A Mode and Search Range Decision Scheme for Fast Motion Estimation in H.264

Chien-Li Lin and Jin-Jang Leou
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan, Republic of China

Abstract: In this study, a mode and search range decision scheme for fast motion estimation in H.264 is proposed. The proposed scheme is used to speed up the motion estimation (ME) procedure in H.264 with slight video quality degradation. For a general video sequence, most of the macroblocks (MBs) belonging to the background (foreground) in a video frame are usually the "background" (foreground) MBs in the next video frame. Thus, the minimum matching errors of MBs in a video frame may be predicted by that of the corresponding MBs in the previously-encoded video frame so that the ME operation of an MB may be performed over a reduced set of inter modes and a reduced search range.

Based on the QP value and the minimum matching errors of the MBs within the previously-encoded video frame, two thresholds, $T_{background}$ and $T_{foreground}$, are use to classify each MB into one of three classes. Different sets of inter modes will be selected for different MB classes. Additionally, if the maximum magnitude of the x and y components of the MV(s) of the variable size block(s) within the corresponding MB in the previously-encoded video frame of an MB in the current video frame, denoted as MV_{mm}, can be used to determine the reduced search range, $R_{reduced}$, of its integer pixel ME operation. Compared with five fast motion estimation algorithms, the proposed scheme usually has the best performance (the minimum ME processing time) in most simulation cases.

Key-Words: *mode and search range decision, fast motion estimation, H.264 video, minimum matching error, full search algorithm, reduced search range.*

1. Introduction

In video coding, temporal redundancy within consecutive video frames can be reduced by motion estimation (ME) and compensation. ME is usually performed by searching the best motion vector (MV), which is the displacement of the coordinate of the best similar block within a given search range in the previously-encoded video frame for a block in the current video frame using some block distortion measure [1]. The simplest way of performing block-based ME is to apply the so-called full search (FS) algorithm [2], in which the best MV is fully searched within a given search range. Because the FS ME procedure may consume up to 80% of computational power of the H.264 encoder, a fast ME algorithm is highly desired.

Many fast search ME algorithms have been proposed, which can be classified into four general categories. First, fast ME algorithms using search patterns contain the three-step search (TSS) algorithm and its modified version [3], the four-step search (4SS) algorithm [4], the diamond search (DS) algorithm combining the 4SS and block-based gradient descent search (BBGDS) algorithms [5], the hexagon-based search (HEXBS) algorithm [6], the adaptive rood pattern search (ARPS) algorithm and its modified version [7], …, etc. Second, fast ME algorithms using subsampled pixels contain, for example, the algorithm proposed in [8].

+This work was supported in part by National Science Council and Ministry of Economic Affairs, Republic of China under Grants NSC 93-2213-E-194-016 and 93-EC-17-A-02-S1-032.

Third, fast zonal search ME algorithms contain, for example, the advanced predictive diamond zonal search (APDZS) algorithm [9]. In this study, a mode and search range decision scheme for fast motion estimation in H.264 is proposed.

This paper is organized as follows. The proposed scheme is addressed in Section 2. Simulation results are included in Section 3, followed by concluding remarks.

2. Proposed Scheme

In H.264 [2], the MV of a variable size block is determined by ME on integer pixel positions, followed by fractional pixel refinement. This will return an MV that minimizes the matching error

$$J(\mathbf{m}, \lambda_{MOTION}) = SAD(s, c(\mathbf{m})) + MV\_COST(\mathbf{m}, \lambda_{MOTION}), \quad (1)$$
$$MV\_COST(\mathbf{m}, \lambda_{MOTION}) = \lambda_{MOTION} \cdot R(\mathbf{m} - \mathbf{p}),$$

where $\mathbf{m}=(m_x, m_y)^T$ is the MV, $\mathbf{p}=(p_x, p_y)^T$ is the predictive MV, and λ_{MOTION} is the Lagrange multiplier. The rate term $R(\mathbf{m-p})$ represents the motion information only, which is computed by a table-lookup, and *SAD* is given by

$$SAD(s, c(\mathbf{m})) = \sum_{x=1}^{B_1} \sum_{y=1}^{B_2} |s[x, y] - c[x - m_x, y - m_y]|,$$

where s is the original video signal, c is the coded video signal, and B_1 and B_2 are the vertical and horizontal dimensions of the examined block. In H.264 JM84 [2], a hybrid Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm for fast integer pixel ME is proposed [10]. Cooperating with the fast motion estimation (FME) algorithm, namely, UMHexagonS, in H.264 [2], [10], the proposed scheme is used to reduce the ME processing time in H.264.

2.1 Two Observations

Suppose that a variable size block is some part of the background and its MV lies in the range of [-3/4, +3/4]. For this case, although the best integer MV of the variable size block may be (-1, -1), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 1), (-1, 0), (1, 0), and (0, 0), in this study, to reduce the ME processing time, the "best" integer MV of the variable block size is directly set to (0, 0). That is because the same MV at 1/2-pixel (or 1/4-pixel) accuracy will be obtained in the fractional pixel ME procedure.

For the FS algorithm in H.264 (JM-FS) [2], the best MV of a variable size block is searched over seven inter modes within a given search range. It can be found that if the best MV of a variable size block is a small one, the corresponding *MV_COST* value is also a small one, i.e., if a macroblock (MB) is some part of the background, then (1) the best MV of the MB is usually determined in inter mode 1 (INTER16×16), (2) the best matching error is a small one, and (3) the MB is very similar to the "best" corresponding reference MB in the previously-encoded video frame, resulting in a small matching error. In this study, if the minimum matching error of an MB is a small one, the MB is determined as a part of the background. On the contrary, if the sum of the minimum matching errors of the best MVs for variable size blocks within an MB is large, the MB is determined as a part of the foreground and the best MVs of the MB are usually determined within inter modes 4, 5, 6, and 7, i.e., INTER8×8, INTER8×4, INTER4×8, and INTER4×4.

2.2 Proposed Mode and Search Range Decision Scheme

For a general video sequence, except the case of abrupt scene change or very high

motion objects, two consecutive video frames are similar, i.e., most of the MBs belonging to the background (foreground) in a video frame are usually the "background" (foreground) MBs in the next video frame. Thus, the minimum matching error of an MB in a video frame may be predicted by that of the corresponding MB in the previous video frame so that the ME operation of an MB can be performed over a reduced set of inter modes and a reduced search range. If the minimum matching errors of all the MBs within a previous video frame are over a range ($J_{\min}(\mathbf{m}, \lambda_{MOTION})$, $J_{\max}(\mathbf{m}, \lambda_{MOTION})$) two thresholds, $T_{\text{background}}$ and $T_{\text{foreground}}$, are empirically defined as

$$T_{\text{background}} = (m_1 + m_2 + m_3)/3 + F_1(QP), \quad (2)$$

$$T_{\text{foreground}} = (M_1 + M_2 + M_3)/3 - F_2(QP), \quad (3)$$

where m_1, m_2, m_3, M_1, M_2, and M_3 are the three smallest and largest values of the corresponding minimum matching errors of the MBs within the previously-encoded video frame, and $F_1(QP)$ and $F_2(QP)$ are two functions related to the quantization parameter *(QP)*. Based on the simulation results shown in Tables 1 and 2, $F_1(QP)$ and F_2(QP) are empirically given by

$$F_1(QP) = (QP/4) \cdot 300 + k \cdot 100, \quad (4)$$

$$F_2(QP) = (M_1 + M_2 + M_3)/3 \cdot (9/10), \quad (5)$$

where k is a weight to adjust the trade-off between video quality *(PSNR)* and ME processing time. By Table 1, using the larger $F_1(QP)$, the proposed scheme can speed up the ME processing time as *QP* increases. Because the whole range ($J_{\min}(\mathbf{m}, \lambda_{MOTION})$, $J_{\max}(\mathbf{m}, \lambda_{MOTION})$) of the distribution of minimum matching errors will enlarge if *QP* increases, the larger $F_1(QP)$ can be used to maintain the percentage of the "background" MBs. In Eq. (4), $F_1(QP)$ is empirically set to be linearly proportional to *QP*. However, video quality will be degraded if the ME (processing) speed as well as k increase, and vice versa. By Table 2, video quality and the ME (processing) speed for variant *QP*s are approximately not affected by $F_2(QP)$. Thus, $F_2(QP)$ is empirically defined as Eq. (5), which is a constant for variant *QP*s. The two thresholds, $T_{\text{background}}$ and $T_{\text{foreground}}$, determined by the ME information within the current video frame will be used to classify the MBs in the next video frame. As shown in Fig. 1, if the minimum matching error of an MB is less than $T_{\text{background}}$, the MB is classified as a class-1 MB (a "background" MB). If the minimum matching error of an MB is larger than $T_{\text{foreground}}$, the MB is classified as a class-2 MB (a "foreground" MB). Otherwise, the MB is classified as a class-3 MB.

Based on the second observation, the class of an MB in the current video frame can be set to that of the corresponding MB in the previously-encoded video frame. Additionally, for a class-1 MB with its minimum matching error being very small, the ME operation of the MB will be performed over a reduced set of inter modes and a reduced search range.

In this study, the maximum magnitude of the x and y components of the MV(s) of the variable size block(s) within the corresponding MB in the previously-encoded video frame of an MB in the current video frame is denoted as MV_{mm}. For example, for two MVs, (-3, 2) and (-5, 1), $MV_{\text{mm}} = 5$. In this study, the reduced search range, R_{reduced}, of an MB in the current frame, as a function MV_{mm} of the corresponding MB in the previous video frame, is determined as follows. (1) if $MV_{\text{mm}} < 1$, $R_{\text{reduced}} = 1$, i.e., the search window is 3×3 in size, (2) if $1 \le MV_{\text{mm}} < 3$, $R_{\text{reduced}} = 2$, i.e., the search window is 5×5 in size, and (3) if $MV_{\text{mm}} \ge 3$, $R_{\text{reduced}} = MV_{\text{mm}}$.

As a summary, the proposed mode and search range decision scheme is as follows.

Case 1 For a class-1 MB (the background), the FME operations [10] are performed only on INTER16×16, INTER8×8, and INTER4×4, and $R_{reduced}$ is determined by its MV_{mm}.

Case 2 For a class-2 MB (the foreground), the FME operation [10] are performed only on INTER8×8, INTER8×4, INTER4×8 and INTER4×4, and $R_{reduced}$ is the original search range.

Case 3 For a class-3 MB, the FME operations [10] are performed on the six possible inter modes (excluding INTER16×16) and $R_{reduced}$ is set to its MV_{mm}.

3. Simulation Results

Six QCIF video sequences, "*Akiyo*," "*Carphone*," "*Container*," "*Foreman*," "*Hall_monitor*," and "*Salesman*," with different search ranges, *R*, and different number of reference frames, *NRF*, are used to evaluate the performance of the proposed scheme. Each of the six QCIF video sequences consists of an I frame, followed by 99 P frames. Six performance measures are employed in this study, including (1) the average encoding time (ms) per frame, (2) the average integer pixel ME time (ms) per frame, (3) the average number of *SAD* computations per frame, (4) the average number of *MV_COST* computations per frame, (5) the average *PSNR* (dB), and (6) the average number of bits per frame.

To evaluate the performance of the proposed scheme, five existing fast search algorithms and the proposed scheme are implemented: (1) JM_FS, the full search algorithm in H.264 [2] (denoted as JM_FS), (2) the diamond pattern search algorithm [5] (denoted as DS), (3) the hexagon pattern search algorithm [6] (denoted as HEXS), (4) the adaptive rood pattern search algorithm (version 2) [7] (denoted as ARPS2), (5) JM_FME, the fast search algorithm in H.264 (denoted as JM_FME) [10], (6) the proposed scheme (denoted as Proposed). The simulation results for the six QCIF video sequences with *R*=16 and *NRF*=1, 3, 5 of the six fast search algorithms for comparison are listed in Table 3.

4. Concluding Remarks

Based on the simulation results obtained in this study, the two important measures, namely, the average encoding time per frame and the average integer pixel ME time per frame, of the proposed scheme are better than that of the five comparison methods, namely, JM_FS, DS, HEXS, APRS2, and JM_FME, but with slight degradations in average PSNR and average number of bits per frame. The proposed scheme is very effective in reducing the ME processing time in H.264. The proposed scheme may cooperate with other fast ME algorithms (e.g., the motion vector prediction technique in [11]) to reduce further the ME processing time in H.264.

References

[1] F. Dufaus and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 858–876, June 1995.

[2] ITU-T Recommendation H.264/ISO/IEC 11496-10, "Advance Video Coding," Final Committee Draft, Document JVT-F100, Dec. 2002.

[3] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block

motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

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

[5] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. on Image Processing*, vol. 9, no. 2, pp. 287-290, Feb. 2000.

[6] C. Zhu, X. Lin, and L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 5, pp. 349–355, May 2002.

[7] K. K. Ma and G. Qiu, "An improved adaptive rood pattern search for fast block-matching motion estimation in JVT/H.26L," in *Proc. of IEEE Int. Symposium on Circuits and Systems,* May 2003, pp. 708-711.

[8] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 3, no. 2, pp. 148-157, April 1993.

[9] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 934-947, Oct. 2002.

[10] Z. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Document JVT-F017, the 6th meeting, Awaji Island, JP, Dec. 2002.

[11] A. Chang, P. H. W. Wong, Y. M. Yeung, and O. C. Au, "Fast integer motion estimation for H.264 video coding standard," in *Proc. of IEEE Int. Conference on Multimedia and Expo*, June 2004, pp. 289-292.

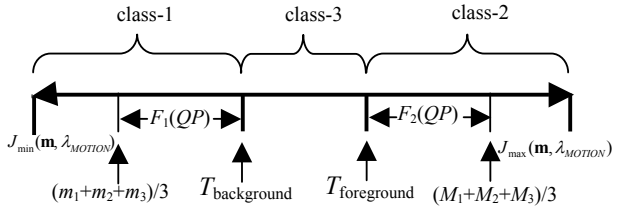

Fig. 1. The distribution of minimum matching errors of the MBs within a video frame.

Table 1. For a fixed $F_2(QP)$, the encoding results (total no. of bits, average *PSNR*, and total ME time/total encoding time) of the QCIF "*Foreman*" sequence for variant *QP*s and $F_1(QP)$s.

QP	$F_1(QP)$	$F_2(QP)$	Total no. of bits (bits)	Average *PSNR* (dB)	Total ME time/total encoding time
16	1200	-400	2517176	45.53	37.61%
16	1600	-400	2582016	45.51	36.11%
16	2000	-400	2629216	45.49	29.62%
20	1200	-400	1464432	42.68	49.18%
20	1600	-400	1515408	42.66	42.69%
20	2000	-400	1556088	42.64	35.53%
24	1200	-400	833040	39.95	60.65%
24	1600	-400	875736	39.89	48.09%
24	2000	-400	905536	39.86	45.66%
28	1200	-400	488456	37.67	71.01%
28	1600	-400	499880	37.52	59.35%
28	2000	-400	521168	37.44	52.01%
32	1200	-400	309288	35.64	74.01%
32	1600	-400	304440	35.49	71.54%
32	2000	-400	303608	35.33	62.62%
36	1200	-400	212272	33.78	74.39%
36	1600	-400	208608	33.59	74.22%
36	2000	-400	204072	33.56	70.63%
40	1200	-400	154680	32.02	77.85%
40	1600	-400	153824	31.97	75.97%
40	2000	-400	149480	31.88	70.66%

Table 2. For a fixed $F_1(QP)$, the encoding results (total no. of bits, average *PSNR*, and total ME time/total encoding time) of the QCIF "*Foreman*" sequence for variant *QP*s and $F_2(QP)$s.

QP	$F_1(QP)$	$F_2(QP)$	Total no. of bits (bits)	Average *PSNR* (dB)	Total ME time/total encoding time
16	800	-500	2415696	45.57	51.59%
16	800	-400	2413752	45.57	50.41%
16	800	-300	2418312	45.57	49.10%
22	800	-500	1081888	41.51	63.96%
22	800	-400	1084248	41.51	65.65%
22	800	-300	1086336	41.50	62.67%
28	800	-500	489048	37.81	73.44%
28	800	-400	489512	37.78	74.59%
28	800	-300	486808	37.81	74.25%
34	800	-500	258840	34.83	76.96%
34	800	-400	260024	34.85	78.63%
34	800	-300	258224	34.80	76.90%
40	800	-500	157504	32.10	79.36%
40	800	-400	157504	32.09	78.97%
40	800	-300	157864	32.11	79.21%

Table 3. The simulation results of six video sequences with *R*=16 and *NRF*=1, 3, and 5 of six fast search algorithms for comparison.

Method	Average encoding time (ms)	Average ME time (ms)	Improve	No. of *SAD*s (10^6)	No. of *MV_COST*s (10^6)	Average *PSNR* (dB)	Bits
NRF=1							
JM_FS	626.97	371.76	—	23.95	437.60	38.06	2568.00
DS	510.51	252.47	28.73%	26.77	8.85	38.04	2583.71
HEXS	507.28	249.79	29.47%	20.82	6.88	38.03	2618.13
APRS2	507.34	244.89	31.00%	15.17	5.07	37.97	2769.21
JM_FME	240.36	34.97	90.61%	19.99	17.18	38.05	2547.29
Proposed	166.91	21.40	94.31%	14.87	21.42	38.01	2582.11
NRF=3							
JM_FS	1800.72	1185.04	—	1489.75	3248.88	38.10	2419.81
DS	1376.12	751.53	33.47%	79.49	26.27	38.08	2446.96
HEXS	1365.42	735.92	34.86%	61.82	20.43	38.07	2483.29
APRS2	1357.61	724.34	35.95%	45.48	15.25	38.00	2636.55
JM_FME	580.14	100.22	91.69%	53.22	39.23	38.10	2416.71
Proposed	362.99	53.13	95.69%	32.44	34.47	38.07	2476.23
NRF=5							
JM_FS	3008.01	2032.07	—	2528.03	5359.55	38.16	2405.89
DS	2239.38	1248.84	35.10%	131.14	43.35	38.13	2427.60
HEXS	2223.11	1227.46	36.12%	101.97	33.71	38.11	2450.07
APRS2	2197.13	1200.79	37.54%	76.34	25.55	38.04	2608.85
JM_FME	902.06	159.65	92.26%	86.07	60.21	38.14	2388.79
Proposed	555.64	83.20	96.05%	49.38	46.46	38.10	2453.19