Fast Multiple Reference Frame Motion Estimation For H.264/AVC

Yeping Su, *Member, IEEE*, and Ming-Ting Sun, *Fellow, IEEE*

***Abstract*—Multiple reference frame motion compensation is a new feature introduced in H.264/MPEG-4 AVC to improve video coding performance. However, the computational cost of Multiple Reference Frame Motion Estimation (MRF-ME) is very high. In this paper, we propose an algorithm that takes into account the correlation/continuity of motion vectors among different reference frames. We show that the algorithm effectively reduces the computations of MRF-ME, and achieves similar coding gain compared to the motion search approaches in the reference software.**

***Index Terms*—H.264/AVC, multiple reference frames, motion estimation.**

I. Introduction

H.264/MPEG-4 AVC is the newest international video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group [1]. It represents the state-of-the-art video compression technology, and addresses the full range of video applications including low bit-rate wireless video applications, standard-definition & high-definition broadcast television, and video streaming over the Internet. In terms of compression performance, it provides more than 50% bit-rate savings for equivalent video quality relative to the performance of MPEG-2 video coding standard. To achieve such a high coding efficiency, AVC includes many new features such as variable blocksize motion compensation, quarter-pixel accuracy motion compensation, and multiple reference frame motion compensation.

In the variable blocksize motion compensation, AVC supports luma block-sizes of 16x16, 16x8, 8x16, and 8x8 in the inter-frame prediction. In case 8x8 is chosen, further smaller block-sizes of 8x4, 4x8, and 4x4 can be used.

In the multiple reference frame motion compensation, a signal block with uni-prediction in P slices is predicted from one reference picture out of a large number of decoded pictures. And similarly, a motion compensated bi-prediction block in B slices is predicted from two reference pictures, both can be chosen out of their candidate reference picture lists. A scenario of Multiple Reference Frame Motion Estimation is shown in Figure 1. It is an effective technique to improve the coding efficiency [2]. However, MRF-ME dramatically increases the computational complexity of the encoders because the Motion Estimation (ME) process needs to be performed for each of the reference frames. Considering motion estimation is the most computationally intensive functional block in the video codec, this increased complexity penalizes the benefit gained from the better coding efficiency, and thus may restrict its applicability.

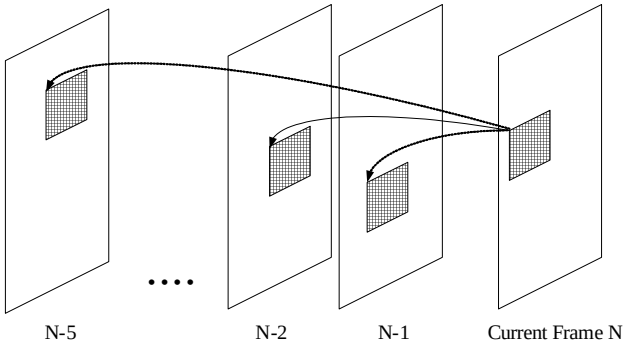


Figure 1. Multiple reference frame motion estimation

The reference software of AVC JM 8.6 [3] performs the motion estimation for all block-sizes across all reference frames in the encoder. In [4], a fast algorithm is proposed to speed-up the MRF-ME by considering the different sub-pixel sampling position of each block, and performing ME on the selected reference frames with similarly sampled contents. In [5][6], several heuristics are used to decide whether it is necessary to search more than the most recent reference frame, and hence reduce the computations. In [7], a fast multi-frame motion estimation algorithm based on Motion Vector (MV) reusing similar to our basic ideas described in [8][9] is independently proposed. The motion vector composition in [7] is done by choosing a dominant MV, and 5~7 checking points are needed to refine the composed MV. The proposed multi-frame motion estimation method in this paper differs from [7] in using a weighted average for motion composition, and there is no further refinement needed.

In this paper, we first investigate why multiple reference frames provide better predictions, based on the observations from experiments on standard test sequences. We then propose a fast MRF-ME algorithm, which can achieve nearly the same coding efficiency as the search approaches in the

Manuscript received October 20, 2004; revised January 8, 2005. This paper was recommended by Associate Editor C. Guillemot.

Y. Su is with Thomson Corporate Research Lab, Princeton NJ 08540 USA (e-mail: yeping.su@thomson.net).

M.-T. Sun is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: sun@ee.washington.edu).

Digital Object Identifier ****

reference software, but cut down the computations significantly.

II. Analysis of Multiple Reference Frame Motion Estimation

A. Multiple Reference Frame Motion Estimation Review Stage

In the AVC reference encoder [3], the MV search for a given block referring to a given reference frame returns the MV $\overrightarrow{mv}$ that minimizes a cost function $J(\overrightarrow{mv}, \lambda_{MOTION}) = D(s, c(\overrightarrow{mv})) + \lambda_{MOTION} \cdot R(\overrightarrow{mv} - \overrightarrow{mv}_p)$, where $\overrightarrow{mv}_p$ being the prediction for the MV, and λ being the Lagrange multiplier. Here the arrowed symbol $\overrightarrow{mv}$ denotes a motion vector with horizontal and vertical components.

In AVC reference software JM8.6 [3], exhaustive motion search algorithm and a fast motion search algorithm are supported. In both cases, the MV search process is the most computationally intensive part in video encoders. When multiple reference frames are used, the same search process is applied to each reference frame. Thus the amount of computations increases linearly with the number of reference frames.

B. Why Do Multiple Reference Frame Help Predictions?

There are many reasons for the MRF-ME to achieve better predictions than those using just single reference picture. Often cited reasons include [5]:

1) Repetitive motions. Due to the repetitive nature of the motion, there are better appearances of the same object/texture several frames ago.

2) Uncovered background. Some parts of the picture may originally be covered by a moving object. As the object moves, the uncovered backgrounds may not find a good match from the previous frame, but may be able to find a good match from several frames ago when they were also uncovered.

3) Alternating camera angles that switch back and forth between two different scenes.

Besides these reasons, based on our observations from experiments on standard video sequences, there are several other reasons why MRF-ME performs better than single reference frame motion estimation:

4) Sampling. When an object moves with a non-integer pixel displacement, the sampling positions of the object in different frames may be different. Due to this different sampling, the current block may get a better match to a block in more previous reference frames. This phenomenon is also addressed in [4].

5) Shadow and lighting changes. An area or a moving object may not have exactly the same pixel values as those at the previous locations in the previous frame since they may have different shadowing, lighting conditions, or reflections.

6) Camera shaking, such as the last part of the Foreman sequence. When a camera is moving up and down, the current frame may better resemble a frame appeared several frames ago. This reason can also be attributed to sampling effects.

7) Noises in the source signal produced by the camera and other factors. Even in the stationary areas of the picture, some blocks may find a better match in more previous reference pictures, which happens in many sequences.

In practice, the situations of 4)-7) actually occur quite often, and we found that they are the dominant reasons for the advantages of using MRF in standard video sequences. Due to their similar impact on the video signal, we summarize these reasons in a term "noise effect". An example is the moving calendar in the MobileCalendar sequence, with a zoom-in view of the current block to be coded and the best matched blocks in the previous two reference frames shown in Figure 2. The ResidualBlock(N-1) has MSE=203.8, versus MSE=32.4 for ResidualBlock(N-2) although they were referenced to the same object position in the corresponding reference frames.

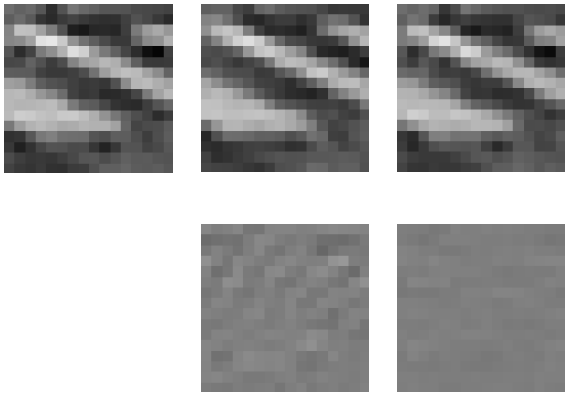


Figure 2. Example of sampling/noise effect

Under these situations with noise effects, there are strong correlations among the motion vector fields with multiple reference frames, which are discussed in the next section.

C. Correlations in Multiple Reference Picture Motion Vectors

The continuity of the motion can be explored in order to facilitate the motion estimation across multiple reference frames.

Assume a part/block of object is moving in an image sequence and keeps the similar appearance in adjacent frames. The continuity of this block moving across images will result in strong correlation among the motion vector fields in multiple reference frames, which can be expressed simply as:

$$\overrightarrow{MV}_n^{-k} \approx \overrightarrow{MV}_n^{-k_1} + \overrightarrow{MV}_{n-k_1}^{-(k-k_1)} \quad (1)$$

which is shown in Figure 3. In (1) $\overrightarrow{MV}_n^{-k}$ represents the motion vector of Frame *n* referring to Frame *(n-k)*, which is called *k*-step MV.

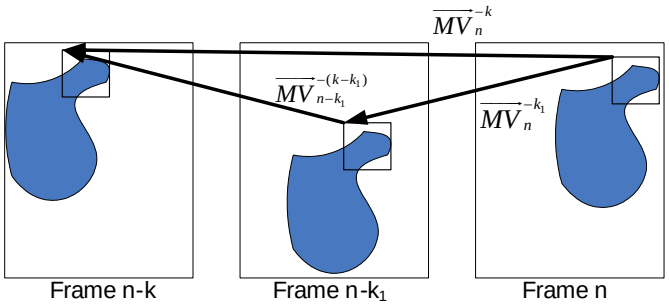


Figure 3. Illustration of motion continuity

We try to use this correlation to save the computation of the MRF-ME process, i.e., to compose the motion vector estimate $\widehat{\overrightarrow{MV}}_n^{-k}$ by combining $\overrightarrow{MV}_n^{-k_1}$ and $\overrightarrow{MV}_{n-k_1}^{-(k-k_1)}$, or to perform ME only along the motion trajectories. In this paper, a *k*-step MV $\overrightarrow{MV}_n^{-k}$ is composed by combining *k* 1-step MV's recursively, i.e., for *l=2:k*,

$$\widehat{\overrightarrow{MV}}_n^{-l} = \widehat{\overrightarrow{MV}}_n^{-(l-1)} + \overrightarrow{MV}_{n-(l-1)}^{-1} \quad (2)$$

In order to verify the strong MV correlations across multiple reference frames, simulations are performed to compare the true MV's $\overrightarrow{MV}_n^{-k}$ using the motion search in the reference software and MV estimates $\widehat{\overrightarrow{MV}}_n^{-k}$ from motion compositions. The details of motion composition are described in Section III. Motion Composition Error (MCE) is used as the difference measure, which is the *L-1* distance between the true and the composed motion vectors:

$$\begin{aligned} MCE &= \left| \overrightarrow{MV}_n^{-k} - \widehat{\overrightarrow{MV}}_n^{-k} \right|_{L1} \\ &= \left| \overrightarrow{MV}_n^{-k}(x) - \widehat{\overrightarrow{MV}}_n^{-k}(x) \right| + \left| \overrightarrow{MV}_n^{-k}(y) - \widehat{\overrightarrow{MV}}_n^{-k}(y) \right| \end{aligned} \quad (3)$$

Table I lists the percentage of times when MCE is no greater than a threshold *d* (in pixel). The statistics are collected from encoding the whole Mobile sequence (CIF size and 300 frames) with different *k*. All experiments are conducted with fixed *Qp=20* (high bitrate scenario), and exhaustive search is used as motion search method. The high bit-rate scenario is of more interest since the MRF-ME is more important for applications where video quality is more important than computational complexity.

TABLE I
MOTION COMPOSITION ERROR

Sequence	*d*=0	*d*=1	*d*=2	*d*=3
Mobile, *k*=2	81%	92%	95%	96%
Mobile, *k*=3	80%	89%	92%	94%
Mobile, *k*=4	78%	87%	90%	92%

Clearly, the motion composition gives good estimates for the real motions, with majority of them within a small spatial neighborhood of $\overrightarrow{MV}_n^{-k}$. Because the composition of motions are based on the continuity assumption, this also shows that d)-g) in the previous section are the dominant reasons in the advantage of enabling MRF in those sequences.

There are rare cases the motion vector continuity assumption may fail, for example on object boundaries where covering/uncovering happens and MV's are unreliable. To handle these rare cases, in our proposed algorithm to be described in detail later, we incorporate a simple boundary macroblock detector and perform processing adaptively. We found that in most situations, our algorithm performs well even without the special handling of the boundary macroblocks.

III. MOTION VECTOR COMPOSITION FOR VARIABLE BLOCK-SIZE ME

AVC allows seven different block-sizes. To compose the motion vector for different block-sizes, all MV's are stored in the 4x4 blocks, which is the smallest common unit in the variable block-size ME.

A. Motion Composition

The motion composition process described in this section applies on a block *B* with the size $s \times t \in \{16\times16, 16\times8, 8\times16, 8\times8, 8\times4, 4\times8, 4\times4\}$. The inputs of the motion composition process are two 4x4 block-based motion vector fields: $\widehat{\overrightarrow{\mathbf{MV}}}_n^{-(l-1)}$ and $\overrightarrow{\mathbf{MV}}_{n-(l-1)}^{-1}$. The output is the composed motion vector $\widehat{\overrightarrow{MV}}_n^{-l}$ for *B*. Note here that the bold symbol $\overrightarrow{\mathbf{MV}}$ represents the motion vector field for the whole frame, not a single motion vector.

The motion composition has two parts: motion concatenation and weighted average estimate:

1) Motion concatenation:

Collect all block motion vectors $\overrightarrow{mv}_i \in \widehat{\overrightarrow{\mathbf{MV}}}_n^{-(l-1)}$ covered by current block *B*, we have $\frac{s \cdot t}{16}$ possibly different MV's, one for each 4x4 block. Each such MV $\overrightarrow{mv}_i$ will point to a 4x4 area in the frame *(n-l+1)*, but generally will not align with block boundaries. Thus each $\overrightarrow{mv}_i$ will usually refer to 4 neighboring 4x4 blocks in the frame *(n-i+1)*, which in turn covers 4 $\overrightarrow{mv}_j \in \widehat{\overrightarrow{\mathbf{MV}}}_{n-(l-1)}^{-1}$. For each $\overrightarrow{mv}_i$ and its associated $\overrightarrow{mv}_j$, the overlapping area is denoted as w_{ij}. Adding $\overrightarrow{mv}_i$'s and $\overrightarrow{mv}_j$'s: $\overrightarrow{mv}_{ij} = \overrightarrow{mv}_i + \overrightarrow{mv}_j$, we have a set of candidate MV's with their corresponding overlapping areas $S = \{\overrightarrow{mv}_{ij}, w_{ij}\}$.

An example of this motion concatenation is shown in Figure 4. The current block has size 4x8, which covers two 4x4 blocks. There are four referred blocks in the frame *(n-l+1)* for one $\overrightarrow{mv}_i$, each with possibly different overlapping

areas w_{ij} and motion vector $\overrightarrow{mv}_j$. Only one concatenation is shown for clearer illustrations.

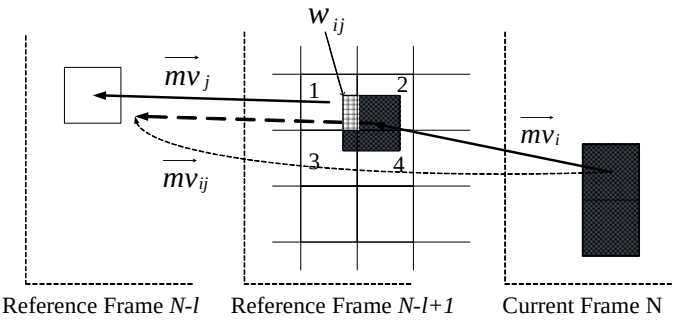


Figure 4. Illustration of the MV concatenation

2) Weighted average estimation:

After the motion concatenation process, a weighted average vector $\overrightarrow{mv}_c$ is formed from the set S, treating w_{ij}'s as weights: $\overrightarrow{mv}_c = \frac{\sum w_{ij} \cdot \overrightarrow{mv}_{ij}}{\sum w_{ij}}$, where $\overrightarrow{mv}_c$ is the output of the motion composition process.

The motion flow across multiple frames is captured effectively by the motion concatenation in step *1)*. The weighted average in step *2)* summarizes the dispersive output of motion concatenation, providing a robust and computational efficient estimation.

B. Summary of the Proposed Algorithm

The process of the proposed fast MRP-ME algorithm for the *n*-th frame is summarized as follows:

Step 1) First motion estimation referring to frame *(n-1)* is performed, which results in $\overrightarrow{\mathbf{MV}}_n^{-1}$. A motion dispersion measure is computed for each Macroblock (MB). The dispersion measure is defined as the sum of absolute difference between any horizontal pair and vertical pair of 4x4 MV's within the current MB. If the dispersion exceeds certain threshold, this MB is declared as a boundary block and the 4x4-blocksize motion search is performed as in the reference software for further reference frames, otherwise, the search process is described in step 2). The threshold is set to 32 in later simulations.

Step 2) For *l* from 2 to *k*, for each MB:

Step 2.1) For each block of each blocksize, motion composition process as defined in Section III is invoked with $\overrightarrow{mv}_c$ as output. $\overrightarrow{mv}_c$ and the prediction MV $\overrightarrow{mv}_p$ are compared using the cost function *J*, and the winner is the final output. After that, sub-pixel motion search is conducted.

Step 2.2) Perform the block-size mode decision among variable blocksizes using motion search costs, and a 4x4 based MV field $\overrightarrow{\mathbf{MV}}_n^{-l}$ is obtained for later processing.

In Step 1), any single reference frame fast ME algorithm can be applied to obtain $\overrightarrow{\mathbf{MV}}_n^{-1}$. Also, the proposed algorithm does not assume that MV's of all variable blocksizes are available, which is the case for many fast AVC ME algorithms such as [10].

In step 2.1), both temporal and spatial correlations among MV's are considered: $\overrightarrow{mv}_p$ for spatial correlation and $\overrightarrow{mv}_c$ for temporal correlation.

Due to the variable blocksize ME in AVC, there are MV's associated with different blocksizes for the same MB. Since the proposed algorithm only assumes one 4x4-based MV field representation in the motion composition step, mode decisions among variable blocksizes must be performed for each reference frame, which is done in step 2.2).

C. Complexity Analysis

Assuming the image size is *W*-by-*H* in pixel units. We need following additional memories to implement the motion compositions:

1) *k* motion vector fields storing one step motion vectors, each contains *W/4*H/4*2* integers

2) One temporary motion vector field as we traverse the reference frames from 1 to *k*, with size *W/4*H/4*2* integers.

Clearly the additional memories needed are very moderate.

For computational complexities, the majority of MB's are non-boundary. For non-boundary blocks, motion composition is conducted. In one motion composition process, only two check points are needed in the ME: one from MV composition $\overrightarrow{mv}_c$ and one from prediction MV $\overrightarrow{mv}_p$.

IV. Simulation Results

The proposed MRF-ME algorithm was tested on several standard video sequences which show significant gains when MRF-ME is enabled: Mobile.cif (300 frames), Foreman.cif (300 frames), Tempete.cif (260 frames) and Carphone.qcif (382 frames).

The scheme is implemented based on the reference software JM8.6 [3]. Some common simulation settings are listed as follows:

- Microsoft® Windows platform, P4 2.0G CPU
- Compiled using Visual Studio® 6, Release mode
- Using RDO_Off in mode decisions
- SearchRange = ±32
- P frames coding only, with the first I-frame
- No rate control
- Peak Signal to Noise Ratio (PSNR) of Luminance in dB is used as distortion measure

Also, both exhaustive search (FME_Off) and fast motion search (FME_On) in JM8.6 are tested and compared with our proposed algorithm. In each comparison, the first reference frame ME in the proposed algorithm is the same as in the JM8.6.

A. Coding Efficiency Results

The rate-distortion (R-D) coding performance comparisons are conducted for the following three test cases:

- 1 reference frame, JM8.6
- 5 reference frames, JM8.6
- 5 reference frames, proposed algorithm

Figure 5 and Figure 6 show the R-D plots using fixed $Q_p \in \{20, 25, 30, 35, 40\}$, for FME_Off and FME_On respectively. The proposed scheme performs almost the same as JM8.6 in all coding efficiency results.

B. Computational Efficiency Results

The total motion estimation runtime for the following two test cases:

- 5 reference frames, JM8.6
- 5 reference frames, proposed algorithm

are listed in Table II and Table III, for FME_Off and FME_On respectively. Three fixed Q_p's {20,30,40} are used for high, median, and low bitrates, respectively. The timings only include the integer pixel ME parts.

Assuming t_1 is the runtime of the ME algorithm for one reference frame in the JM8.6, and t_2 is the runtime of our proposed fast MRF-ME algorithm for each additional reference frame. The total runtime of the proposed algorithm is $T_{FastSearch} = t_1 + (k-1) \cdot t_2$ and the total runtime of JM8.6 is $T_{FullSearch} = k \cdot t_1$. The averaged speedup ratio r is defined as $r = T_{FullSearch} / T_{FastSearch}$, which is also listed in Table II and Table III for five reference frames, showing significant computational savings of the proposed algorithm comparing to JM8.6. To better visualize the computational saving, Figure 7 shows the relationship between the normalized runtime of MRF motion estimation module and the number of reference frames. The linear behaviors of the curves in Figure 7 with different slopes justify the advantage of the proposed approach.

V. Conclusion

In this paper, a novel multiple reference frame motion estimation algorithm is proposed. The proposed algorithm is based on the conceptually simple idea of tracing motion across frames. In the MRF-ME process, MVs are formed based on the motion trajectories and spatial MV predictions. Results show the scheme is very effective in reducing the computational cost comparing with both exhaustive search and fast motion search, while keeping good coding efficiency.

Acknowledgment

The authors thank the anonymous reviews for their careful reviews and useful suggestions that improved this paper.

References

[1] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2003.

[2] T. Wiegand et al, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 560-576, July 2003.

[3] Joint Video Team software, JM8.6. http://iphome.hhi.de/suehring/tml/download/

[4] A. Chang et al, "A Novel Approach to Fast Multi-Frame Selection for H.264 Video Coding," IEEE ICASSP 2003, Hong-Kong, April 2003

[5] Yu-Wen Huang et al, "Analysis and Reduction of Reference Frames for Motion Estimation in MPEG-4 AVC/JVT/H.264," IEEE ICASSP 2003, Hong-Kong, April 2003

[6] B. -Y. Hsieh et al, "Fast motion estimation algorithm for H.264/MPEG-4 AVC by using multiple reference frame skipping criteria," SPIE VCIP 2003, Lausanne, Switzerland, July 2003

[7] Mei-Juan Chen et al, "Efficient Multi-Frame Motion Estimation Algorithms for MPEG-4 AVC/JVT/H.264," IEEE ISCAS 2004, Vancouver, Canada, May 2004

[8] Jeongnam Youn and Ming-Ting Sun, "A Fast Motion Vector Composition Method For Temporal Transcoding," IEEE ISCAS 1999, Orlando, Florida, June 1999.

[9] Yeping Su and Ming-Ting Sun, "Fast Multiple Reference Frame Motion Estimation for H.264," IEEE ICME 2004, Taipei, Taiwan, June 2004.

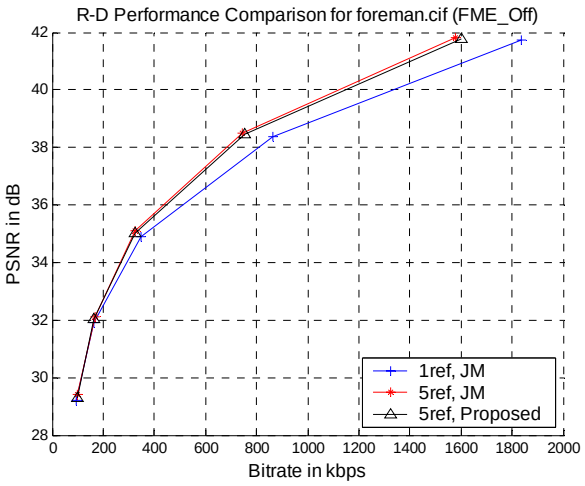
[10] Zhi Zhou, Ming-Ting Sun and Spencer Hsu, "Fast Variable Block-Size Motion Estimation Based on Merging and Splitting Procedures for H.264/MPEG-4 AVC," IEEE ISCAS 2004, Vancouver, Canada, May 2004.

TABLE II
COMPUTATIONAL COMPARISON WITH FME_OFF

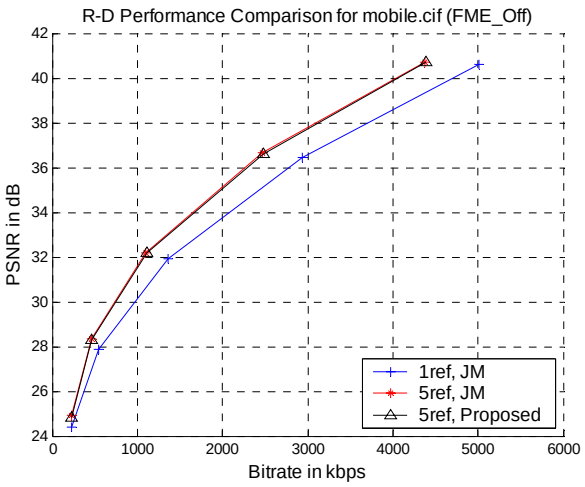
	Carphone.qcif			Mobile.cif		
	Q_p = 20	Q_p = 30	Q_p = 40	Q_p = 20	Q_p = 30	Q_p = 40
Ref5_Orig	351.0s	360.2s	374.2s	4005.3s	4048.1s	4181.6s
Ref5_New	110.3s	103.9s	95.8s	1096.7s	1011.2s	930.8s
r	3.18	3.47	3.90	3.65	4.00	4.49

TABLE III
COMPUTATIONAL COMPARISON WITH FME_ON

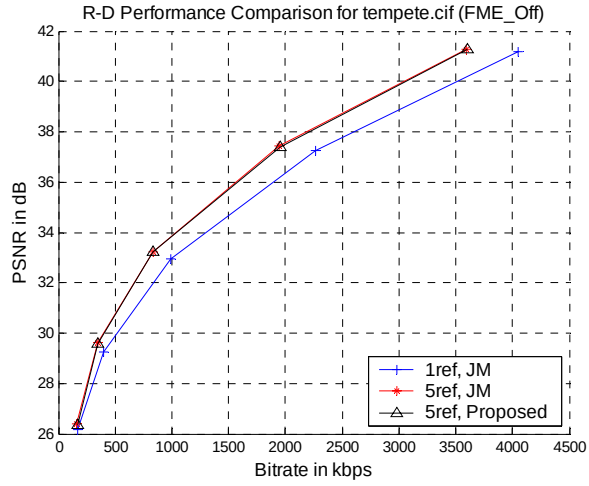
	Carphone.qcif			Mobile.cif		
	Q_p = 20	Q_p = 30	Q_p = 40	Q_p = 20	Q_p = 30	Q_p = 40
Ref5_Orig	59.6s	55.8s	50.1s	352.6s	354.2s	385.0s
Ref5_New	30.8s	28.7s	28.3s	131.6s	133.5s	142.4s
r	1.94	1.94	1.77	2.68	2.65	2.70



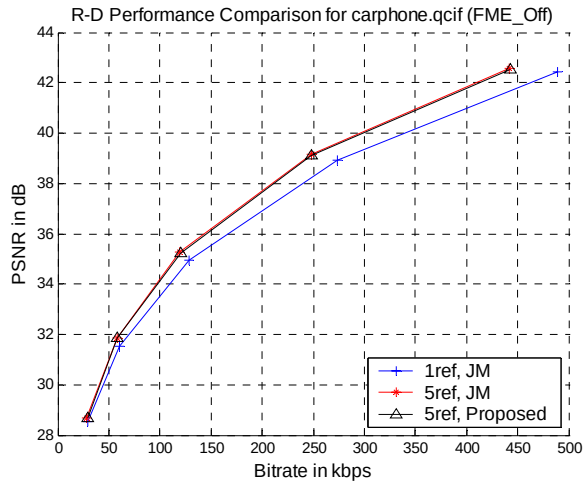
(a)



(b)

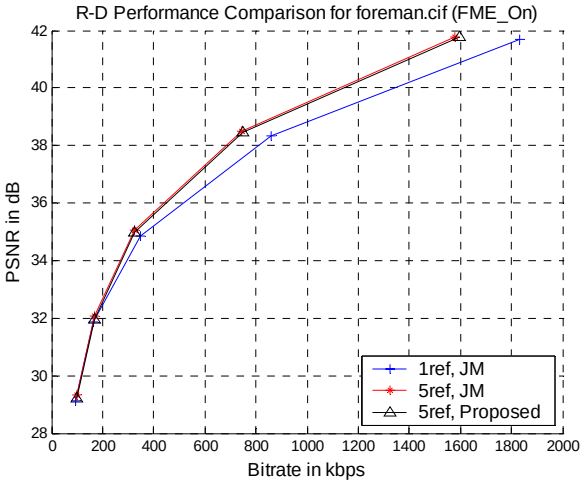


(c)

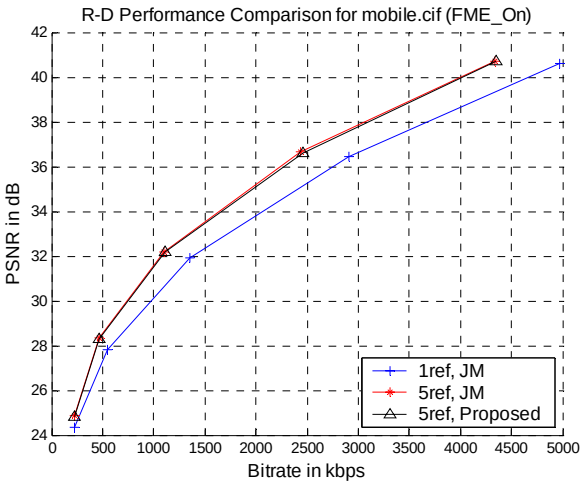


(d)

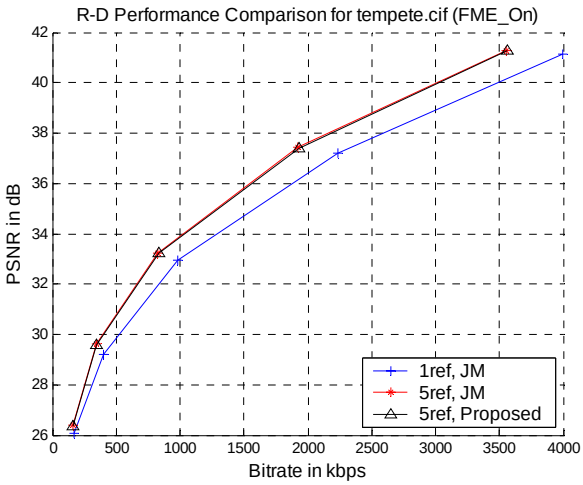
Figure 5. Rate-Distortion efficiency comparisons with FME_Off for (a) Foreman, (b) Mobile, (c) Tempete, (d) Carphone



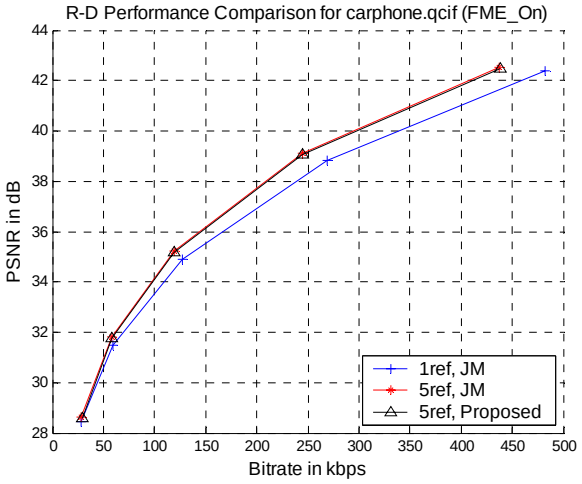
(a)



(b)

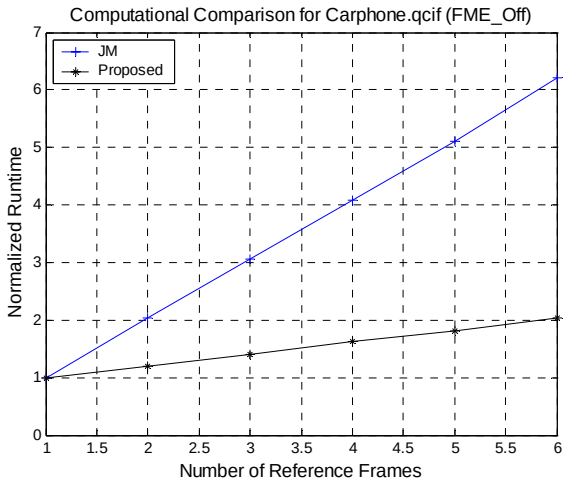


(c)

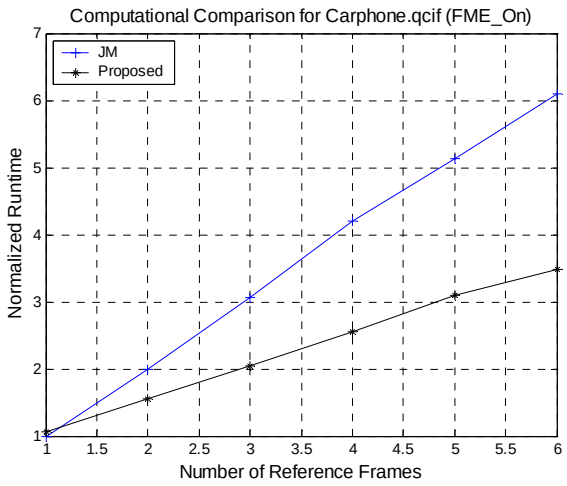


(d)

Figure 6. Rate-Distortion efficiency comparisons with FME_On for (a) Foreman, (b) Mobile, (c) Tempete, (d) Carphone



(a)



(b)

Figure 7. Computational comparisons with different number of reference frames for Carphone sequence (a) FME_Off (b) FME_On