Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC

Yu-Wen Huang, Bing-Yu Hsieh, Shao-Yi Chien, *Member*, Shyh-Yih Ma, and Liang-Gee Chen, *Fellow*

***Abstract*—In the new video coding standard H.264/AVC, motion estimation (ME) is allowed to search multiple reference frames. Therefore, the required computation is highly increased, and it is in proportion to the number of searched reference frames. However, the reduction in prediction residues is mostly dependent on the nature of sequences, not on the number of searched frames. Sometimes the prediction residues can be greatly reduced, but frequently a lot of computation is wasted without achieving any better coding performance. In this paper, we propose a context-based adaptive method to speed up the multiple reference frames ME. Statistical analysis is first applied to the available information for each macroblock (MB) after intra-prediction and inter-prediction from the previous frame. Context-based adaptive criteria are then derived to determine whether it is necessary to search more reference frames. The reference frame selection criteria are related to selected MB modes, inter-prediction residues, intra-prediction residues, motion vectors of subpartitioned blocks, and quantization parameters. Many available standard video sequences are tested as examples. The simulation results show that the proposed algorithm can maintain competitively the same video quality as exhaustive search of multiple reference frames. Meanwhile, 76%–96% of computation for searching unnecessary reference frames can be avoided. Moreover, our fast reference frame selection is orthogonal to conventional fast block matching algorithms, and they can be easily combined to achieve further efficient implementations.**

***Index Terms*—ISO/IEC 14496-10 AVC, ITU-T Rec. H.264, JVT, motion estimation (ME), multiple reference frames.**

I. Introduction

EXPERTS from ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) formed the Joint Video Team (JVT) in 2001 to develop a new video coding standard, H.264/AVC [1]. Compared with MPEG-4 [2], H.263 [3], and MPEG-2 [4], the new standard can achieve 39%, 49%, and 64% of bit-rate reduction, respectively [5]. The functional blocks of H.264/AVC, as well as their features, are shown in Fig. 1. Like previous standards, H.264/AVC still uses motion compensated transform coding. The improvement in coding performance comes mainly from the prediction part. Motion estimation (ME) at quarter-pixel accuracy with variable block sizes and multiple reference frames greatly reduces the prediction errors. Even if inter-frame prediction cannot find a good match, intra-prediction will make it up instead of directly coding the texture as before.

The reference software of H.264/AVC, JM [6], adopts full search for both motion estimation (ME) and intra-prediction. The instruction profile of the reference software on Sun Blade 1000 with UltraSPARC III 1 GHz CPU shows that real-time encoding of CIF 30 Hz video (baseline options, search range $[-16.75, +16.75]$, five reference frames) requires 314 994 million instructions per second and memory access of 471 299 Mbytes/s. The run time percentage of each function is shown in Fig. 2. Apparently, ME is the most computationally intensive part. In H.264/AVC, although there are seven kinds of block size (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) for motion compensation (MC), the complexity of ME in the reference software is not seven times of that for one block size. The search range centers of the seven kinds of block size are all the same, so that the sum of absolute difference (SAD) of a 4×4 block can be reused for the SAD calculation of larger blocks. In this way, variable block size ME does not lead to much increase in computation. Intra-prediction allows four modes for 16×16 blocks and nine modes for 4×4 blocks. Its complexity can be estimated as the SAD calculation of 13 16×16 blocks plus extra operations for interpolation, which are relatively small compared with ME. As for the multiple reference frames ME, it contributes to the heaviest computational load. The required operations are proportional to the number of searched frames. Nevertheless, the decrease in prediction residues depends on the nature of sequences. Sometimes the prediction gain by searching more reference frames is very significant, but usually a lot of computation is wasted without any benefits.

In this paper, an effective method for accelerating the multiple reference frames ME without significant loss of video quality is proposed. The available information after intra-prediction and ME from the previous frame is first analyzed. Then we decide to keep on searching one more reference frame or to terminate the block matching process in the early stage. The rest of this paper is organized as follows. In Section II, the related background including the flow of macroblock (MB) mode decision in the reference software and our modifications is reviewed. In Section III, we first analyze the statistics of selected MB modes, inter-prediction residues, intra-prediction residues, motion vector (MVs) of subpartitioned blocks, and quantization parameter (QP). Then we propose our fast algorithm in Section IV. Simulation results will be shown in Section V. Finally, Section VI gives a conclusion.

Manuscript received May 22, 2003; revised July 10, 2005. This paper was recommended by Associate Editor S.-U. Lee.

Y.-W. Huang and B.-Y. Hsieh were with the Department of Electrical Engineering II, National Taiwan University, Taipei 10617, Taiwan, R.O.C. They are now with MediaTek, Inc., Hsinchu 300, Taiwan, R.O.C. (e-mail: yuwen@video.ee.ntu.edu.tw; BY_Hsieh@mtk.com.tw).

S.-Y. Chen and L.-G. Chen are with the Department of Electrical Engineering II, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: shoayi@video.ee.ntu.edu.tw; steve@vivotek.com; lgchen@video.ee.ntu.edu.tw).

S.-Y. Ma is with Vivotek Inc., Taiwan, R.O.C. (e-mail: steve@vivotek.com)

Digital Object Identifier 10.1109/TCSVT.2006.872783

1051-8215/$20.00 © 2006 IEEE

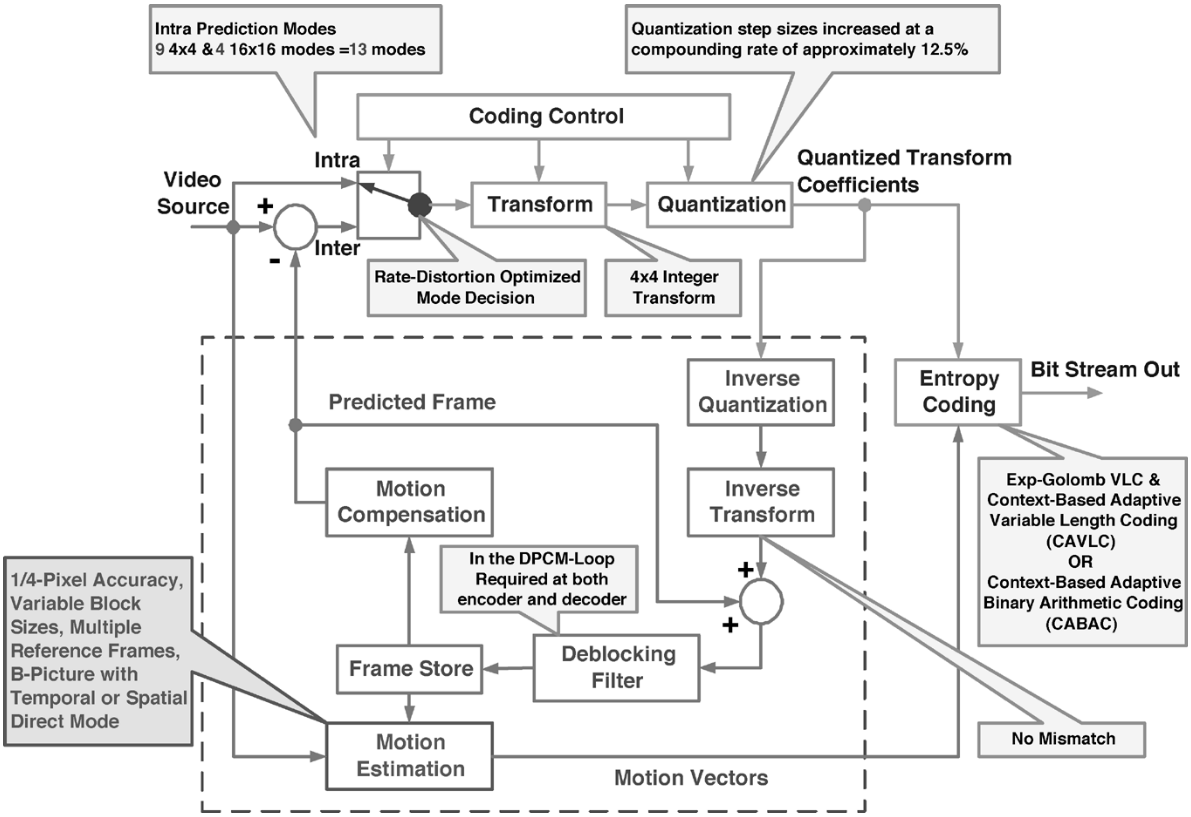

Intra Prediction Modes
9 4x4 & 4 16x16 modes =13 modes
Quantization step sizes increased at a compounding rate of approximately 12.5%
Coding Control
Video Source
Intra
+
Inter
-
Transform
Quantization
Quantized Transform Coefficients
Rate-Distortion Optimized Mode Decision
4x4 Integer Transform
Inverse Quantization
Entropy Coding
Bit Stream Out
Predicted Frame
Motion Compensation
Inverse Transform
Exp-Golomb VLC & Context-Based Adaptive Variable Length Coding (CAVLC) OR Context-Based Adaptive Binary Arithmetic Coding (CABAC)
In the DPCM-Loop Required at both encoder and decoder
1/4-Pixel Accuracy, Variable Block Sizes, Multiple Reference Frames, B-Picture with Temporal or Spatial Direct Mode
+
+
Frame Store
Deblocking Filter
Motion Estimation
Motion Vectors
No Mismatch

Fig. 1. Functional blocks and features of H.264.

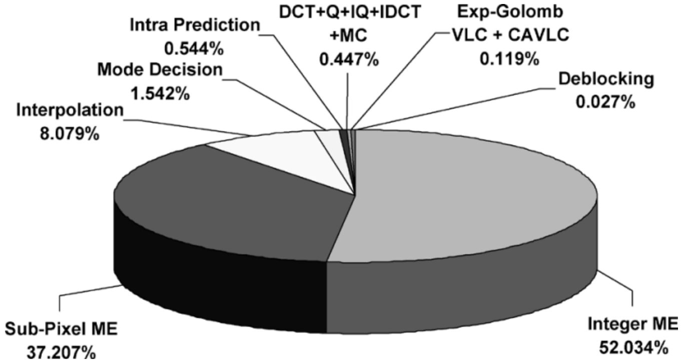


Fig. 2. Runtime percentages of functional blocks in H.264/AVC baseline encoder.

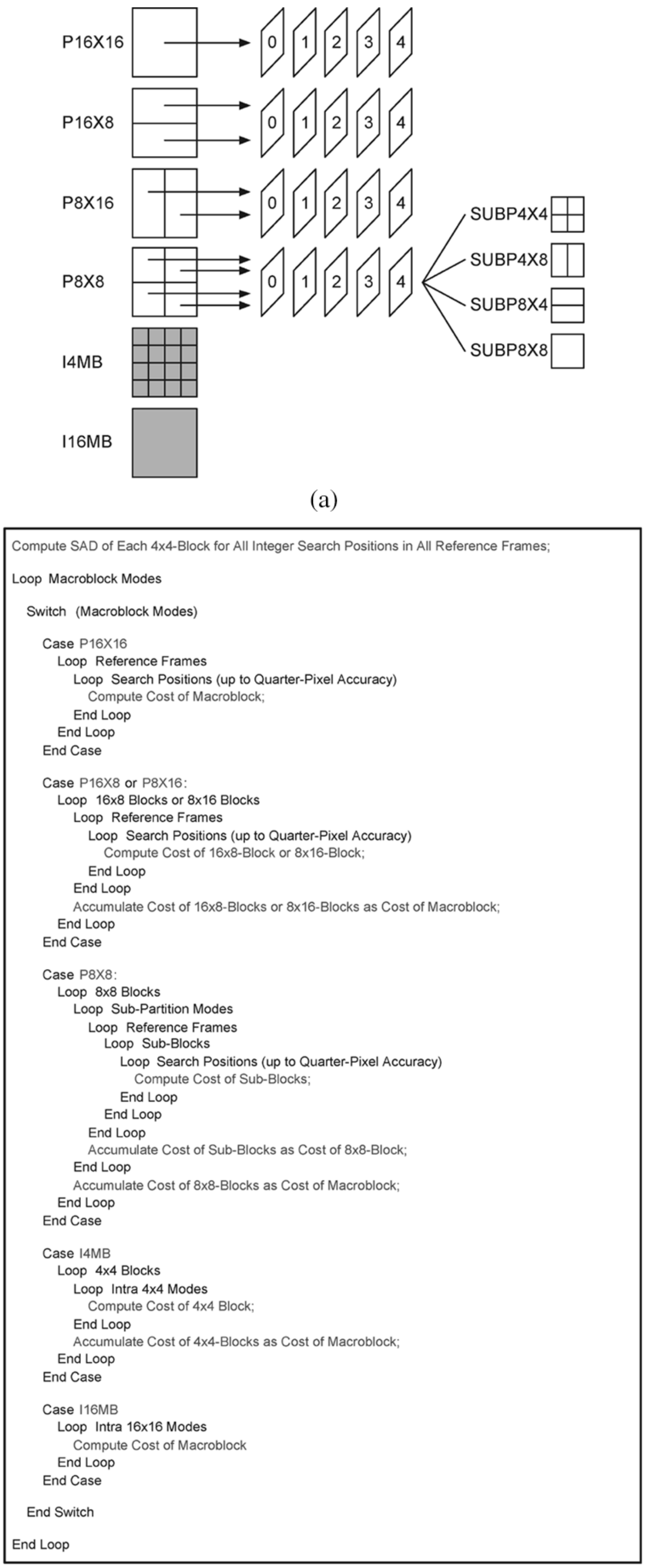
II. Fundamentals

Fig. 3(a) illustrates the prediction part in the reference software baseline encoder. The prediction of a MB is performed mode by mode with full search scheme. There are four inter-MB modes (P8X8, P8X16, P16X8, and P16X16) and two intra-MB modes (I4MB and I16MB). Each 8 × 8 block of the P8X8 mode can be further subpartitioned into smaller blocks (SUBP4 × 4, SUBP4 × 8, SUBP8 × 4, and SUBP8X8), and the subpartitioned blocks in an 8 × 8 block must be predicted from the same reference frame. The best MB mode is chosen by considering a Lagrangian cost function, which includes both distortion and rate (number of bits required to code the side information). Let the QP and the Lagrange parameter λ_{MODE} (a QP-dependent variable) be given. The Lagrangian mode decision for a MB MB_k proceeds by minimizing

$$J_{\mathrm{MODE}}(\mathrm{MB}_k, I_k|\mathrm{QP}, \lambda_{\mathrm{MODE}}) = \mathrm{Distortion}(\mathrm{MB}_k, I_k|\mathrm{QP}) + \lambda_{\mathrm{MODE}} \cdot \mathrm{Rate}(\mathrm{MB}_k, I_k|\mathrm{QP}) \quad (1)$$

where the MB mode I_k is varied over all possible coding modes. Experimental selection of Lagrangian multipliers is discussed in [7] and [8]. Fig. 3(b) shows the pseudocodes of MB mode decision in the reference software baseline encoder. It is clear that the outer loop is for MB modes, and the loops for reference frames are inner. At the beginning of mode decision, SADs of the 16 4 × 4 blocks of a MB are calculated at all integer search positions in all reference frames. These SAD values can be reused for other blocks of larger sizes because their search range centers are exactly the same. When computing the Lagrangian cost of integer search positions, the reference software uses the SAD value as the distortion part. After the best integer search position is found, the Lagrangian costs of the surrounding eight half-search positions are computed, and then again the Lagrangian costs of eight quarter-search positions surrounding the best half-search position are computed. For fractional pixel ME, the reference software allows users to select SAD, sum of absolute transform difference (SATD), or sum of square difference (SSD) as the distortion criterion.

In order to support fast algorithms for multiple reference frames ME, we modified the reference software as follows. Fig. 4(a) and (b) are the illustration and pseudocodes of MB mode decision in our baseline encoder, respectively. Note that now the outer loop is changed to reference frames, and the



(a)

```
Compute SAD of Each 4x4-Block for All Integer Search Positions in All Reference Frames;
Loop  Macroblock Modes
  Switch  (Macroblock Modes)
    Case P16X16
      Loop  Reference Frames
        Loop  Search Positions (up to Quarter-Pixel Accuracy)
          Compute Cost of Macroblock;
        End Loop
      End Loop
    End Case
    Case P16X8 or P8X16:
      Loop  16x8 Blocks or 8x16 Blocks
        Loop  Reference Frames
          Loop  Search Positions (up to Quarter-Pixel Accuracy)
            Compute Cost of 16x8-Block or 8x16-Block;
          End Loop
        End Loop
        Accumulate Cost of 16x8-Blocks or 8x16-Blocks as Cost of Macroblock;
      End Loop
    End Case
    Case P8X8:
      Loop  8x8 Blocks
        Loop  Sub-Partition Modes
          Loop  Reference Frames
            Loop  Sub-Blocks
              Loop  Search Positions (up to Quarter-Pixel Accuracy)
                Compute Cost of Sub-Blocks;
              End Loop
            End Loop
          End Loop
          Accumulate Cost of Sub-Blocks as Cost of 8x8-Block;
        End Loop
        Accumulate Cost of 8x8-Blocks as Cost of Macroblock;
      End Loop
    End Case
    Case I4MB
      Loop  4x4 Blocks
        Loop  Intra 4x4 Modes
          Compute Cost of 4x4 Block;
        End Loop
        Accumulate Cost of 4x4-Blocks as Cost of Macroblock;
      End Loop
    End Case
    Case I16MB
      Loop  Intra 16x16 Modes
        Compute Cost of Macroblock
      End Loop
    End Case
  End Switch
End Loop
```

(b)

Fig. 3. MB mode decision in the reference software baseline encoder. (a) Illustration. (b) Pseudocodes.

loop for MB modes becomes inner. At the beginning of each reference frame loop, we compute SADs of each 4 × 4 block at all search positions only in the current reference frame. At the end of each reference frame loop, we add a checking procedure to see whether it is all right to skip the remaining reference frames and break the loop. In this way, the computational complexity of the prediction part in our baseline encoder is exactly proportional to the number of reference frames searched.

III. Statistical Analysis

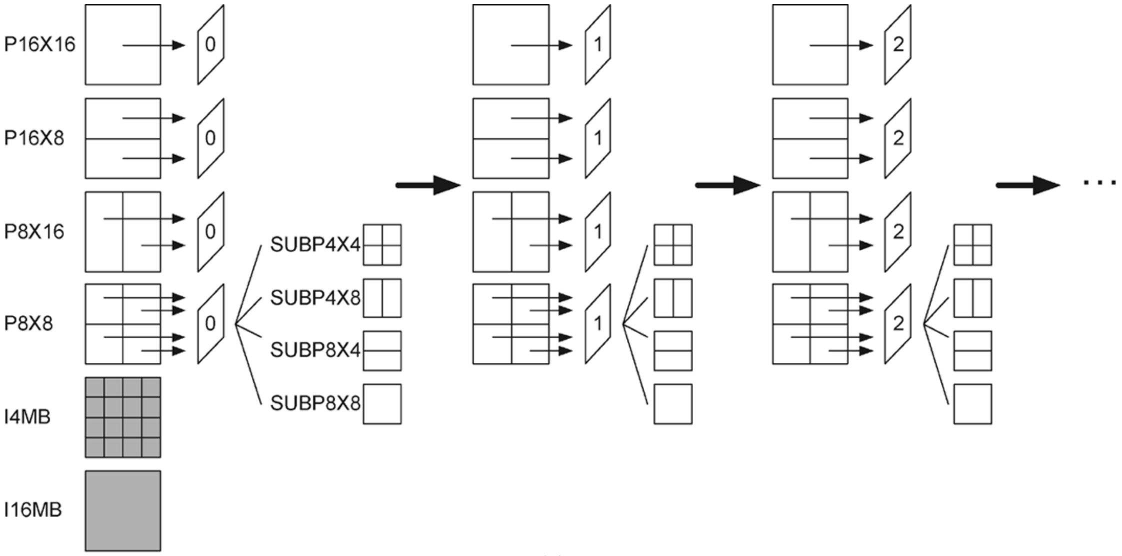
In this section, various statistical analyzes of ten standard sequences under three different QP values ($\mathrm{QP}=20, 30$, and 40), which represent high bit rate, medium bit rate, and low bit rate situations, respectively, will be shown.

A. Percentages of Selected Reference Frames

We first analyze the percentages of inter-MBs predicted from different reference frames and intra-MBs. Ref 0 means the MB predicted from the previous frame, and Ref 1 stands for the previous frame of previous frame, and so on. Note that one MB may be predicted from different reference frames, and we take the farthest reference frame into the statistics. As seen in Table I, 67.58%, 81.35%, and 91.97% of the optimal MVs determined by the full search scheme belong to the nearest reference frame for $\mathrm{QP}=20, 30$, and 40, respectively. The possibilities that the optimal MVs fall in other reference frames are much smaller. In other words, in many cases, a lot of computation is wasted on useless reference frames. The result is quite reasonable. Intuitively, the closer the reference frame is, the higher the correlation should be, except for occluded and uncovered objects. Therefore, we should proceed the block matching process from the nearest reference frame to the farthest reference frame. Another interesting point is that low bit rate cases are more likely to have best reference frames close to the current frame than higher bit rate cases are. This is because the second term (Rate) of the Lagrangian cost function related to reference frames increases with QP, which results in a larger positive bias on the Lagrangian cost for a larger QP. That is, at lower bit rates, the effect of Rate becomes much more obvious.

B. Percentages of Selected MB Modes

The result of MB mode decision after intra-prediction and ME from the previous frame is also a very important cue. In Table II, $A|B$ is defined as follows. A is the percentage of a selected MB mode after intra-prediction and ME from the previous frame. B is the percentage of A that the optimal reference frames remain unchanged after five reference frames are searched. We can see that for $\mathrm{QP} = 20$, there are 59.84%, 05.00%, 04.88%, 28.11%, and 02.17% of the MBs selected as P16X16, P16X8, P8X16, P8X8, and intra, respectively, when only one previous frame is searched. For $\mathrm{QP} = 30$, there are 75.97%, 05.36%, 05.45%, 11.04%, and 02.18% of the MBs selected as P16X16, P16X8, P8X16, P8X8, and intra, respectively. For $\mathrm{QP} = 40$, the corresponding percentages are 89.34%, 03.21%, 03.07%, 01.69%, 02.69%. After the remaining four reference frames are searched, 78.95%, 89.33%, and 96.53% of the 16 × 16-MBs still remain the same selection for $\mathrm{QP} = 20, 30$, and 40, respectively. When the block size becomes smaller, such as P16X8, P8X16, and P8X8-MBs, the percentages that the best reference frames do not change tend



(a)

```
Loop  Reference Frames
  Compute SAD of Each 4x4-Block for All Integer Search Positions in This Reference Frame;
  Loop  Macroblock Modes
    Switch  (Macroblock Modes)
      Case  P16X16
        Loop  Search Positions (up to Quarter-Pixel Accuracy)
          Compute Cost of Macroblock;
        End Loop
      End Case
      Case  P16X8 or  P8X16:
        Loop  8x16 Blocks or 16x8 Blocks
          Loop  Search Positions (up to Quarter-Pixel Accuracy)
            Compute Cost of 16x8-Block or 8x16-Block;
          End Loop
          Accumulate Cost of 16x8-Blocks or 8x16-Blocks as Cost of Macroblock;
        End Loop
      End Case
      Case  P8X8:
        Loop  8x8 Blocks
          Loop  Sub-Partition Modes
            Loop  Sub-Blocks
              Loop  Search Positions (up to Quarter-Pixel Accuracy)
                Compute Cost of Sub-Blocks;
              End Loop
            End Loop
            Accumulate Cost of Sub-Blocks as Cost of 8x8-Block;
          End Loop
          Accumulate Cost of 8x8-Blocks as Cost of Macroblock;
        End Loop
      End Case
      Case  I4MB (Only Required for the First Reference Frame)
        Loop  4x4 Blocks
          Loop  Intra 4x4 Modes
            Compute Cost of 4x4 Block;
          End Loop
          Accumulate Cost of 4x4-Blocks as Cost of Macroblock;
        End Loop
      End Case
      Case  I16MB (Only Required for the First Reference Frame)
        Loop  Intra 16x16 Modes
          Compute Cost of Macroblock
        End Loop
      End Case
    End Switch
  End Loop
  Determine to Keep on Searching the Next Reference Frame or to Skip the Rest;
End Loop
```

(b)

Fig. 4. MB mode decision in our baseline encoder. (a) Illustration. (b) Pseudocodes.

TABLE I
STATISTICS OF SELECTED REFERENCE FRAMES IN PERCENTAGES

Sequences	Ref 0	Ref 1	Ref 2	Ref 3	Ref 4	Intra
Coastguard	70.98	09.07	06.79	06.65	06.31	00.19
Container	87.58	03.68	02.59	02.57	02.82	00.76
Foreman	52.28	14.15	11.20	09.31	09.30	03.76
Hall Monitor	43.22	07.03	10.85	14.92	21.57	02.41
Mobile Calendar	29.59	15.03	17.90	16.90	20.37	00.22
Mother and Daughter	84.06	05.54	03.69	02.79	02.87	01.05
Silent	86.31	04.12	02.72	02.42	02.87	01.56
Stefan	51.24	14.40	10.99	08.10	09.32	05.94
Table Tennis	81.29	06.52	04.52	03.15	03.40	01.12
Weather	89.21	03.98	02.59	01.99	02.14	00.08
Average	67.58	08.35	07.38	06.88	08.10	01.71

CIF size, search range [-16.75, +16.75], *QP*=20.

Sequences	Ref 0	Ref 1	Ref 2	Ref 3	Ref 4	Intra
Coastguard	81.84	05.77	03.91	03.55	03.54	01.41
Container	91.11	02.28	01.33	01.71	01.77	01.81
Foreman	75.41	07.69	05.17	03.94	03.80	03.99
Hall Monitor	92.87	01.76	01.48	01.27	01.51	01.11
Mobile Calendar	38.89	14.33	16.30	14.39	15.85	00.25
Mother and Daughter	93.89	02.05	01.32	01.04	01.15	00.55
Silent	91.62	02.09	01.38	01.27	01.63	02.02
Stefan	67.06	10.54	07.36	05.03	05.48	04.52
Table Tennis	87.35	04.02	02.37	01.58	01.60	03.07
Weather	93.51	02.41	01.50	01.20	01.22	00.15
Average	81.35	05.29	04.21	03.50	03.75	01.89

CIF size, search range [-16.75, +16.75], *QP*=30.

Sequences	Ref 0	Ref 1	Ref 2	Ref 3	Ref 4	Intra
Coastguard	89.07	02.46	01.51	01.27	01.33	04.37
Container	97.32	00.55	00.43	00.42	00.28	01.01
Foreman	88.54	02.62	01.47	01.16	01.29	04.93
Hall Monitor	98.59	00.33	00.20	00.16	00.20	00.53
Mobile Calendar	78.95	07.52	04.98	03.93	04.00	00.62
Mother and Daughter	97.61	00.35	00.23	00.21	00.27	01.33
Silent	94.87	00.70	00.50	00.46	00.58	02.89
Stefan	84.83	03.50	02.53	02.21	02.54	04.39
Table Tennis	92.56	01.56	00.86	00.58	00.57	03.87
Weather	97.34	00.93	00.53	00.43	00.40	00.37
Average	91.97	02.05	01.32	01.08	01.14	02.43

CIF size, search range [-16.75, +16.75], *QP*=40.

to be lower. For $\mathrm{QP} = 20$, it can be seen that $59.84\% \times 78.95\% + 05.00\% \times 53.95\% + 04.88\% \times 53.44\% + 28.11\% \times 47.81\% + 02.17\% \times 81.63\% = 67.76\%$ of MBs need only one reference frame, which is quite consistent with the results in Table I. For $\mathrm{QP} = 30$ and 40, the derived percentages are 82.52% and 94.45%, respectively, which are also very consistent with Table I. Additionally, if a MB is split into smaller blocks when only the previous frame is used for ME, it implies that the motion field is discontinuous. In these cases, the MB may cross the object boundaries, where occlusion and uncovering often occur. Thus, there is a greater possibility that the best matched candidates belong to the other four reference frames. Last but not least, it is easier for MBs to split into smaller partitions at higher bit rates. The reason is still related to the Lagrangian cost function. The second term (Rate) of the Lagrangian cost function related to MB modes increases with QP, which makes it much more difficult for low bit rate situations to select the partition modes with more side information.

C. Detection of All-Zero Transform Quantized Residues

Now we treat the saving of computation for multiple reference frames in terms of compression. After the prediction procedure, residues are transformed, quantized, and then entropy coded. If we can detect the situation that the transformed and quantized coefficients are very close to zero in the first reference frame, we can turn off the matching process for the remaining frames since more computation will not further reduce prediction errors. This concept is very simple and effective. Moreover, discrete cosine transform (DCT), quantization (Q), inverse quantization (IQ), and inverse discrete cosine transform (IDCT) can be also saved by this early detection of all-zero transform quantized coefficients. The quantization steps of 4 × 4 residues in H.264/AVC are described in the following equations:

$$qp\_per = \frac{\mathrm{QP}}{6} \tag{2}$$

$$qp\_rem = \mathrm{QP}\%6 \tag{3}$$

$$qp\_bits = qp\_per + 15 \tag{4}$$

$$qp\_const = \frac{(1 \ll q\_bits)}{6} \tag{5}$$

$$\begin{aligned}\mathrm{QM}[i][j] = (|\mathrm{TR}[i][j]| \cdot quant\_coef[qp\_rem][i][j] \\ +qp\_const) \gg q\_bits\end{aligned} \tag{6}$$

where QP is the (0–51), $\mathrm{QM}[i][j]$ is the 4 × 4 quantized magnitude (QM), $\mathrm{TR}[i][j]$ is the 4 × 4 transformed residues (TRs), and $quant\_coef[qp\_rem][i][j]$ is a three-dimensional (3-D) 6 × 4 × 4 matrix. If the following inequality (derived from $20, 30$, and 40 (6) by setting $\mathrm{QM} = 0$) holds:

$$|\mathrm{TR}[i][j]| < \frac{(2^{q\text{\_bits}} - qp\_const)}{quant\_coef[qp\_rem][i][j]} \equiv \mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j] \tag{7}$$

the QM will be zero, which means each of the 4 × 4 threshold values becomes a function of QP and can be implemented as a look-up table. However, TR is not available before transformation. The only information we have after prediction is SAD (SATD or SSD). Thus, we should find the relation between SAD (SATD or SSD) and TR, and then apply the thresholds directly on SAD (SATD or SSD).

The derivation of threshold values is similar to [9]. The distribution of the pixel values after linear prediction in image can be modeled by a Laplacian distribution, which has a significant peak at zero with exponentially decayed probability at both sides [10]. In addition, the correlation for the pixel values after linear prediction is separable in both horizontal and vertical directions. Therefore, we can approximate the input values at the input of transform by a Laplacian distribution with zero mean and a separable covariance $r(m,n) = \sigma_f^2 \rho^{|m|} \rho^{|n|}$, where m and n are the horizontal and vertical distances, respectively, between two pixels, and $|\rho| < 1$ is the correlation coefficient. Typically, ρ ranges from 0.4 to 0.75, and we use $\rho = 0.6$ in all our simulations. Next, let $f(m,n) = f_{nm}$ and $F(u,v) = F_{vu}$ denote the4 × 4 residues at the DCT input and output, respectively. The DCT can be represented in matrix form as follows:

$$\begin{aligned} F &= AfA^T \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \end{aligned} \tag{8}$$

TABLE II
STATISTICS OF SELECTED MB MODES IN PERCENTAGES

Sequences	P16X16		P16X8		P8X16		P8X8		Intra	
Coastguard	34.30	75.15	06.38	68.64	05.89	62.43	53.19	69.83	00.23	82.12
Container	83.06	91.82	04.65	79.79	04.68	75.32	06.82	59.87	00.79	95.85
Foreman	44.15	65.76	09.82	46.18	10.33	47.74	31.60	43.63	04.11	91.65
Hall Monitor	68.57	56.84	05.08	14.10	03.27	15.71	18.52	16.29	04.56	52.82
Mobile Calendar	35.10	42.22	04.15	25.00	04.38	22.64	56.08	22.72	00.30	74.02
Mother and Daughter	76.41	92.19	05.52	61.88	06.05	60.40	10.90	60.10	01.12	93.90
Silent	74.51	96.22	02.89	66.68	03.41	68.33	17.17	60.34	02.02	77.27
Stefan	32.32	78.85	05.61	54.03	04.41	54.95	50.54	40.18	07.12	83.39
Table Tennis	66.17	93.75	04.10	69.58	03.99	73.26	24.36	55.35	01.38	80.99
Weather	83.76	96.72	01.79	53.67	02.39	53.57	11.96	49.82	00.09	84.26
Average	59.84	78.95	05.00	53.95	04.88	53.44	28.11	47.81	02.17	81.63

CIF size, search range [-16.75, +16.75], *QP*=20.

Sequences	P16X16		P16X8		P8X16		P8X8		Intra	
Coastguard	63.15	90.15	09.18	78.02	09.22	74.53	16.88	64.41	01.57	89.32
Container	93.17	94.76	01.56	63.75	01.78	58.78	01.63	47.44	01.85	97.81
Foreman	69.25	85.63	09.05	63.26	09.35	63.11	08.01	56.01	04.34	91.94
Hall Monitor	91.08	97.27	01.70	51.04	01.17	60.34	04.71	57.19	01.32	83.49
Mobile Calendar	50.79	51.71	09.40	27.04	09.80	27.98	29.71	24.70	00.30	82.25
Mother and Daughter	90.01	96.87	03.55	74.96	04.05	73.32	01.78	59.67	00.61	90.04
Silent	84.82	97.83	03.29	74.09	04.11	75.45	05.28	58.74	02.51	80.52
Stefan	50.94	84.78	08.41	63.74	06.89	63.10	28.14	50.35	05.62	80.41
Table Tennis	77.51	96.60	05.15	69.78	05.23	72.50	08.65	58.91	03.46	88.71
Weather	88.96	97.65	02.34	65.11	02.87	64.27	05.66	57.82	00.17	91.96
Average	75.97	89.33	05.36	63.08	05.45	63.34	11.04	53.52	02.18	87.64

CIF size, search range [-16.75, +16.75], *QP*=30.

Sequences	P16X16		P16X8		P8X16		P8X8		Intra	
Coastguard	85.99	95.60	04.86	75.90	03.36	74.55	01.21	54.77	04.57	95.53
Container	97.30	98.73	00.56	76.90	00.88	76.72	00.23	66.67	01.03	97.87
Foreman	85.64	95.45	04.50	77.27	03.89	74.59	00.70	61.71	05.28	93.31
Hall Monitor	96.72	99.76	00.70	81.30	01.04	82.59	00.98	68.52	00.56	94.21
Mobile Calendar	80.34	85.04	06.69	58.39	07.54	58.86	04.72	48.34	00.70	88.13
Mother and Daughter	97.09	99.26	00.66	80.79	00.83	82.37	00.02	48.00	01.39	95.64
Silent	93.35	98.94	01.27	76.44	01.85	74.73	00.27	53.58	03.25	88.92
Stefan	73.20	95.30	08.35	80.01	06.36	77.83	06.44	53.38	05.65	77.73
Table Tennis	88.91	98.27	02.92	77.08	02.85	77.21	01.21	60.54	04.10	94.42
Weather	94.80	98.99	01.59	76.29	02.08	75.94	01.14	61.18	00.39	95.40
Average	89.34	96.53	03.21	76.04	03.07	75.54	01.69	57.67	02.69	92.12

CIF size, search range [-16.75, +16.75], *QP*=40.

A|B is defined as follows.
A: % of macroblocks when only 1 ref. frame is allowed for motion estimation.
B: % of A that does not change the best ref. frame after 5 ref. frames are all searched.

where A is the matrix of transform coefficients. The variance of the (u,v)th DCT coefficients $\sigma_F^2(u,v)$ can be written as [11]

$$\sigma_F^2(u,v) = \sigma_f^2[ARA^T]_{u,u}[ARA^T]_{v,v} \tag{9}$$

where

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix} \tag{10}$$

and $[\cdot]_{u,u}$ is the (u,u)th component of the matrix. Since the distribution of $f(m,n)$ has a zero mean, the mean of each $F(u,v)$ will also be zero. The variance of the DCT coefficients will be

$$\left[\sigma_F^2(u,v)\right] = \sigma_f^2 \begin{bmatrix} 89.72 & 85.17 & 16.97 & 26.98 \\ 85.17 & 80.86 & 16.11 & 25.61 \\ 16.97 & 16.11 & 3.21 & 5.10 \\ 26.98 & 25.61 & 5.10 & 8.11 \end{bmatrix} \tag{11}$$

for $\rho = 0.6$. In this way, the standard deviation of the DCT coefficients can be estimated by the standard deviation of the pixel values at the DCT input as

$$\begin{bmatrix} \sigma_{F00} & \sigma_{F01} & \sigma_{F02} & \sigma_{F03} \\ \sigma_{F10} & \sigma_{F11} & \sigma_{F12} & \sigma_{F13} \\ \sigma_{F20} & \sigma_{F21} & \sigma_{F22} & \sigma_{F23} \\ \sigma_{F30} & \sigma_{F31} & \sigma_{F32} & \sigma_{F33} \end{bmatrix} = \sigma_f \begin{bmatrix} 9.47 & 9.23 & 4.12 & 5.19 \\ 9.23 & 8.99 & 4.01 & 5.06 \\ 4.12 & 4.01 & 1.79 & 2.26 \\ 5.19 & 5.06 & 2.26 & 2.85 \end{bmatrix} \tag{12}$$

Although the generalized Gaussian distribution is the most accurate representation of transformed coefficients, for the sake of simplicity, we choose the more commonly used Laplacian distribution to model them [12], [13]. For a zero-mean Laplacian distribution, the probability that a value will fall with $(-3\sigma, 3\sigma)$ is about 99%. According to the all-zero situation shown in (7), if

$$\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j] > 3\sigma_F(i,j) \tag{13}$$

TABLE III
VALUES OF $\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j]$

Quantization Parameter	(*i*, *j*)= (0, 0), (2, 0), (0, 2), (2, 2).	(*i*, *j*)= (1, 0), (3, 0), (0, 1), (2, 1), (1, 2), (3, 2), (0, 3), (2, 3).	(*i*, *j*)= (1, 1), (3, 1), (1, 3), (3, 3).
QP=00	2.08	3.39	5.21
QP=05	3.75	5.99	9.44
QP=10	6.67	10.42	16.28
QP=15	11.67	18.75	29.95
QP=20	21.67	33.33	52.09
QP=25	36.67	58.33	93.76
QP=30	66.67	108.33	166.66
QP=35	120.00	191.67	302.04
QP=40	213.33	333.33	520.90
QP=45	373.34	600.04	958.39
QP=50	693.37	1066.60	1666.79

(i, j): position index of a 4x4-block.

then the DCT coefficients will be zero after quantization almost all the time. More generally, we can use

$$\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j] > n \cdot \sigma_F(i,j) \tag{14}$$

where n controls the probability of all-zero transform quantized coefficients. For $n = 2$, the probability reduces to 94%. As stated before, $\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j]$ is a function of QP and (i,j). We list some corresponding values of $\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j]$ in Table III. Originally, given a σ_f, we can get the 16 values of σ_F. In order to detect the all-zero case, we have to compare the 16 values of σ_F with different thresholds. However, as shown in (12), $\sigma_F(0,0)$ is the largest among $\sigma_F(u,v)$, which means the probability of a zero dc term after DCT is smaller than that of zero ac terms. Meanwhile, as shown in Table III, $\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][0][0]$ is the smallest among $\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][i][j]$, which means the quantization step size of dc term is the smallest. Therefore, we only have to compare the dc term in (7) for all-zero detection.

In order to save the extra computation for the standard deviation of pixel values at DCT input, we can estimate it from SAD as follows. The expected mean absolute value of a zero-mean Laplacian distributed random variable x is

$$\begin{aligned} E(|x|) &= \int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{2}\sigma} e^{\frac{-\sqrt{2}|x|}{\sigma}} \cdot dx \\ &= 2\int_{0}^{\infty} |x| \frac{1}{\sqrt{2}\sigma} e^{\frac{-\sqrt{2}|x|}{\sigma}} \cdot dx = \frac{\sigma}{\sqrt{2}}. \end{aligned} \tag{15}$$

We can approximate the $E(|x|)$ by the SAD of current MB, so we have

$$\sigma_f = \sqrt{2} \cdot \frac{\mathrm{SAD}}{256} = \sqrt{2} \cdot (\mathrm{SAD} \gg 8) \tag{16}$$

If the MB mode is not P16X16, we can still accumulate the SADs of all partitions as SAD of the MB. For example, if the best block mode is P16X8, we can add the SADs of the two 16 × 8 blocks as the SAD value used in (16). According to (12), (14), and (16), now the all-zero situation holds if SAD is smaller than a predefined threshold, as follows:

$$\begin{aligned} \mathrm{SAD} &= \frac{256\sigma_f}{\sqrt{2}} = \frac{256\sigma_f}{\sqrt{2}} \cdot \frac{\sigma_{F00}}{\sigma_{F00}} = \frac{256\sigma_f}{\sqrt{2}} \cdot \frac{\sigma_{F00}}{9.47\sigma_f} \\ &< \frac{256\mathrm{TH}_{\mathrm{TR}}[\mathrm{QP}][0][0]}{9.47\sqrt{2} \cdot n} \equiv \mathrm{TH}_{\mathrm{SAD}}. \end{aligned} \tag{17}$$

Sometimes the user may choose SATD or SSD, instead of SAD, as the criterion of block matching. Now we first derive the relation between SATD and SAD. Once the correlation is found, (17) can be still used for all-zero detection. In the reference software, the SATD of a 4 × 4 block is defined as half of the sum of absolute values of a two-dimensional (2-D) 4 × 4 Hadamard-transformed residues.

$$\begin{aligned} R &= HfH^T \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} f_{00} & f_{01} & f_{02} & f_{03} \\ f_{10} & f_{11} & f_{12} & f_{13} \\ f_{20} & f_{21} & f_{22} & f_{23} \\ f_{30} & f_{31} & f_{32} & f_{33} \end{bmatrix} \\ &\quad \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \end{aligned} \tag{18}$$

$$\mathrm{SATD} = \frac{1}{2} \cdot \sum_{v=0}^{3}\sum_{u=0}^{3} |R(u,v)|. \tag{19}$$

For simplicity, we again assume that the residues after Hadamard transform is still Laplacian distributed, so the SATD of a 4 × 4 block can be approximated by replacing A in (9) with H

$$\begin{aligned} \mathrm{SATD} &= \frac{1}{2} \cdot \sum_{v=0}^{3}\sum_{u=0}^{3} |R(u,v)| = \frac{1}{2} \cdot \sum_{v=0}^{3}\sum_{u=0}^{3} \frac{\sigma_R(u,v)}{\sqrt{2}} \\ &= \frac{\sigma_f}{2\sqrt{2}} \cdot \sum_{v=0}^{3}\sum_{u=0}^{3} r(u,v) \cong \frac{26\sigma_f}{\sqrt{2}} \end{aligned} \tag{20}$$

where

$$\begin{aligned} \sigma_R^2(u,v) &= r(u,v) \cdot r(u,v) \cdot \sigma_f^2 \\ &= \sigma_f^2 \begin{bmatrix} 89.72 & 31.52 & 16.97 & 13.34 \\ 31.52 & 11.08 & 5.96 & 4.69 \\ 16.97 & 5.96 & 3.21 & 2.52 \\ 13.34 & 4.69 & 2.52 & 1.98 \end{bmatrix}. \end{aligned} \tag{21}$$

From (16), the SAD of a 4 × 4 block can be approximated as

$$\mathrm{SAD} \cong \frac{16\sigma_f}{\sqrt{2}}. \tag{22}$$

The relation between SATD and SAD can be seen from (20) and (22). As for the relation between SSD and SAD, it can be deduced by the same probability theory.

D. SKIP Mode

In H.264/AVC, SKIP mode is a special case of P16X16. Nothing but run of SKIP MBs is transmitted to the decoder. At the decoder side, a SKIP MB is directly reconstructed as a 16 ×

TABLE IV
STATISTICS OF SKIP MBS

Sequences	SKIP(%)	Unchanged(%)	Unchanged I4MB Cost	Changed I4MB Cost
Coastguard	20.22	09.24	2405.29	3195.05
Container	66.91	39.69	2745.07	2821.77
Foreman	21.99	34.76	1118.75	2386.08
Hall Monitor	35.31	00.33	0936.21	1489.14
Mobile Calendar	11.80	07.70	1028.58	6482.76
Mother and Daughter	63.15	62.18	0805.38	1455.58
Silent	68.39	78.08	2721.77	2776.93
Stefan	20.41	42.12	0646.90	4337.60
Table Tennis	58.43	68.56	2506.98	4992.85
Weather	79.08	91.46	4457.16	4084.41
Average	44.57	43.41	1937.21	3402.22

CIF size, search range [-16.75, +16.75], *QP*=20.

Sequences	SKIP(%)	Unchanged(%)	Unchanged I4MB Cost	Changed I4MB Cost
Coastguard	44.99	51.89	3174.86	3886.71
Container	84.56	89.45	2801.75	4781.62
Foreman	46.75	80.63	2155.24	3302.36
Hall Monitor	87.29	71.22	2115.50	2818.27
Mobile Calendar	20.00	36.36	4159.93	7163.43
Mother and Daughter	78.25	92.52	1447.04	2438.47
Silent	77.85	91.95	3068.86	3231.85
Stefan	34.93	66.05	1800.74	6868.38
Table Tennis	68.66	90.51	3619.96	4606.13
Weather	83.67	94.64	4804.33	5166.19
Average	62.69	76.52	2914.82	4426.34

CIF size, search range [-16.75, +16.75], *QP*=30.

Sequences	SKIP(%)	Unchanged(%)	Unchanged I4MB Cost	Changed I4MB Cost
Coastguard	69.63	92.01	4578.40	6502.98
Container	92.76	98.42	3989.78	9347.05
Foreman	69.16	94.43	3567.66	5008.36
Hall Monitor	95.47	96.85	3468.95	5520.13
Mobile Calendar	53.73	78.99	7913.19	9219.22
Mother and Daughter	92.76	98.24	2499.82	3883.65
Silent	88.74	97.21	4135.21	4856.89
Stefan	56.79	88.84	5250.04	9090.37
Table Tennis	82.11	96.67	4388.72	5726.88
Weather	90.25	97.22	6100.61	6973.20
Average	79.14	93.89	4589.24	6612.87

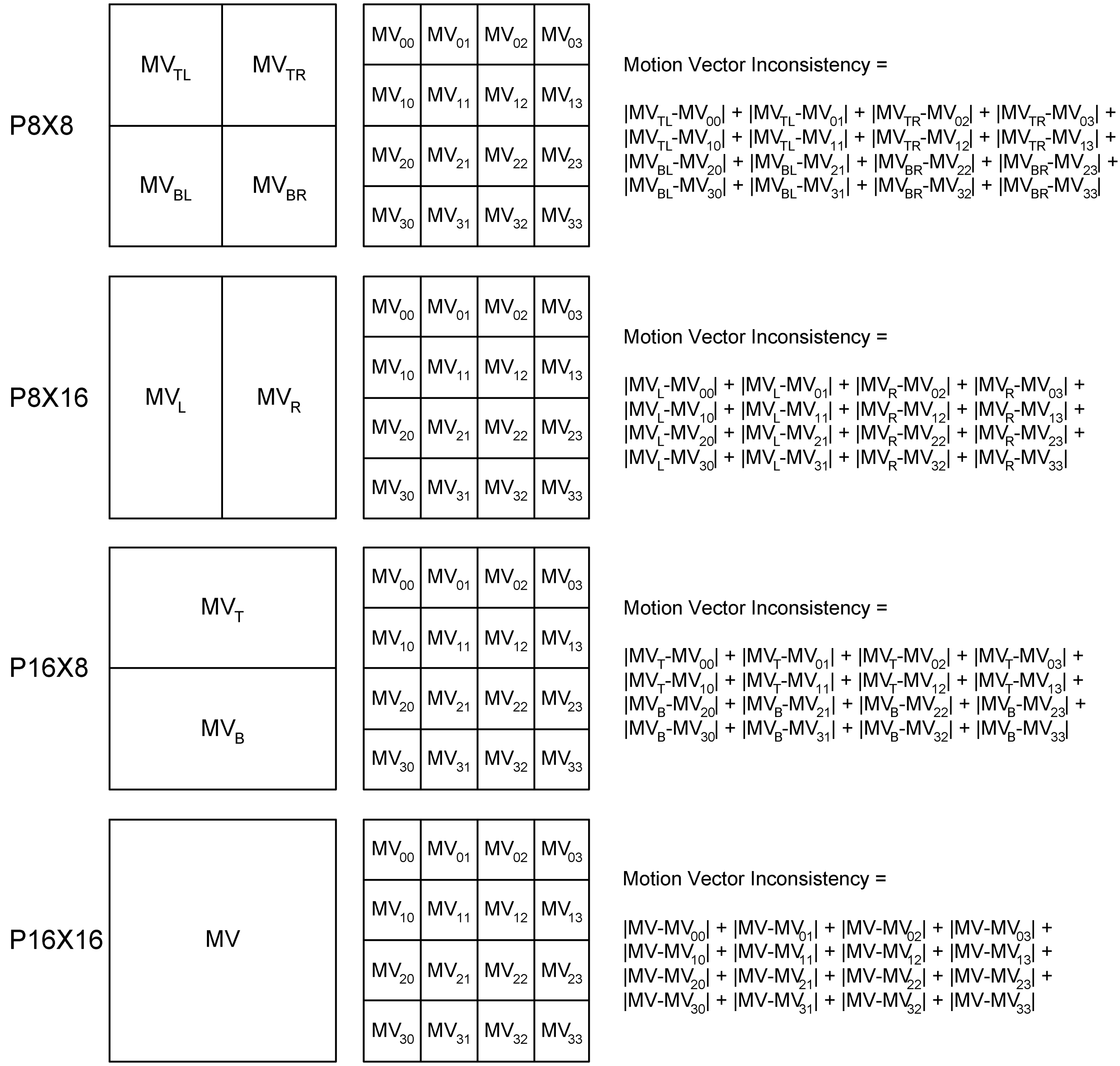
CIF size, search range [-16.75, +16.75], *QP*=40.

SKIP: % of SKIP macroblocks when only 1 ref. frame is allowed for motion estimation.
Unchanged: % of SKIP macroblocks that remain the same after 5 ref. frames are all searched.
Unchanged I4MB Cost: average I4MB cost of the unchanged SKIP macroblocks.
Changed I4MB Cost: average I4MB cost of the changed SKIP macroblocks.

16 block by prediction from previous one frame with predefined MV (MV predictor in general or zero MV at slice boundary) without any residues. Statistics of SKIP modes are shown in Table IV. The first column is sequence name, the second column is the percentage of MBs selected as SKIP after one reference frame is searched, the third column is the percentage of SKIP MBs that still remain SKIP after five reference frames are searched, the fourth column is the average I4MB cost of the unchanged SKIP MBs (remaining SKIP after five reference frames are searched), and the last column is the average I4MB cost of the changed SKIP MBs (failing to remain SKIP after five reference frames are searched). According to the statistics, for low bit rate cases (high QP), SKIP mode is more favored. The percentages of SKIP MBs after one reference frame is searched are 44.57%, 62.69%, and 79.14% for $\mathrm{QP} = 20, 30$, and 40, respectively. Out of these SKIP MBs, 43.41%, 76.52%, and 93.89% still remain SKIP after five reference frames are searched for $\mathrm{QP} = 20, 30$, and 40, respectively. This is quite reasonable because at low bit rate situations, side information becomes very "expensive." In addition to QP values, we found that texture is also a useful information. We use I4MB cost as a rough estimation of texture. Given a QP value, the higher the I4MB cost is, the more highly textured the MB is. Based on the statistics, SKIP mode is also preferred in flat areas. The average I4MB cost of the SKIP MBs remaining SKIP after full search among five reference frames is much smaller than that of SKIP MBs changing best modes.

E. MV Inconsistency

Now we try to find the correlation between MVs of variable block sizes and optimal reference frames. After ME from the previous frame, there are one MV for a 16 × 16 block, two MVs for 16 × 8 blocks, two MVs for 8 × 16 blocks, four MVs for 8 × 8 blocks, and 16 MVs for 4 × 4 blocks. If the best mode is P16X16, P16X8, P8X16, or P8X8, the definition of MV inconsistency for each of the four MB modes is shown in Fig. 5, respectively. Note that $|MV1 - MV2|$ denotes the sum of the L1 distance of the two components between MV1 and MV2. Intuitively, the best MVs of larger blocks and those of smaller blocks should be very similar for objects with homogeneous motion and sufficient texture. For object boundaries where the motion fields are discontinuous, MVs of different block sizes tend to be different. Next, we keep on searching the other four reference frames. If the optimal reference frame of a MB does


P8X8
MV_{TL}
MV_{TR}
MV_{BL}
MV_{BR}
P8X16
MV_L
MV_R
P16X8
MV_T
MV_B
P16X16
MV
MV_{00} MV_{01} MV_{02} MV_{03}
MV_{10} MV_{11} MV_{12} MV_{13}
MV_{20} MV_{21} MV_{22} MV_{23}
MV_{30} MV_{31} MV_{32} MV_{33}

Motion Vector Inconsistency =

$$|MV_{TL}-MV_{00}| + |MV_{TL}-MV_{01}| + |MV_{TR}-MV_{02}| + |MV_{TR}-MV_{03}| + |MV_{TL}-MV_{10}| + |MV_{TL}-MV_{11}| + |MV_{TR}-MV_{12}| + |MV_{TR}-MV_{13}| + |MV_{BL}-MV_{20}| + |MV_{BL}-MV_{21}| + |MV_{BR}-MV_{22}| + |MV_{BR}-MV_{23}| + |MV_{BL}-MV_{30}| + |MV_{BL}-MV_{31}| + |MV_{BR}-MV_{32}| + |MV_{BR}-MV_{33}|$$

Motion Vector Inconsistency =

$$|MV_L-MV_{00}| + |MV_L-MV_{01}| + |MV_R-MV_{02}| + |MV_R-MV_{03}| + |MV_L-MV_{10}| + |MV_L-MV_{11}| + |MV_R-MV_{12}| + |MV_R-MV_{13}| + |MV_L-MV_{20}| + |MV_L-MV_{21}| + |MV_R-MV_{22}| + |MV_R-MV_{23}| + |MV_L-MV_{30}| + |MV_L-MV_{31}| + |MV_R-MV_{32}| + |MV_R-MV_{33}|$$

Motion Vector Inconsistency =

$$|MV_T-MV_{00}| + |MV_T-MV_{01}| + |MV_T-MV_{02}| + |MV_T-MV_{03}| + |MV_T-MV_{10}| + |MV_T-MV_{11}| + |MV_T-MV_{12}| + |MV_T-MV_{13}| + |MV_B-MV_{20}| + |MV_B-MV_{21}| + |MV_B-MV_{22}| + |MV_B-MV_{23}| + |MV_B-MV_{30}| + |MV_B-MV_{31}| + |MV_B-MV_{32}| + |MV_B-MV_{33}|$$

Motion Vector Inconsistency =

$$|MV-MV_{00}| + |MV-MV_{01}| + |MV-MV_{02}| + |MV-MV_{03}| + |MV-MV_{10}| + |MV-MV_{11}| + |MV-MV_{12}| + |MV-MV_{13}| + |MV-MV_{20}| + |MV-MV_{21}| + |MV-MV_{22}| + |MV-MV_{23}| + |MV-MV_{30}| + |MV-MV_{31}| + |MV-MV_{32}| + |MV-MV_{33}|$$

Fig. 5. Definition of MV inconsistency of an MB.

not change after five reference frames are searched, we classify these MBs as type I. Otherwise, we classify them as type II. The average MV inconsistency of type I and that of type II for each sequence is shown in Table V. As can be seen, for P16X16, the MV inconsistency of MBs with optimal reference frame belonging to the farther four frames are much larger than that of MBs predicted by the previous frame. It is easy to apply a clear threshold value on the MV inconsistency for P16X16-MBs after ME from the previous frame is done. If the MV inconsistency is very small, it should be fine to skip the remaining reference frames. For P16X8, P8X16, and P8X8, the MV inconsistency is still a good cue to distinguish between the two categories of MBs, but the threshold value is not that clear. In order not to sacrifice video quality, the threshold values for the MV inconsistency should not be located in between the average value of type I MBs and that of type II MBs. The threshold should be biased toward type I to prevent quality loss. Hence, the miss detection rate of the best reference frame for type II MBs can be reduced, but the false alarm rate of the best reference frame for type I MBs will be increased.

F. MV Location

The above analyses provide useful clues to detect uncovered backgrounds, new objects, or periodic motions for multiple reference frames ME to achieve better coding efficiency without wasting useless computation. In this subsection, we focus on the sampling, which is also one of the reasons why multiple reference frame ME can achieve much better prediction. In the real word, the light field is continuous. On the contrary, the sensor array of camera is discrete. Assume a perfect edge is right across a line of sensor grids. When the object undergoes a

TABLE V
STATISTICS OF AVERAGE MV INCONSISTENCY IN UNITS OF QUARTER PIXEL

Sequences	P16X16		P16X8		P8X16		P8X8	
Coastguard	22.52	21.88	37.21	29.98	33.78	31.19	53.60	57.26
Container	11.74	09.41	25.43	17.64	22.37	14.57	19.67	15.91
Foreman	25.78	36.08	45.35	43.80	41.02	41.67	57.49	50.72
Hall Monitor	17.43	26.93	62.34	61.63	58.73	64.52	60.77	60.19
Mobile Calendar	17.56	33.85	30.54	52.26	28.59	51.07	36.89	55.79
Mother and Daughter	08.63	24.29	30.74	30.06	29.58	27.73	41.31	43.80
Silent	06.22	27.07	42.56	40.10	31.83	32.65	49.66	55.67
Stefan	28.88	44.42	69.84	76.04	51.30	58.60	49.98	61.18
Table Tennis	08.74	20.49	42.41	55.01	34.49	40.08	55.38	63.40
Weather	04.07	29.85	34.74	40.81	32.74	37.27	37.27	42.69
Average	15.16	27.43	42.12	44.73	36.44	39.94	46.20	50.66

CIF size, search range [-16.75, +16.75], *QP*=20.

Sequences	P16X16		P16X8		P8X16		P8X8	
Coastguard	26.08	39.47	48.53	52.23	42.15	43.31	53.92	62.54
Container	07.66	12.76	34.10	24.61	25.13	16.12	22.55	18.19
Foreman	24.17	41.36	58.58	52.19	54.62	47.47	66.88	60.27
Hall Monitor	04.37	22.06	41.68	37.31	43.78	53.22	50.86	52.69
Mobile Calendar	17.10	27.26	27.70	37.16	25.33	34.22	38.48	43.24
Mother and Daughter	10.78	42.40	52.03	55.51	47.77	49.54	63.21	72.46
Silent	10.76	49.85	60.57	59.62	51.52	54.53	67.29	80.36
Stefan	21.29	34.13	41.53	40.44	34.31	40.00	45.61	51.74
Table Tennis	10.36	52.13	58.18	68.36	53.56	62.07	63.62	77.05
Weather	06.49	34.85	37.88	42.89	34.35	39.49	42.35	50.02
Average	13.91	35.63	46.08	47.03	41.25	44.00	51.48	56.86

CIF size, search range [-16.75, +16.75], *QP*=30.

Sequences	P16X16		P16X8		P8X16		P8X8	
Coastguard	25.14	61.80	65.99	79.79	60.00	67.94	55.02	67.11
Container	05.46	18.03	30.67	25.33	21.94	20.09	30.56	40.11
Foreman	20.86	63.05	67.42	73.25	70.00	71.69	64.91	72.27
Hall Monitor	03.95	45.05	59.93	64.68	46.46	57.57	57.66	72.01
Mobile Calendar	16.53	28.49	32.55	34.70	28.03	30.57	45.21	63.44
Mother and Daughter	08.21	69.23	78.82	78.18	70.71	78.79	80.70	77.55
Silent	07.98	73.35	83.64	96.98	76.00	82.92	90.64	99.24
Stefan	16.29	46.21	41.21	64.03	40.45	67.77	68.26	98.40
Table Tennis	08.97	73.40	73.39	87.16	68.73	83.26	70.30	89.26
Weather	08.39	47.28	53.27	60.31	43.38	50.19	57.17	74.46
Average	12.18	52.59	58.69	66.44	52.57	61.08	62.05	75.38

CIF size, search range [-16.75, +16.75], *QP*=40.

I|II is defined as follows.
I: maintain the same selected ref. frame and macroblock mode after 5 ref. frames are all searched.
II: change selected ref. frame and macroblock mode after 5 ref. frames are all searched.

TABLE VI
STATISTICS OF MV LOCATION IN PERCENTAGES

Sequences	INT	Ref=0
Coastguard	06.05	75.49
Container	64.02	98.66
Foreman	08.45	82.44
Hall Monitor	63.19	58.92
Mobile Calendar	04.60	77.18
Mother and Daughter	59.32	97.33
Silent	67.75	98.14
Stefan	11.01	90.63
Table Tennis	58.79	95.60
Weather	78.81	98.68
Average	42.20	87.31

CIF size, search range [-16.75, +16.75], *QP*=20.

Sequences	INT	Ref=0
Coastguard	12.49	91.99
Container	83.69	99.12
Foreman	21.85	92.36
Hall Monitor	89.29	97.84
Mobile Calendar	06.91	83.32
Mother and Daughter	72.94	99.07
Silent	75.51	99.21
Stefan	15.91	95.01
Table Tennis	64.11	98.85
Weather	82.32	99.32
Average	52.50	95.61

CIF size, search range [-16.75, +16.75], *QP*=30.

Sequences	INT	Ref=0
Coastguard	47.89	97.96
Container	91.99	99.75
Foreman	45.30	98.02
Hall Monitor	95.40	99.88
Mobile Calendar	15.68	92.83
Mother and Daughter	91.78	99.72
Silent	86.43	99.62
Stefan	23.09	97.48
Table Tennis	76.93	99.64
Weather	88.13	99.73
Average	66.26	98.46

CIF size, search range [-16.75, +16.75], *QP*=40.

INT: % of macroblocks whose MV's are all at integer locations after 1 ref. frame ME.
Ref=0: % of the INT macroblocks that still remain the same after 5 ref. frames ME.

fractional-pixel motion, the perfect edge will be blurred. Fractional-pixel ME can significantly improve the prediction by interpolating the subpixels. However, in some cases, especially for highly textured video sequences, fractional-pixel ME is not enough. Multiple reference frames ME provides a better chance for the object to find a matched candidate without the discrete sampling defects. Therefore, we presume that if the MV location is an integer position instead of a fractional one, multiple reference frames ME may not be very helpful (unless the MB is an uncovered area or the motion field is not homogeneous). Table VI can somewhat support our claim. After one reference frame ME, if the MVs of a MB are all at integer locations, the probabilities that the optimal reference frame is the previous one frame are 87.31%, 95.61%, and 98.46% for $\mathrm{QP} = 20, 30$, and 40, respectively.

IV. Summary and Proposed Algorithm

Before proposing our fast multiple reference frames ME algorithm, let us summarize the statistical analyzes as follows.

- Reference frames with smaller temporal distance are more likely to be optimal, especially for the previous frame.
- The Lagrangian mode decision is more in favor of closer reference frames at low bit rates than at high bit rates.
- The Lagrangian mode decision is more in favor of MB modes with fewer partitions at low bit rates than at high bit rates.
- If P16X16 mode is selected, the optimal reference frame tends to remain the same.
- If inter-modes with smaller blocks (P16X8, P8X16, and P8X8) are selected, searching more reference frames tends to be helpful.
- For all-zero transform quantized residues, more block matching may be a waste of computation.
- SAD, SATD, or SSD can be used to detect the all-zero case.
- If texture is not significant and if the QP value is large, SKIP mode is often selected as the best MB mode.
- I4MB cost value can be used as a criterion to measure the complexity of texture.
- If MVs of larger blocks are very similar to those of smaller blocks, it is likely that no occlusion or uncovering occurs in the MB, so one reference frame may be enough.
- If MVs of larger blocks are very different from those of smaller blocks, the MB may cross object boundaries where

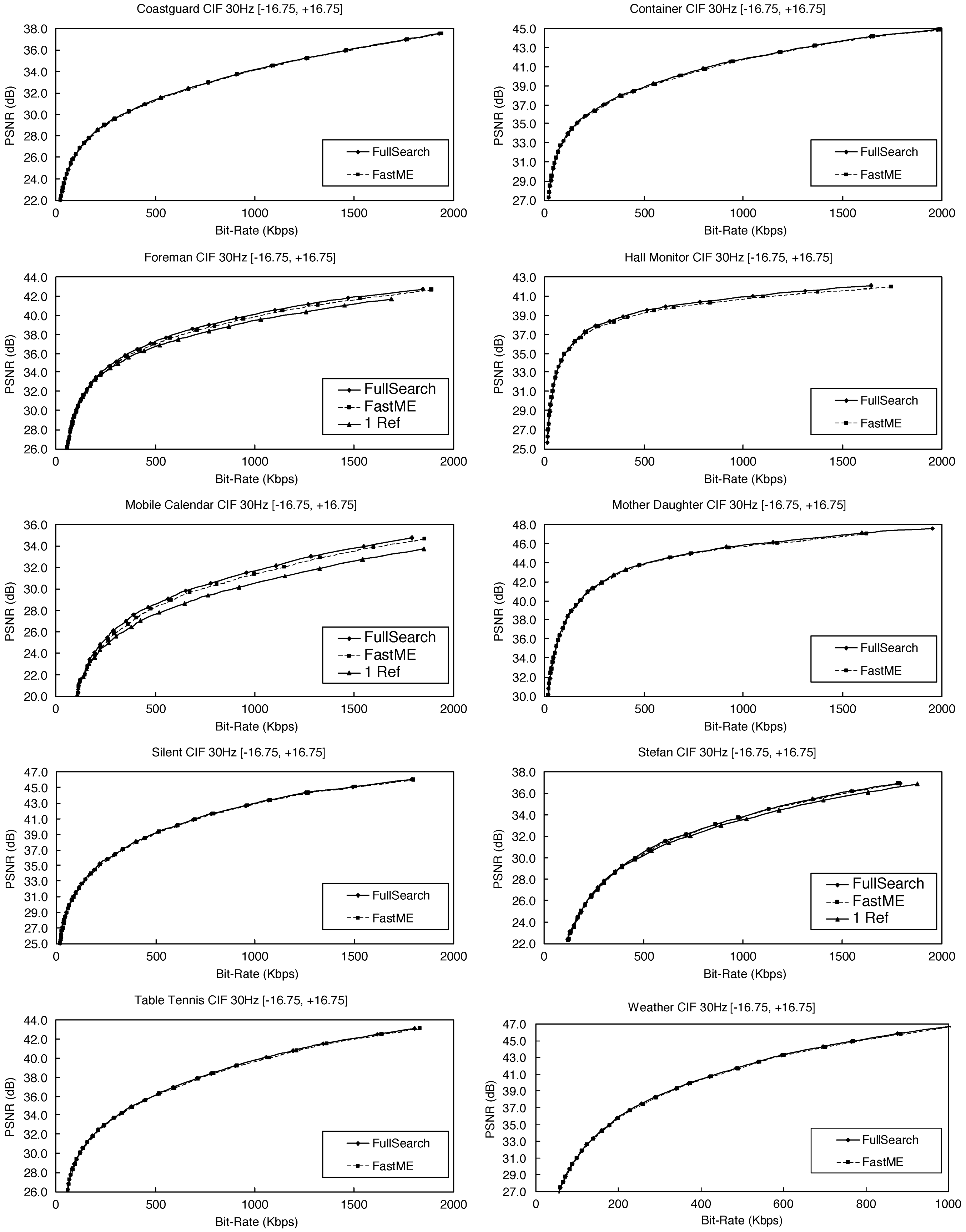


Fig. 6. Rate distortion curves.

the motion field is inhomogeneous, and thus requires more reference frames.

- Sampling the continuous light field by discrete sensor array may blur the sharp edges.

- Fractional-pixel ME can improve the prediction for sampling defects, but for highly textured areas, multiple reference frames are more helpful.
- If the object undergoes an integer motion, the prediction gain of multiple reference frames may not be significant.

According to the above summary, we propose a fast algorithm for multiple reference frame ME to save the computation of full search and to maintain the same video quality. The flowchart and pseudocodes have already been shown in Fig. 4. In the following, we list the steps for each MB to check whether it is necessary to search the next reference frame at the end of each reference frame loop.

- Criterion 1: if $(\mathrm{SATD} \leq \mathrm{TH}_{\mathrm{SATD}})$, early terminate.
- Criterion 2: if $((\mathrm{Ref} == 0)$ && $(\mathrm{best\,MV} == \mathrm{SKIP\,MV})$ && $(\mathrm{QP} > \mathrm{TH}_{\mathrm{QP}}))$, early terminate.
- Criterion 3: if (all MV locations $\in$ integer values), early terminate.
- Criterion 4: if (motion vector inconsistency $\leq \mathrm{TH}_{\mathrm{MVD}}$), early terminate.

Criterion 1 states that when the best SATD does not exceed $\mathrm{TH}_{\mathrm{SATD}}$, we will stop the searching process. The determination of $\mathrm{TH}_{\mathrm{SATD}}$ is described in Section III-C. Note that SATD can be replaced by SAD or SSD.

Criterion 2 is related to SKIP mode. If the best reference frame is previous frame and if the best MV is the same as that of SKIP mode, we call this MB as potential SKIP MB (because before transforming and quantizing the residues, we cannot know for sure if it is really a SKIP MB). For a potential SKIP MB, if its QP is larger than $\mathrm{TH}_{\mathrm{QP}}$, the multiple reference frames loop will be early terminated. The determination of $\mathrm{TH}_{\mathrm{QP}}$ is empirically obtained. Recall that SKIP mode is more favored in flat areas and at low bit rate cases, I4MB cost is used to decide $\mathrm{TH}_{\mathrm{QP}}$ as follows:

$$\begin{aligned} &\mathrm{if}(I4\mathrm{MBCost} < 2000), \quad \mathrm{TH}_{\mathrm{QP}} = 0 \\ &\mathrm{else\ if}(I4\mathrm{MBCost} > 8000), \quad \mathrm{TH}_{\mathrm{QP}} = 35 \\ &\mathrm{else} \quad \mathrm{TH}_{\mathrm{QP}} = \frac{35(I4\mathrm{MBCost} - 2000)}{6000}. \end{aligned} \tag{23}$$

First, if I4MB cost is smaller than 2000, which means the MB is rarely textured, the potential SKIP MB will omit the remaining reference frames. Second, if QP is larger than 35, the potential SKIP MB will also give up searching the remaining reference frames. Third, $\mathrm{TH}_{\mathrm{QP}}$ ranges from 0 to 35. It is linearly dependent of I4MB cost when I4MB cost ranges from 2000 to 8000. The lower the I4MB cost, the smaller the $\mathrm{TH}_{\mathrm{QP}}$, and the more possible the potential SKIP MB can be early terminated.

Criterion 3 deals with the discrete sampling problem, as described in Section III-F. If all the MVs of a MB are at integer locations, the block matching process will be finished.

Criterion 4 detects whether the motion field is homogeneous or not. The definition of MV inconsistency can be found in Section III-E. In our experiments, $\mathrm{TH}_{\mathrm{MVD}}$ is also empirically obtained. For P16X16 MBs, $\mathrm{TH}_{\mathrm{MVD}}$ can be set larger to increase the probability of early termination. For other split MBs (P16X8, P8X16, and P8X8), $\mathrm{TH}_{\mathrm{MVD}}$ can be set smaller to decrease the opportunity of early termination.

TABLE VII
AVERAGE MISS DETECTION RATE AND FALSE ALARM RATE

Sequences	Miss Detection	False Alarm
Coastguard	10.97%	05.94%
Container	04.74%	02.41%
Foreman	15.59%	06.12%
Hall Monitor	15.02%	04.24%
Mobile Calendar	24.07%	11.75%
Mother and Daughter	05.44%	03.27%
Silent	04.72%	02.68%
Stefan	13.58%	08.01%
Table Tennis	06.15%	03.44%
Weather	03.98%	02.53%
Average	10.43%	05.04%

CIF size, search range [-16.75, +16.75].

V. Simulation Results and Discussion

Fig. 6 compares the rate-distortion curves of various standard sequences. For sufficiently textured videos with large and complex motion, the prediction gain of multiple reference frame ME is very significant. Foreman contains an abrupt fast pan of camera, Mobile Calendar has sophisticated texture and zooming camera, and Stefan is a close-up shot of a tennis player running rapidly in the court with many colorful spectators. At medium or high bit rates, the peak signal-to-noise ratio (PSNR) differences between searching five reference frames and searching only one reference frame are about 0.8, 1.2, and 0.4 dB, respectively, for Foreman, Mobile Calendar, and Stefan. For other test sequences, searching five reference frames does not provide noticeable coding gains, and the rate-distortion curves of searching one reference frame is omitted for clarity. The average PSNR drop of our fast algorithm compared with full search is less than 0.05 dB, so that the full search curve and our fast search curve are hardly distinguishable for each sequence. The degradation of PSNR is only slightly noticeable for Foreman, Hall Monitor, and Mobile Calendar, and is about 0.1–0.2 dB.

Table VII shows the average miss detection rate and the false alarm rate of the optimal reference frames in percentages of MBs. The MVs generated by the full search scheme are used as ground true answers. Miss detection of optimal reference frames terminates the block matching process in the early stage to save computation but runs the risk of video quality degradation. False alarm of optimal reference frames guarantees the same video quality as full search but wastes computation. It is shown that the average miss detection rate is 10.43%, which means about 89.57% of the MBs select the same reference frame as full search. The average false alarm rate is 5.04%, which means 94.96% of the MBs do not require extra unnecessary computation after the best reference frames are found.

Fig. 7 shows the number of average searched frames for the reference software and the proposed algorithm. It is shown that the average number of searched reference frames ranges from 1 to 3.5, which means 30%–80% of ME operations can be saved because the ME operations are in proportion to the number of searched reference frames. We also show the optimal number of searched reference frames, which is defined as

$$0 \cdot \mathrm{Intra} + 1 \cdot \mathrm{Ref0} + 2 \cdot \mathrm{Ref1} + 3 \cdot \mathrm{Ref2} + 4 \cdot \mathrm{Ref3} + 5 \cdot \mathrm{Ref4} \tag{24}$$

where Intra, $\mathrm{Ref0}$, $\mathrm{Ref1}$, $\mathrm{Ref2}$, $\mathrm{Ref3}$, and $\mathrm{Ref4}$ denote the percentages of intra-MBs and inter-MBs predicted by reference

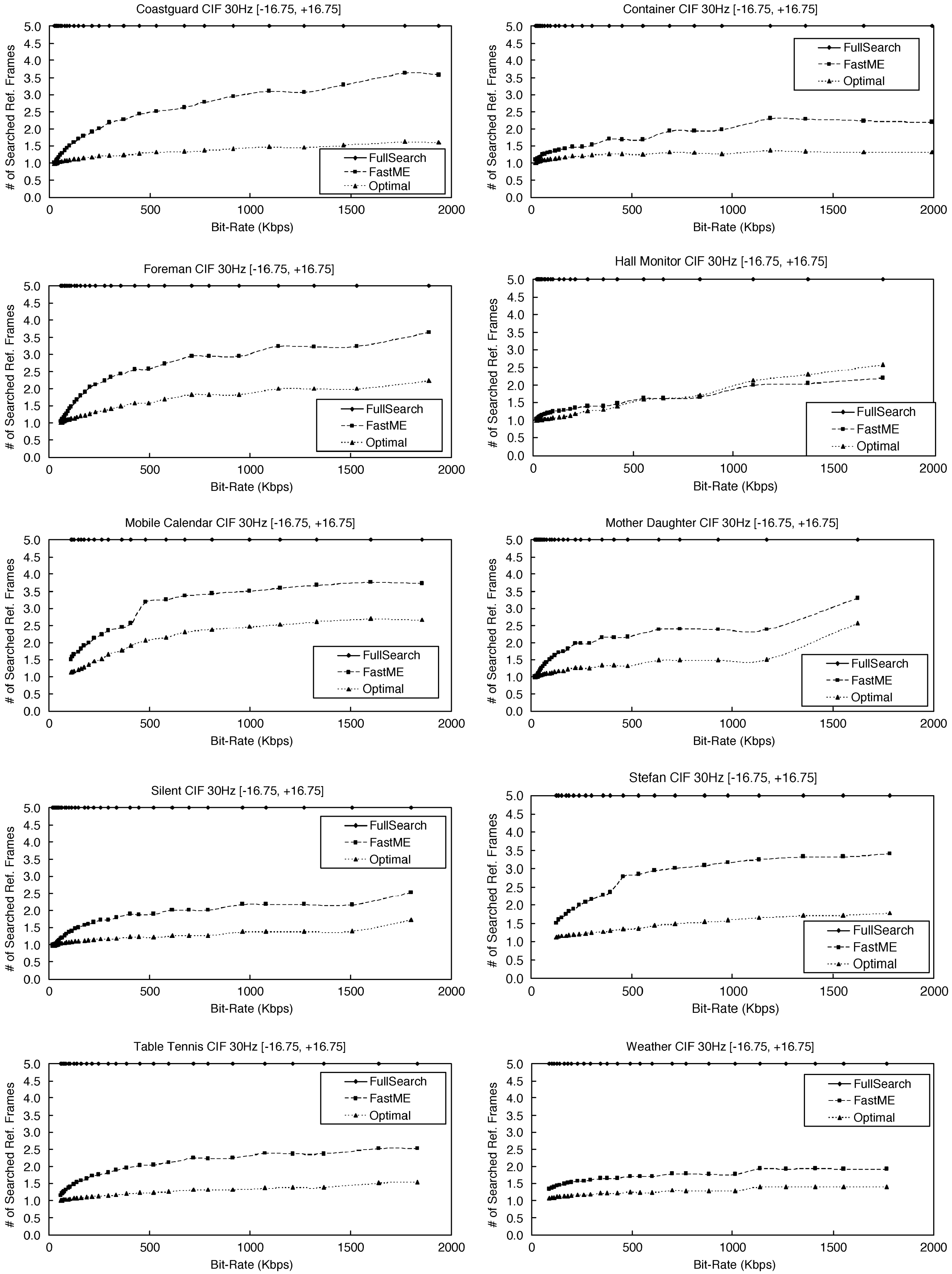


Fig. 7. Number of searched reference frames.

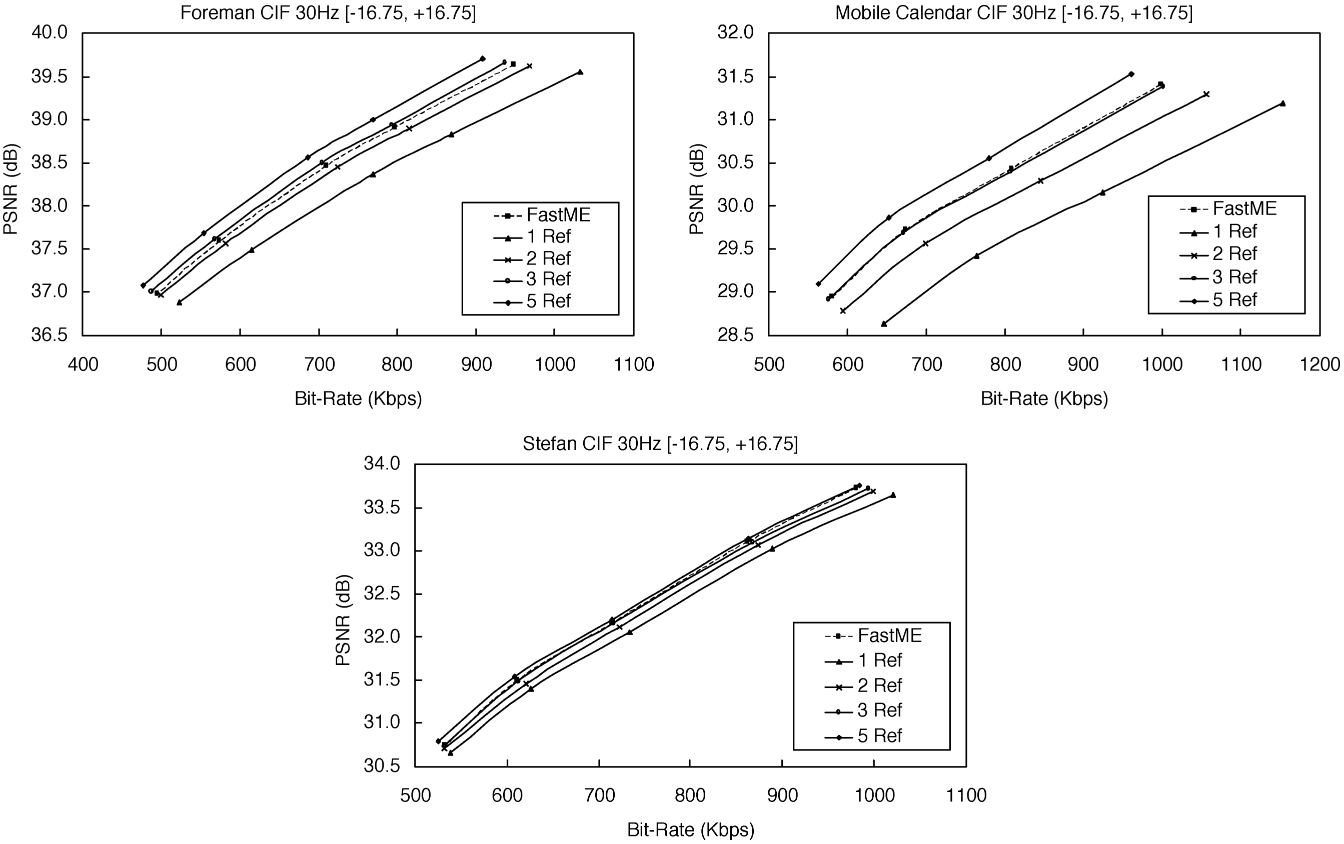


Fig. 8. Comparison with fixed reference frames.

frames 0–4, respectively, by using the reference software. The optimal curves stand for the lower limits of computation that do not suffer any loss of video quality (suppose full search is applied for every reference frame). The gap between the curve of our algorithm and that of the optimal case indicates what we can still improve for more effective reference frame selection. For more motionless sequences, such as Container, Hall Monitor, Mother and Daughter, Silent, and Weather, the numbers of searched reference frames are close to the optimal ones. However, for other sequences with higher motion activity, there is still room for further improvement.

Fig. 8 shows the comparison between the rate-distortion performance of our fast algorithm and those of fixed reference frames. Only Foreman, Mobile Calendar, and Stefan are shown here because for other sequences, multiple reference frames cannot contribute to noticeable PSNR gains. Please note that "2 Ref" and "3 Ref" denote 60% and 40% of computation reductions, respectively. The rate-distortion performance of our algorithm is very close to that of "3 Ref" and is better than "2 Ref." As shown in Fig. 7, our algorithm results in 2.5–3.3 reference frames for the three video sequences. In sum, for cases that multiple reference frames are helpful in compression, "fixed 2 reference frames" provides worse video quality with less complexity, and "fixed 3 reference frames" provides similar video quality with similar complexity. However, for cases that multiple reference frames are useless (low bit rates or other seven sequences), our algorithm can lead to much lower complexity than fixed reference frames. In other words, our algorithm successfully detects when we should or should not search more frames.

Fig. 9 shows the rate-distortion curves and the rate-computation curves for three more test sequences that were not used in the statistics. All the parameters and threshold values are kept the same. Two of the video clips are extracted from the action movies, "Taxi-I" and "Crouching Tiger & Hidden Dragon." The motion fields of the three test sequences are all very large and complex with moving camera. Simulation results show that our algorithm still performs well in these cases. Most unnecessary calculations of full search scheme are detected. Please note that the optimal number of reference frames for "Taxi-I" is smaller than one because the movie clip has great high motion making the encoder select a lot of intra-MBs.

Now we compare our algorithm with other works. In general, our fast reference frame selection scheme can save a lot of unnecessary ME operations while maintaining the video quality almost identical to full search. Our first idea was published in [14]. It is different from the conventional fast block matching algorithms, such as three-step search [15], one-dimensional full search [16], four-step search [17], diamond search [18]–[20], partial distortion elimination [21], successive elimination [22], and global elimination [23]. The main concepts of conventional methods are decimation of search positions, prediction of MVs, simplification of matching criterion, and early termination of unnecessary SAD calculations. Many researchers [24]–[27] developed their fast algorithms for ME in H.264. Although the concepts are old, the results are quite promising. In [28] and [29], they both proposed the concept of MV composition. MVs of different frames are highly correlated and can be used to predict the MVs for farther reference frames. Fortunately, our algorithm and other fast algorithms are orthogonal. The multi-

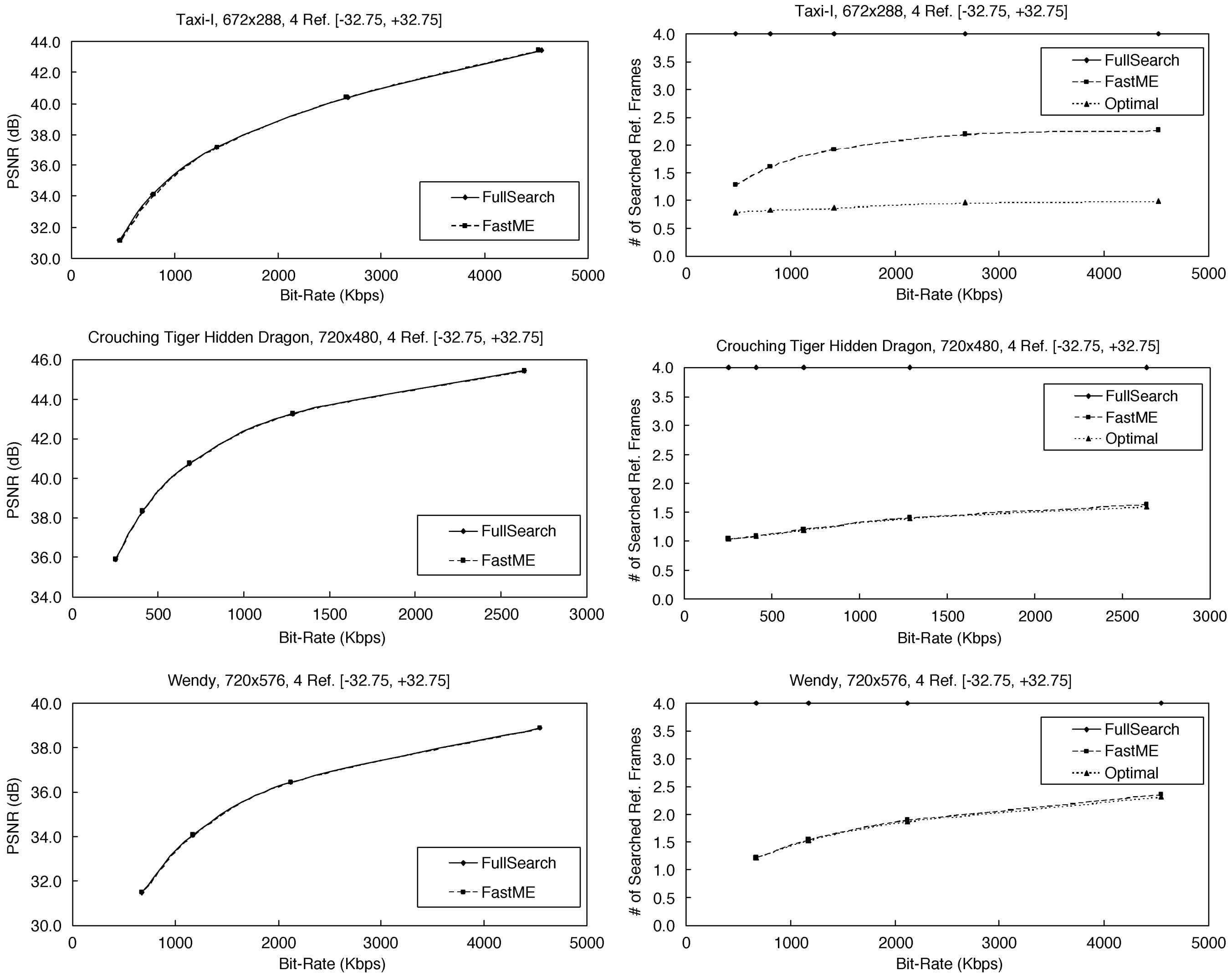


Fig. 9. Experimental results of other test sequences.

frame ME procedure can combine different algorithms as follows. Given the first reference frame, we can adopt the conventional fast search methods. Then, our reference selection criteria can be applied to determine if the remaining frames should be skipped or not. Finally, if it is necessary to keep on searching the next frame, the new concept of MV composition provides a very good solution to further speed up the block matching process for farther reference frames.

VI. Conclusion

We proposed a simple and effective fast algorithm for multiple reference frames ME. First, the available information after intra-prediction and ME from the previous reference frame is analyzed. Then, detection of all-zero residues, SKIP mode and complexity of texture, sampling defects, and MV inconsistency are applied to determine if it is necessary to search more frames. Experimental results showed that our method can save 30%–80% of ME computation depending on the sequences while keeping the quality nearly the same as full search scheme. Besides, our scheme can be easily combined with other conventional fast algorithms to achieve better performance.

References

[1] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Rec. H.264 and ISO/IEC 14 496-10 AVC, Joint Video Team, Mar. 2003.

[2] *Information Technology—Coding of Audio-Visual Objects—Part 2: Visual*, ISO/IEC 14 496-2, 1999.

[3] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263, 1998.

[4] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13 818-2 and ITU-T Rec. H.262, 1996.

[5] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. J. Sullivan, "Performance comparison of video coding standards using lagragian coder control," in *Proc. IEEE Int. Conf. Image Process.*, 2002, pp. 501–504.

[6] Joint Video Team Reference Software JM8.2 May 2004 [Online]. Available: http://bs.hhi.de/~suehring/tml/download/

[7] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[8] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," in *Proc. IEEE Int. Conf. Image Process.*, 2001, pp. 542–545.

[9] I.-M. Pao and M.-T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 608–616, Jun. 1999.

[10] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[11] A. K. Jain, *Fundamentals of Digital Image Process.*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[12] J. R. Price and M. Rabbani, "Biased reconstruction for jpeg decoding," *IEEE Signal Process. Lett.*, vol. 6, pp. 297–299, Dec. 1999.

[13] R. C. Reininger and J. D. Gibson, "Distributions of two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. 31, pp. 835–839, Jun. 1983.

[14] Y.-W. Huang, B.-Y. Hsieh, T.-C. Wang, S.-Y. Chen, S.-Y. Ma, C.-F. Chen, and L.-G. Chen, "Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2003, pp. 145–148.

[15] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. National Telecommun. Conf.*, 1981, pp. C9.6.1–C9.6.5.

[16] M.-J. Chen, L.-G. Chen, and T.-D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 5, pp. 504–509, Oct. 1994.

[17] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.

[18] S. Zhu and K. —. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Proc. Int. Conf. Inf., Commun. Signal Process.*, 1997, pp. 292–296.

[19] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, Aug. 1998.

[20] A. M. Tourapis, M. L. Liou, O. C. Au, G. Shen, and I. Ahmad, "Optimizing the mpeg-4 encoder—advanced diamond zonal search," in *Proc. IEEE Int. Symp. Circuits Systems*, 2000, pp. 647–677.

[21] ITU-T Rec. H.263 Software Implementation. Telenor R&D Digital Video Coding Group, 1995.

[22] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.

[23] Y.-W. Huang, S.-Y. Chien, B.-Y. Hsieh, and L.-G. Chen, "An efficient and low power architecture design for motion estimation using global elimination algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2002, pp. 3120–3123.

[24] W. I. Choi, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," in *Proc. IEEE Int. Conf. Image Process.*, 2003, pp. 371–374.

[25] H. Y. C. Tourapis and A. M. Tourapis, "Fast motion estimation within the H.264 codec," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2003, pp. 517–520.

[26] C.-H. Kuo, M. Shen, and C.-C. J. Kuo, "Fast inter-prediction mode decision and motion search for H.264," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2004, pp. 663–666.

[27] P. Yang, Y.-W. He, and S.-Q. Yang, "An unisymmetrical-cross multi-resolution motion search algorithm for MPEG-4 AVC/H.264 coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2004, pp. 531–534.

[28] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2004, pp. 695–698.

[29] M.-J. Chen, Y.-Y. Chiang, H.-J. Li, and M.-C. Chi, "Efficient multi-frame motion estimation algorithms for MPEG-4 AVC/JVT/H.264," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. 737–740.

Yu-Wen Huang was born in Kaohsiung, Taiwan, R.O.C., in 1978. He received the B.S. degree in electrical engineering and the Ph. D. degree from the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 2000 and 2004, respectively.

He joined MediaTek, Inc., Hsinchu, Taiwan, R.O.C., in 2004, where he develops integrated circuits related to video coding systems. His research interests include video segmentation, moving object detection and tracking, intelligent video coding technology, motion estimation, face detection and recognition, H.264/AVC video coding, and associated VLSI architectures.

Bing-Yu Hsieh received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 2001 and 2003, respectively.

He joined MediaTek, Inc., Hsinchu, Taiwan, R.O.C., in 2003, where he develops integrated circuits related to multimedia systems and optical storage devices. His research interests include object tracking, video coding, baseband signal processing, and VLSI design.

Shao-Yi Chien (M'03) was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, in 1999 and 2003, respectively.

During 2003 to 2004, he was a Research Staff Member in Quanta Research Institute, Tao Yuan Shien, Taiwan. In 2004, he joined the Graduate Institute of Electronics Engineering, Department of Electrical Engineering, National Taiwan University, as an Assistant Professor. His research interests include video segmentation algorithm, intelligent video coding technology, image processing, computer graphics, and associated VLSI architectures.

Shyh-Yih Ma received the B.S.E.E, M.S.E.E, and Ph.D. degrees from National Taiwan University, Taipei, Taiwan, R.O.C., in 1992, 1994, and 2001, respectively.

He joined Vivotek Inc., Taipei, Taiwan, R.O.C., in 2000, where he developed multimedia communication systems on DSPs. His research interests include video processing algorithm design, algorithm optimization for DSP architecture, and embedded system design.

Liang-Gee Chen (S'84–M'86–SM'94–F'01) received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1979, 1981, and 1986, respectively.

In 1988, he joined the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. During 1993–1994, he was a Visiting Consultant in the DSP Research Department, AT&T Bell Laboratories, Murray Hill, NJ. In 1997, he was a Visiting Scholar in the Department of Electrical Engineering, University of Washington, Seattle. Currently, he is a Professor at National Taiwan University. His current research interests are DSP architecture design, video processor design, and video coding systems.

Dr. Chen has served as an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology since 1996, as Associate Editor of the IEEE Transactions on VLSI Systems since 1999, and as Associate Editor of IEEE Transactions on Circuits and Systems Part II: Express Briefs since 2000. He has been the Associate Editor of the *Journal of Circuits, Systems, and Signal Processing* since 1999, and a Guest Editor for the *Journal of Video Signal Processing Systems*. He is also the Associate Editor of the Proceedings of the IEEE. He was the General Chairman of the 7th VLSI Design/CAD Symposium in 1995 and of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He is the Past-Chair of Taipei Chapter of IEEE Circuits and Systems (CAS) Society, and is a Member of the IEEE CAS Technical Committee of VLSI Systems and Applications, the Technical Committee of Visual Signal Processing and Communications, and the IEEE Signal Processing Technical Committee of Design and Implementation of SP Systems. He is the Chair-Elect of the IEEE CAS Technical Committee on Multimedia Systems and Applications. During 2001–2002, he served as a Distinguished Lecturer of the IEEE CAS Society. He received Best Paper Awards from the R.O.C. Computer Society in 1990 and 1994. Annually from 1991 to 1999, he received Long-Term (Acer) Paper Awards. In 1992, he received the Best Paper Award of the 1992 Asia-Pacific Conference on Circuits and Systems in the VLSI design track. In 1993, he received the Annual Paper Award of the Chinese Engineers Society. In 1996 and 2000, he received the Outstanding Research Award from the National Science Council, and in 2000, the Dragon Excellence Award from Acer. He is a Member of Phi Tan Phi.