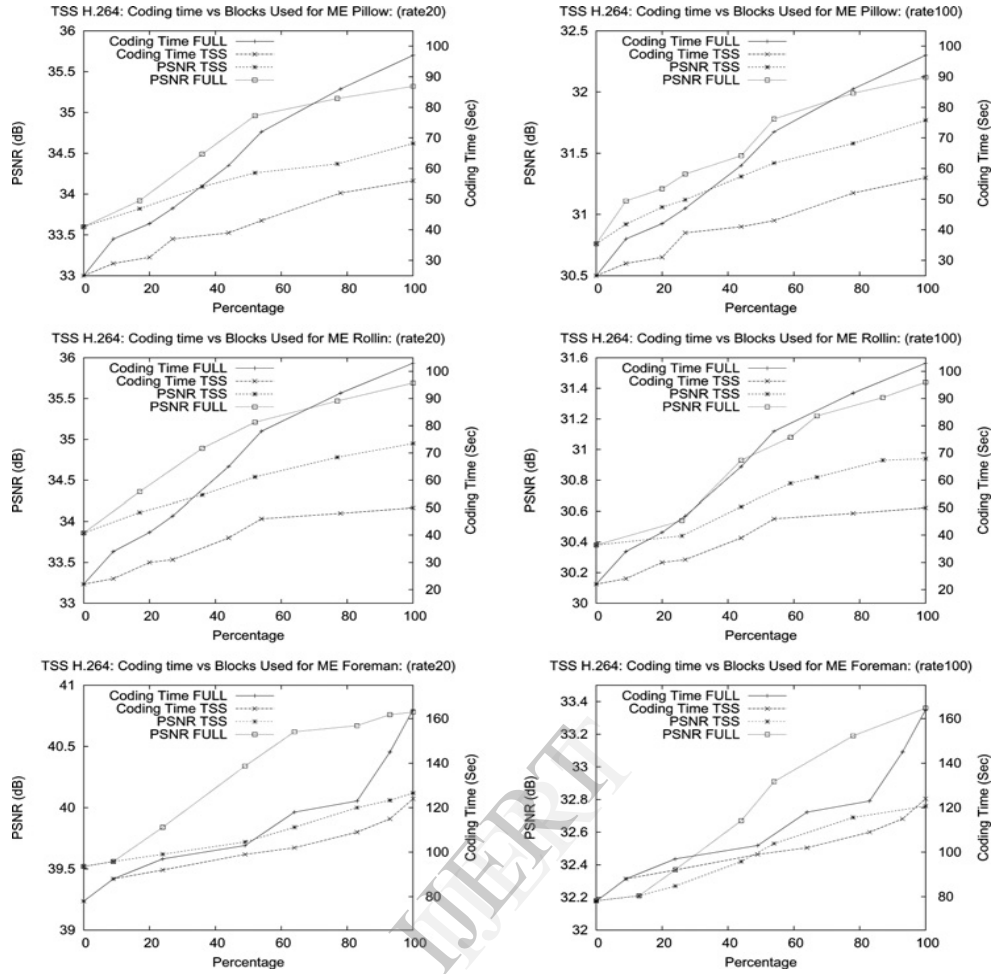


Fig. 7 TSS MPEG-2 block percentage compared with full-frame ME/MC for surveillance-type videos (Pillow, Rollin) and the Foreman sequence: compression rate 20,10

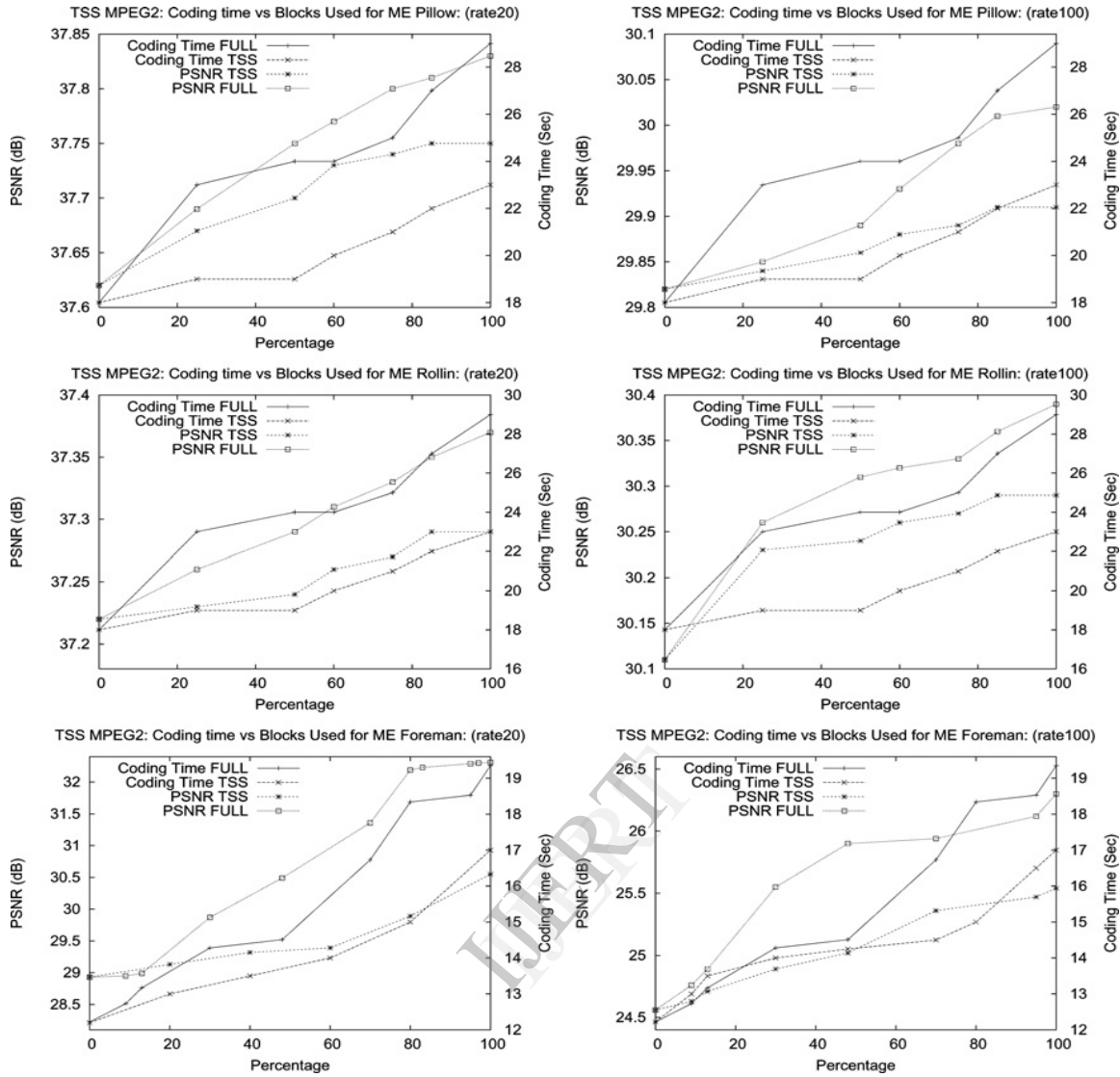


Fig. 9 TSS MPEG-2 block percentage compared with full-frame ME/MC for surveillance-type videos (Pillow, Rollin) and the Foreman sequence: compression rate 20,100

4.1.3 Block-selective MPEG-2: In addition, we port our block-selective scheme to an MPEG-2 coding environment. Please refer to Fig. 9 for visual results. The surveillance-type videos in an MPEG-2 context show that there is a very little gap between 0 and 100% blocks for ME/MC. It can be seen that for high motion content videos (i.e. Foreman sequence, Fig. 9 (third row)) there is a significantly larger gap between 0% of blocks and a full frame motion estimation compared to low motion scenarios. When viewing the results in Table 3, the Garden sequence (camera pan) is the only high motion video (camera pan) that shows better results for the TSS implementation. Thus, video sequences with a camera pan should be examined in more detail as well. The block-selective scheme for the Foreman sequence achieves a significant improvement over the TSS case. (Please refer to Table 3 for detailed results.) Table 3 shows the overall results — how much quality can be yielded with 100% coding time TSS. A positive value means that TSS performs better in this case. For high motion content videos, the block-selective approach

performs better in three of four individual cases and on average, whereas the selective approach yields few desirable results for the rest of the videos (no plateaus, no gain in quality).

4.2 Frame-selective ME/MC

4.2.1 Frame-selective DJPEG2000: Our next experiment compares the block-selective and the frame-selective approach (both using TSS). In the following, Fig. 10 shows the results (not all figures could be included, because of formatting reasons): the frame-selective approach computes at least one motion-compensated differential frame, and thus starts off with a higher average PSNR quality than the selective approach (with zero blocks used in the ME/MC process). This is visible in Fig. 10 for the Carfromlefttoright sequence. There is a difference of; 0.3 dB, with no blocks for the selective case an

Table 3 Selective MPEG-2: TSS against block-selective FS ME/MC

selective MPEG-2						
Compression rate	20		100		150	
Results in PSNR, dB	TSS	Block	TSS	Block	TSS	Block
TSS against block-selective FS ME/MC MPEG-2						
Carfromlefttoright	31.5	31.55	24.45	24.35	22.15	22.1
Cycle	31.9	31.85	24.35	24.3	21.8	21.85
Rollin	36.1	36.45	33.1	33.25	30.1	30.2
Leavincar	35.25	35.35	26.7	27.9	23.75	23.9
Pillow	33.6	33.75	28.5	28.65	26.1	26.15
average of surveillance	33.6	33.8	27.4	27.7	24.8	24.9
Garden	26.1	26.2	19.95	19.95	19.75	19.85
Foreman	30.1	29.8	25.6	25.7	24.2	24.35
Paris	34.05	34.5	23.3	23.35	22.95	22.95

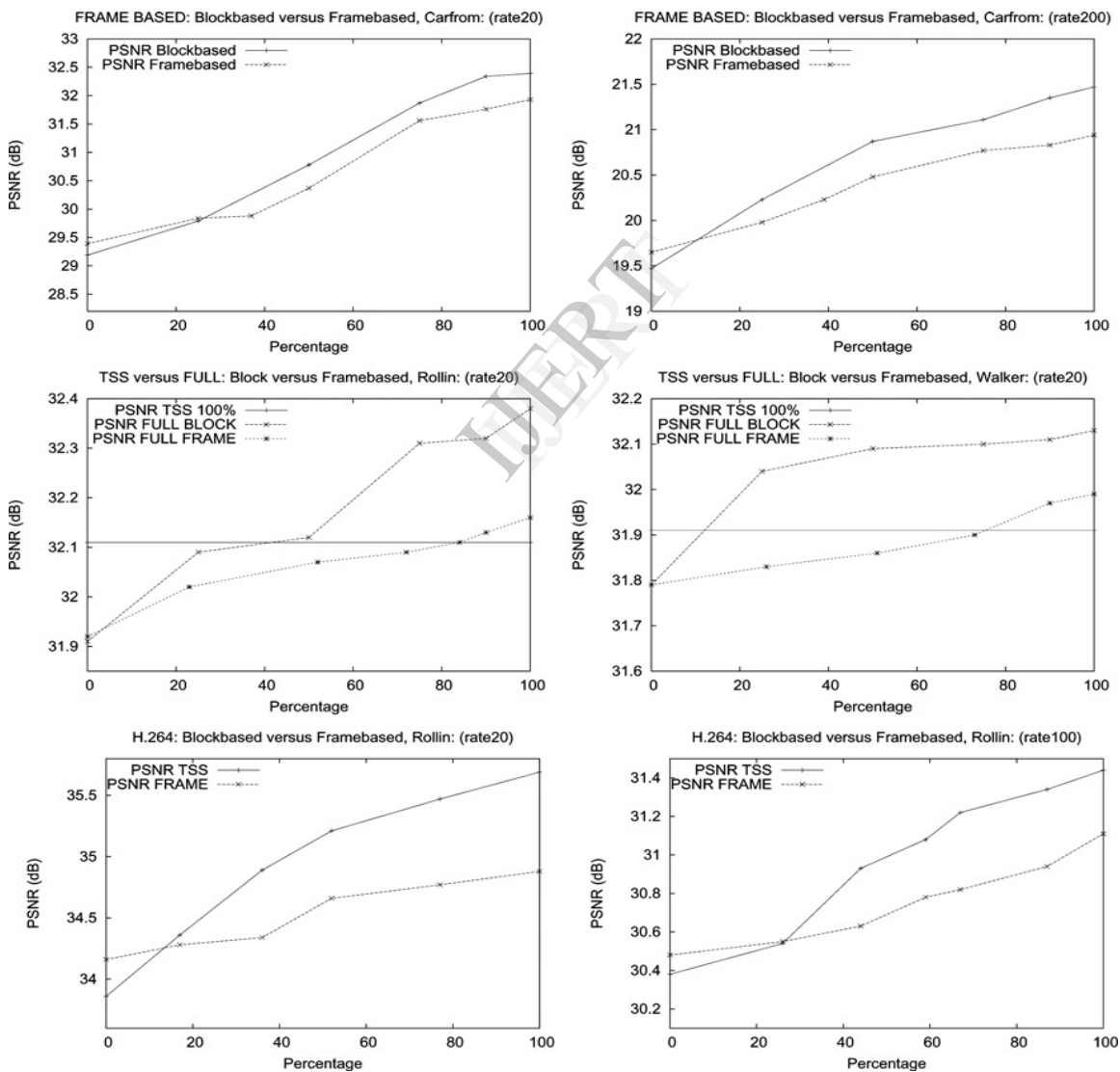


Fig. 10 Block-selective against frame-selective for Carfromlefttoright, Walker and Rollin for DJPEG2000 against H.264: compression ratios 20 and 200

Table 4 Block-selective H.264 against frame-selective H.264

H.264						
Compression rate	20		100		150	
PSNR, dB	H.264		H.264		H.264	
Video	Block	Frame	Block	Frame	Block	Frame
Carfromlefttoright	32.3	31.4	24.25	23.8	22.75	22.45
Cycle	32.2	31.8	24.7	24.3	22	21.7
Pillow	37	36.3	33.1	32.8	30.5	30
Leavincar	36.25	35.4	28.35	27.7	24.25	23.8
Rollin	34.1	33.2	30	29.5	26.5	23.2
Garden	27	26.3	20.7	19.9	20.2	19.9
Foreman	30.4	29.5	26.3	25.6	24.75	24.4
Paris	35.35	35	23.2	23	23.3	23.1

Table 5 Block-selective MPEG-2 against frame-selective MPEG-2

MPEG-2						
Compression rate	20		100		150	
PSNR, dB	MPEG-2		MPEG-2		MPEG-2	
Video	Block	Frame	Block	Frame	Block	Frame
Carfromlefttoright	31.8	31.4	24.65	24.2	22.65	22.4
Cycle	32.1	31.8	24.6	24.3	21.95	21.5
Pillow	36.8	36.3	33.45	33	30.4	30.1
Leavincar	35.7	35.4	28.15	27.7	24.15	23.8
Rollin	33.95	33.2	28.8	28.5	26.4	26.2
Garden	26.75	26.3	20.35	19.9	20.1	19.9
Foreman	30.3	29.5	25.9	25.6	24.65	24.4
Paris	35.15	34.9	23.6	23.6	23.2	23.1

frame-selective (one motion-compensated frame). At about 50% of blocks used for the block-selective scenario, the block-selective approach 'overtakes' the frame-selective approach. The frame-based approach always codes at least one differential frame without ME/MC. Therefore the maximum PSNR of both approaches is not truly comparable.

Finally, Fig. 10 (second row) draws a comparison between the block-selective and frame-selective in FULL ME mode and TSS with 100%. It shows how many blocks for the block-selective approach (respectively, the frame-selective approach) are required to achieve the same quality as TSS with equal complexity.

4.2.2 Frame-selective H.264, MPEG-2: In addition, we provide frame-selective results for MPEG-2 and H.264. We provide one visual example (Fig. 9, whereas the rest are shown in Table 4 (H.264) and Table 5 (MPEG-2)). This table shows the difference between the block-selective approach at 100% of computational load and the frame-selective approach at 100% (measured in PSNR). It is visible that a significant PSNR difference (up to 0.9 dB) has been reached; however, for surveillance-type videos the difference (especially for higher compression ratios) becomes insignificant.

5 Conclusion and future work

We have proposed a block/frame-selective ME/MC approach in the context of an interframe low-complexity JPEG 2000

video coding system specially developed for surveillance-type videos. Our system has been able to significantly improve the PSNR quality of the baseline system at a moderate computational cost compared to standard ME/MC algorithms (TSS). Block-selective ME/MC outperforms its frame-based counterpart and reaches the quality of an approach with full ME/MC with about 70–80% of the blocks in our simulations. In addition, we have shown that our block-selective scheme achieves higher visual image quality if complexity is fixed at an equal level when compared with TSS. In addition, we integrated this block-selective scheme to H.264 and MPEG-2 in order to observe how our approach works in other frameworks. The block-selective approach leads to similar results in different coding environments and, owing to its ease of implementation, constitutes an excellent alternative to commonly used fast ME/MC algorithms in low-complexity environments. Another advantage of our system is its adaptiveness, which allows the ME/MC preferences to be set as well as the best frame type for coding to be chosen according to the motion content of the frame. Hardware components are cheaper than H.264 and MPEG-2 video coding solutions. In future work, we will enhance the threshold finding process of the baseline system by improving our system to adapt to rate and quality dynamically and by integrating more types of motion-compensated coding (diamond search, four-step search) to be able to better cope with high motion content videos, and a realisation in hardware.

6 References

- 1 Chao, H., Wei, M.: 'Rate scalable video coding technology based on flexible block wavelet', J. Comput. Appl. Math., 2004, 163, (1), pp. 91–100
- 2 Cui, S., Wang, Y., Fowler, J.: 'Mesh-based motion estimation and compensation in the wavelet domain'. IEEE Int. Conf. Image Processing ICIP'00, Vancouver, Canada, 2002, vol 1, pp. 693–696
- 3 Cycon, H.L., Palkow, M., Schmidt, T.C., Wahlisch, M., Marpe, D.: 'Fast wavelet-based video codec and its application in an IP version 6-ready serverless videoconferencing system', Int. J. Wavelets Multiresolution Inf. Process. (IJWMIP), 2004, 2, (2), pp. 165–171
- 4 Flierl, M., Girod, B.: 'Video coding with motion-compensated lifted wavelet transforms', Signal Process., Image Commun., 2004, ISSN 19, (7), pp. 561–575
- 5 Morris, T., Britch, D.: 'Intra-frame wavelet video coding'. Second Int. Symp. Video Processing and Multimedia Communications ELMAR, Zadar, Croatia, 28–30 June 2000, pp. 1–4
- 6 Morris, T., Britch, D.: 'Object-based intra-frame wavelet video coding'. Second IEEE R8-EURASIP Symp. Image and Signal Processing and Analysis (ISPA01), Pula, Croatia, 19–21 June, 2001, pp. 599–603
- 7 Schuchter, A., Uhl, A.: 'Low cost JPEG2000 based video coding system', in He, Y., Ostermann, J., Lee, S., Chen, J. (Eds.): PCS06, Proc. Picture Coding Symp. PCS, China, April 2006, Convention Center Beijing, (PCS)
- 8 Sun Young Lee, W.K., Cho, Y.H., Jang, E.S.: 'Selective motion estimation for fast video encoding'. Advances in Multimedia Information Processing – PCM 2004, Korea, 2004, (LNCS, 3333), pp. 133–179
- 9 Schuchter, A., Uhl, A.: 'Block selective motion estimation for low-complexity video coding'. DSP 2009, Santorini, Greece, 2009, pp. 1–5
- 10 Adams, M.D., Jasper, F.K.: 'A software-based JPEG-2000 codec implementation' (Vancouver, Canada, September 2000, vol. 2), pp. 53–56
- 11 Adams, M.: 'The JPEG-2000 still image compression standard. ISO/IEC JTC 1/SC 29/WG 1 N 2412', September 2011
- 12 Bovik, A. (Ed.): 'Handbook of image and video processing', in 'Communications, Networking, and Multimedia' (Academic Press, 2000, 1stedn)

IJERT