Complexity Control of Fast Motion Estimation in H.264/MPEG-4 AVC with Rate-Distortion-Complexity optimization

Wu, Mo; Forchhammer, Søren; Aghito, Shankar Manuel

Published in:
Visual Communications and Image Processing 2007

Publication date:
2007

Document Version
Early version, also known as pre-print

Citation (APA):
Wu, M., Forchhammer, S., & Aghito, S. M. (2007). Complexity Control of Fast Motion Estimation in H.264/MPEG-4 AVC with Rate-Distortion-Complexity optimization. In Visual Communications and Image Processing 2007: Proc. of SPIE-IS&T Electronic Imaging (Vol. SPIE Vol. 6508, 650824, pp. 1-11). San Jose, CA, USA: IS&T and SPIE.

Complexity Control of Fast Motion Estimation in H.264/MPEG-4 AVC with Rate-Distortion-Complexity optimization

Mo Wu[†], Søren Forchhammer and Shankar M. Aghito

Department of Communications, Optics and Materials, COM•DTU, 2800 Lyngby, Denmark

ABSTRACT

A complexity control algorithm for H.264 advanced video coding is proposed. The algorithm can control the complexity of integer inter motion estimation for a given target complexity. The Rate-Distortion-Complexity performance is improved by a complexity prediction model, simple analysis of the past statistics and a control scheme. The algorithm also works well for scene change condition. Test results for coding interlaced video (720 x576 PAL) are reported.

Keywords: H.264/AVC, Fast motion estimation, complexity control and inter motion estimation

1. INTRODUCTION

The recent H.264/MPEG-4 part 10 advanced video coding (AVC) standard[1, 2] has improved encoding performance roughly by a factor of 2 compared to previous MPEG standards.[5] The gain in coding efficiency comes at the price of a significant increase in encoding complexity, mainly due to new features in the inter motion estimation stage, which supports variable block sizes and multiple reference frames. Fast motion estimation may be applied to reduce complexity. To control the computational load and make optimal use of the available processing power, a control mechanism is desirable for real-time applications. To achieve this a computational complexity control is designed aiming at maintaining the rate-distortion (R-D) performance as much as possible with the given computational power. The new control method is based on rate-distortion-complexity (R-D-C) optimization.

The Enhanced Predictive Zonal Search (EPZS) fast motion estimation algorithm[3] for integer inter motion estimation was implemented with some modifications for the H.264 reference software encoder in a previous work.[4] Lagrangian optimization was applied to the extended EPZS algorithm, which is considered as a good fast motion estimation solution for the H.264/AVC codec. The video coding performance is traditionally measured by R-D performance. When complexity is an important factor, the performance may be measured in terms of R-D-C. An operational method for analyzing the optimal selection of the macroblock (MB) partition mode and number of reference frames in the motion compensation was developed.[4] The basic idea is to transform the three-dimensional problem of concurrently optimizing, e.g. minimizing R-D-C into a more tractable two-dimensional problem. The minor changes in distortion due to the complexity control are converted into small changes in rate by using a local slope of the R-D curve for the chosen parameter setting. This effectively eliminates the distortion parameter, and the optimization problem has thus been reduced to a problem of two parameters, modified rate (R^*) and complexity (C). For complexity control, a solution based on increasing or decreasing the percentage of skipped macroblocks using a Lagrangian R-D-C cost function has been proposed.[6] In the following, Section 2 presents the control scheme in detail and Section 3 presents the experimental results.

[†] Further author information: (Send correspondence to Mo Wu)
Mo Wu: E-mail: mw@com.dtu.dk, Telephone: +45 45 25 36 20

2. MOTION ESTIMATION COMPLEXITY CONTROL

A complexity control algorithm is designed for and integrated in the H.264/AVC reference software with an extended EPZS fast motion estimation[4] method. The algorithm is designed for standard definition TV (SDTV) interlaced video with GOP structure, IBBPBBPBBPBB (IBBP$_{(12)}$). Inside each GOP, there are three BBP structures, hereafter denoted as P-GOPs. The scheme is applied at each P-GOP, by controlling the parameters of the encoder's configuration. In this algorithm, the parameters of the configuration that can be adaptively changed are the partition modes and the reference fields that can be allowed for inter motion estimation.[2] The generalization to other GOP structures is straightforward. Only the integer inter motion estimation complexity is considered.

A limitation on computational resources leads to a bound on the complexity, C. Under this constraint, the R-D performance shall be optimized. Transforming R-D into R^*,[4] the optimization is based on considering the slope, $\frac{\delta R^*}{\delta C}$, where δR^* is the small change of the R^* due to variation of the complexity, δC. Thus when complexity needs to be decreased as part of the control process, it can be done by reducing the search space, and skipping the settings that produce the lowest $|\frac{\delta R^*}{\delta C}|$.

2.1. Calculating the integer inter motion estimation complexity

Before describing the control scheme, notation and the method for calculating the integer inter motion estimation complexity is presented in Sections 2.1.1 and 2.1.2. The same notation is used in the following sections.

2.1.1. Weighted search positions

The inter motion estimation complexity is measured by a weighted number of search positions, which is a simple measurement of the motion search complexity.[4] The number of weighted search positions of different block partitions are counted according to the block size in inter mode motion estimation (see Table 1).

Table 1. *Weighted search positions of different block partitions per search*

Block partition	16×16	16×8	8×16	8×8	8×4	4×8	4×4
Weighted search positions per search	16	8	8	4	2	2	1

The parameter indicating the partition modes for frame i is $b(i)$, which is defined by

$$b(i) = \begin{cases} 1, & \textit{block partition } 16 \times 16 \\ 2, & \textit{block partition down to } 8 \times 8 \\ b_{max} = 3, & \textit{all block partitions.} \end{cases} \quad (1)$$

The method is designed for interlaced video, so one frame includes two fields. The parameter indicating the reference fields for frame i is $r(i)$. For $r(i) = 1$, if the i_{th} frame is a P frame, the search is restricted to the first two reference fields in the reference list[2] for forward prediction; if the i_{th} frame is a B frame, an additional frame is used for backward prediction.[2] As $r(i)$ increases by one, an extra reference field for forward prediction can be searched. In the experiments, the maximum number of the reference frames is five, so the maximum value of $r(i)$, $r_{max} = 9$.

Thus the weighted search positions in frame i are a function of $b(i)$ and $r(i)$, and indicated as $C_{PF}(b(i), r(i), i)$ and $C_{BF}(b(i), r(i), i)$ for P and B frames, respectively, if the other parameters are left unchanged throughout the encoding process.

For a window of M frames, we define the total complexity,

$$C = \sum_{i \,\in\, \text{P frames}}^{M} C_{PF}(b(i), r(i), i) + \sum_{i \,\in\, \text{B frames}}^{M} C_{BF}(b(i), r(i), i), \quad (2)$$

where

$$\begin{cases} C_{PF}(b(i),r(i),i) & = \sum_{b=1}^{b(i)} \sum_{r=1}^{r(i)} C_P(b,r,i) \\ C_{BF}(b(i),r(i),i) & = \sum_{b=1}^{b(i)} \sum_{r=1}^{r(i)} C_B(b,r,i), \end{cases} \tag{3}$$

where $C_P(b,r,i)$ and $C_B(b,r,i)$ are the number of weighted search positions searching on the r_{th} reference fields and b_{th} partition modes (shown in Eq. (4)) in the i_{th} frame for a P and B frame, respectively. In the sum in Eq. (2), the notation i ∈ P frames is not strictly correct, since I frames, where one field is encoded with inter motion estimation, should also be included in the sum.

$i \in$ P frame is a general restriction, it may include the I frame i, if frame i has one field, which is also encoded with inter motion estimation.

$$b = \begin{cases} 1, & partition: 16 \times 16 \\ 2, & partitions: 16 \times 8,\ 8 \times 16\ \ and\ \ 8 \times 8 \\ 3, & partitions: 8 \times 4,\ 4 \times 8\ \ and\ \ 4 \times 4. \end{cases} \tag{4}$$

2.1.2. Weighted number of coded blocks

Later we shall evaluate the effect of decreasing the number of motion searches. This could be based on the occurrence of actually chosen combinations so far. $N_P(b,r,i)$ and $N_B(b,r,i)$ are the weighted number of coded blocks that are encoded by referring to the r_{th} reference fields and with the partition mode selected from the b_{th} partition modes for frame i, which is P and B frame, respectively. Because of SKIP mode, direct mode, bi-prediction etc of H.264/AVC codec, the counting method of $N_P(b,r,i)$ and $N_B(b,r,i)$ are not as simple as that of $C_P(b,r,i)$ and $C_B(b,r,i)$. In a P frame's skip mode or a B frame's direct mode, there is no coded motion vectors. The encoder and the decoder will use the same predicted motion vectors (MVs), constructed from previously compressed MVs. Thus, except for the compression of MVs, the same motion compensation scheme is applied. Thus it is reasonable to count them into $N_P(b,r,i)$ or $N_B(b,r,i)$. For collecting the statistics, the counters go through all the macroblocks, and for each macroblock, the counting method is shown in Figure 1.

2.2. Complexity prediction model

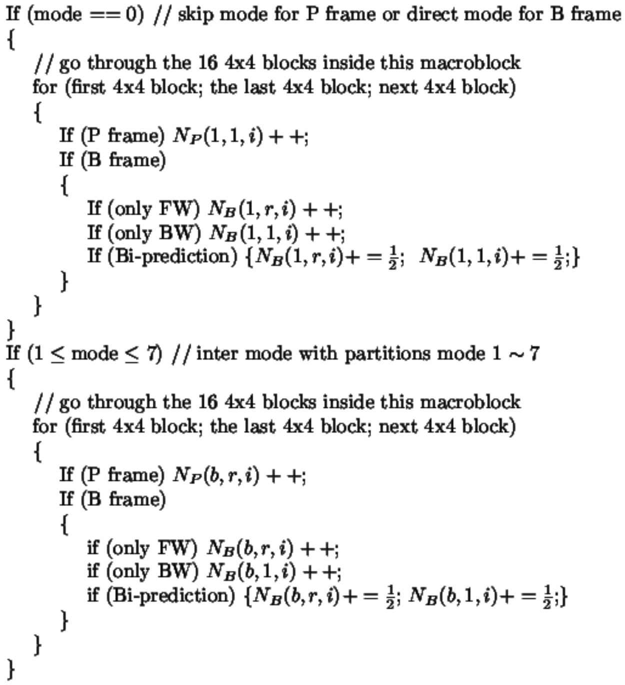
In the control scheme, a given target weighted search positions per second, C_{T}, is provided for a window of M frames. The controller may increase or decrease the number of search positions according to C_{T}. Thus, a motion search complexity prediction model is required to predict the complexity that will be used for coding the following frames with different configurations, and select the appropriate parameters. The model below is applied for adjusting the settings when a (small) increase in complexity is possible and desired.

Through the analysis of the statistics of three training sequences, Mobcal, Cycling and Barcelona, an elaborate method is applied to predict the complexity of the next P-GOP for a given configuration. According to terms used in the previous sections, the complexity of the P-GOP starting at the i_{th} frame, e.g. the index of the first B frame is i, is defined as

$$C_{BBP}(b(i),r(i),b(i+2),r(i+2),i) = \sum_{j=i}^{i+1} C_{BF}(b(j),r(j),j) + C_{PF}(b(i+2),r(i+2),i+2), \tag{5}$$

where $b(i+1) = b(i)$ and $r(i+1) = r(i)$, i.e. the two B frames have the same settings, $b(i)$ and $r(i)$, while $b(i+2)$ and $r(i+2)$ indicate the settings for the P frame. In the following, for a shorter notation, $b(i)$, $r(i)$, $b(i+2)$ and $r(i+2)$ are replaced by b'_i, r'_i, b_i and r_i, respectively. When only the 16x16 partition mode and one reference frame are used for both P and B frames in a P-GOP, the complexity can be described as,

$$C_{BBP}(1,1,1,1,i) = C_{PF}(1,1,i+2) + \sum_{j=i}^{i+1} C_{BF}(1,1,j). \tag{6}$$



```
If (mode == 0) // skip mode for P frame or direct mode for B frame
{
    // go through the 16 4x4 blocks inside this macroblock
    for (first 4x4 block; the last 4x4 block; next 4x4 block)
    {
        If (P frame) $N_P(1,1,i)++$;
        If (B frame)
        {
            If (only FW) $N_B(1,r,i)++$;
            If (only BW) $N_B(1,1,i)++$;
            If (Bi-prediction) {$N_B(1,r,i) += \frac{1}{2}$;  $N_B(1,1,i) += \frac{1}{2}$;}
        }
    }
}
If ($1 \leq$ mode $\leq 7$) // inter mode with partitions mode $1 \sim 7$
{
    // go through the 16 4x4 blocks inside this macroblock
    for (first 4x4 block; the last 4x4 block; next 4x4 block)
    {
        If (P frame) $N_P(b,r,i)++$;
        If (B frame)
        {
            if (only FW) $N_B(b,r,i)++$;
            if (only BW) $N_B(b,1,i)++$;
            if (Bi-prediction) {$N_B(b,r,i) += \frac{1}{2}$; $N_B(b,1,i) += \frac{1}{2}$;}
        }
    }
}
```

Figure 1. The method of updating the counters, $N_P(b,r,i)$ and $N_B(b,r,i)$ over one macroblock for the frame i. FW refers to forward prediction, BW refers to the backward prediction, and the mode is the same notation as defined in H.264 standard.[1]

Let $\hat{C}(\cdot)$ denotes the prediction of $C(\cdot)$. Let $C_{BBP}(b_n, r_n, b'_n, r'_n, n)$ indicate the complexity of the next P-GOP; based on the previous collected statistics, $C_{BBP}(1,1,1,1,i)$, $C_{BBP}(b_n, r_n, b'_n, r'_n, n)$ is obtained as follows,

$$\begin{aligned}\hat{C}_{BBP}(b_n, r_n, b'_n, r'_n, n) &= \hat{C}_{PF}(b_n, r_n, n+2) + \sum_{j=n}^{n+1} \hat{C}_{BF}(b'_n, r'_n, j) \\ &= (f_{P1}(b_n, r_n, n) + f_{B2}(b'_n, r'_n, n) + 1) \cdot C_{BBP}(1,1,1,1,i), \end{aligned} \tag{7}$$

where

$$f_{P1}(b_n, r_n, n) = \frac{\hat{C}_{PF}(b_n, r_n, n+2) - C_{PF}(1,1,i+2)}{C_{BBP}(1,1,1,1,i)} = \begin{cases} k_0(r_n - 1), & if\ b_n = 1 \\ k_1(r_n - 1) + k_2, & if\ b_n = 2 \\ k_3(r_n - 1) + k_4, & if\ b_n = 3 \end{cases} \tag{8}$$

and

$$f_{B2}(b'_n, r'_n, n) = \frac{\sum_{j=n}^{n+1} \hat{C}_{BF}(b'_n, r'_n, j) - \sum_{j=n}^{n+1} C_{BF}(1,1,j)}{C_{BBP}(1,1,1,1,i)} = \begin{cases} k_5(r'_n - 1), & if\ b'_n = 1 \\ k_6(r'_n - 1) + k_7, & if\ b'_n = 2 \\ k_8(r'_n - 1) + k_9, & if\ b'_n = 3, \end{cases} \tag{9}$$

where $k_0 = 0.168$, $k_1 = 0.56$, $k_2 = 0.6$, $k_3 = 0.82$, $k_4 = 1.05$, $k_5 = 0.4$, $k_6 = 1.34$, $k_7 = 2.0$, $k_8 = 2.0$ and $k_9 = 3.6$, and the mathematic model reflects the linear relation between the search complexity used for P and B frames and r_n and $r^{'}_n$, respectively. Because the model is designed for complexity prediction based on fast motion estimation method, the linear relation between the complexity used and the applied different block partitions does not simply exist. The parameters, k_m, are experimentally determined by collecting the results of

$$\frac{\sum_i C_{PF}(b_i, r_i, i) + \sum_i C_{BF}(b'_i, r'_i, i)}{\sum_i C_{BBP}(1,1,1,1,i)}, \tag{10}$$

for the three test sequences, and then calculate the average value, which is assumed to be expressed by

$$\frac{E[C_{PF}(b_i, r_i, i)] + 2 \cdot E[C_{BF}(b'_i, r'_i, i)]}{E[C_{BBP}(1,1,1,1,i)]}, \tag{11}$$

where $E[\cdot]$ denotes expectation. The statistics are fitted to the model determining parameters, k_m.

Similarly, derived from Eq. (7), $\hat{C}_{BBP}(b_n, r_n, b'_n, r'_n, n)$ can also be predicted based on a given previous P-GOP complexity, $C_{BBP}(b_i, r_i, b'_i, r'_i, i)$

$$\hat{C}_{BBP}(b_n, r_n, b'_n, r'_n, n) \quad = \quad \frac{(f_{P1}(b_n, r_n, n) + f_{B2}(b'_n, r'_n, n) + 1)}{(f_{P1}(b_i, r_i, i) + f_{B2}(b'_i, r'_i, i) + 1)} \cdot C_{BBP}(b_i, r_i, b'_i, r'_i, i), \tag{12}$$

2.3. Measurement of benefit terms

When decreasing the complexity, we evaluate the marginal performance of b and r for P frame and b' and r' (the same definition as b and r, respectively) for B frames, e.g. $b = 2$ compared with $r' = 3$. The aim is to determine values of b or r or b' or r' that do not (currently) efficiently utilize (in terms of R^*) the available computational complexity C. These are found by evaluating, before encoding the current P-GOP, the benefit terms, $\Lambda(\cdot)$, defined as

$$\Lambda_P(b, i) = \quad s_P \cdot B_b(i+2) \cdot \frac{\sum_{r_s} N_P(b, r_s, i+2)}{\sum_{r_s} C_P(b, r_s, i+2)}, \tag{13}$$

$$\Lambda_P(r, i) = \quad s_P \cdot B_b(i+2) \cdot \frac{\sum_{b_s} N_P(b_s, r, i+2)}{\sum_{b_s} C_P(b_s, r, i+2)}, \tag{14}$$

$$\Lambda_B(b', i) = \quad \frac{1}{2} \sum_{j=i}^{i+1} \left(B_b(j) \cdot \frac{\sum_{r_s} N_B(b', r_s, j)}{\sum_{r_s} C_B(b', r_s, j)} \right), \tag{15}$$

$$\Lambda_B(r', i) = \quad \frac{1}{2} \sum_{j=i}^{i+1} \left(B_b(j) \cdot \frac{\sum_{b_s} N_B(b_s, r', j)}{\sum_{b_s} C_B(b_s, r', j)} \right), \tag{16}$$

where $\Lambda_P(b, i)$ and $\Lambda_B(b', i)$ are the benefits selecting b_{th} and $b^{'}_{th}$ partition modes for the P and B frames, respectively, in the previous P-GOP, while $\Lambda_P(r, i)$ and $\Lambda_B(r', i)$ are the benefits selecting r_{th} and $r^{'}_{th}$ reference fields for the P and B frames, respectively. $B_b(i)$ is the number of bits utilized for coding the frame i. The scaling factor $s_P = 2$ is introduced in order to prioritize P frames, which are more important, since they may be used as reference frames for B frames. The maximum number of reference frames/fields is set to $r_{max} = 9$. The ratio, $\frac{\sum_{r_s} N_P(b, r_s, i)}{\sum_{r_s} C_P(b, r_s, i)}$ as an example, could be viewed as the weighted number of blocks per complexity unit that can be improved using the partition modes of b for coding the P picture. The computational load due to motion estimation may be decreased by skipping the search for those combinations of picture type and configuration that correspond to the smallest benefit terms, $argmin\{\Lambda_P(b, i), \Lambda_P(r, i), \Lambda_B(b', i), \Lambda_B(r', i)\}$. This model has been constructed based on experiments.

2.4. Control scheme

For each sequence, *best individual settings*[4] may be determined in a R-D-C sense. If the sequence has 'stationary' properties over the whole sequence, it can be assumed that each segment of the sequence also has the same *best individual settings* and R-D-C performance. Because the R-D performance has been simplified as R*, the $\mathrm{R}^* - \mathrm{C}$ curve decided by *best individual settings* is denoted by $\mathrm{R}^*_{\mathrm{opt}} - \mathrm{C}$. By construction, $\mathrm{R}^*_{\mathrm{opt}}$ is a convex function of C. A simple mathematical model of each segment is obtained as follow: the R* value of the k_{th} segment is denoted as $\Delta\mathrm{R}^{*(k)}_{\mathrm{opt}}$ and it is calculated using the *best individual settings*. $\Delta\mathrm{R}^{*(k)}_{\mathrm{opt}}$ can be viewed as a function of the complexity used in segment k, and the complexity is denoted by $\Delta\mathrm{C}^{(k)}$. i.e., $\Delta\mathrm{R}^{*(k)}_{\mathrm{opt}} = \mathrm{g}(\Delta\mathrm{C}^{(k)})$.

Let us assume 'stationarity', which we shall use for steady state operation (State 0 below), and the sequence is divided in N segments of equal length of frames. If the same setting is utilized in every segment, thus $\Delta C^{(k)} = \Delta C^{(l)}, \forall\ 1 \le k, l \le N, k \ne l$, regardless of the start of the sequence. So,

$$\Delta C^{(k)} = \quad \Delta C = \quad \frac{C}{N}, \tag{17}$$

$$\Delta R^{*(k)}_{opt} = \quad \Delta R^{*}_{opt} = \quad \frac{R^{*}_{opt}}{N}. \tag{18}$$

When the complexity control is applied, the best settings used can be different for each segment of the sequence. In this case the complexity measure for the k_{th} segment, $\Delta C^{'(k)}$, is not constant across the whole sequence. With the complexity control scheme,

$$N \cdot \Delta C = \sum_{i}^{N} \Delta C^{'(k)}. \tag{19}$$

Jensen's inequality states that if $g(x)$ is a convex function, $E[g(X)] \ge g(E[X])$.[8] So,

$$R^{*}_{opt} = N \cdot \Delta R^{*}_{opt} = N \cdot g(\Delta C) = N \cdot g(E[\Delta C^{'(k)}]) \le N \cdot E[g(\Delta C^{'(k)})] = \sum_{k=1}^{N} g(\Delta C^{'(k)}) = \sum_{k=1}^{N} \Delta R^{*'(k)}_{opt} = R^{*}_{opt}{}', \tag{20}$$

where R^{*}_{opt} is the R^{*} obtained by selecting the same best individual settings for each segment, while $\Delta R^{*'(k)}_{opt}$ is the R* obtained in the k_{th} segment by the control scheme, i.e. when different settings are allowed for different segments. Eq. (20) shows: a better $\mathrm{R}^* - \mathrm{C}$ (R-D-C) performance compared with the best fixed setting is not achievable, if the *best individual settings* of the R-D-C are the exact same over the short uniform segments of a 'stationary' video sequence. So, the control scheme not only needs to control the complexity according to the target complexity, but also should avoid dramatically changing the complexity in each P-GOP.

A control scheme (given below) is designed in order to efficiently make use of the available computational power, maintain the R-D (R^*) performance and avoid dramatic changes in complexity. It includes scene change detection and control of the fast motion estimation at scene change. The details of the steady state control process is shown below as the main part of the control scheme in **State 0**. **State 1 - 3** handles transitions at the beginning and after a scene change.

The processing of states 0 - 3 are described in pseudo code below.

State 0: This is the steady state control state. Before encoding one P-GOP, the scene change detector will check if a scene change is taking place, if yes, go to state 1, otherwise stay in state 0.

```
if (scene change detected)
  go to State 1
else
  if (all reference frames are present) // in the first few P-GOPs, the reference frames specified
      by the parameters of the configuration may not be present, e.g. the first P-GOP cannot
    search on the $5_{th}$ reference frame
  {
      if ($C_{\mathrm{P-GOP}} > C^{(i)}_{\mathrm{P-GOPT}}$) // $C_{\mathrm{P-GOP}}$ is the complexity used in the previous P-GOP.
        $C^{(i)}_{\mathrm{P-GOPT}}$ refers to Eq. (23)
      {
        while ($C'_{\mathrm{P-GOP}} + C_{\mathrm{P-GOP}} > 2 \cdot C^{(i)}_{\mathrm{P-GOPT}}$) // $C'_{\mathrm{P-GOP}}$ is the predicted complexity if a
            subset in the setting is switched off
        {
            decrease the complexity by switching off searching on subsets, $b$ or $r$ for P frame or
              $b'$ or $r'$ for B frame selected by Eqs. (13 $\sim$ 16);// for details refer to Section 2.3
        }
    }
    else
    {
        if ($\frac{|C^{(i)}_{\mathrm{P-GOPT}} - C_{\mathrm{P-GOP}}|}{C^{(i)}_{\mathrm{P-GOPT}}} < \delta$) {continue}; // $\delta$ is a threshold to make the control robust
          to the fluctuation of the complexity, $\delta = 0.05$ is applied.
        else {set the setting by complexity prediction;} // increase complexity
    }
}
else // not all reference frames are present
{
    if ($C_{\mathrm{P-GOP}} > C^{(i)}_{\mathrm{P-GOPT}}$) { reset the setting by complexity prediction;}
    else {continue;}
}
```

State 1: After a scene change is detected, the configuration with the 16×16 partition mode and one reference frame is selected for the next P frame. Searching down to 4×4 partition mode and one reference frame is selected for B frames. Finish encoding the current P-GOP, then Go to State 2.

State 2: Configuration with searching down to 4×4 partition mode and one reference frame is selected for both P and B frames. Finish encoding the current P-GOP, then Go to State 3.

State 3: Configuration with searching down to 4×4 partition mode and one reference frame plus one reference field is selected for both P and B frames. Finish encoding the current P-GOP, switch between the block partition priority or reference priority preset orders by comparing $\frac{\sum_{b_s=1}^{b} N_P(b_s,2,i)}{\sum_{b_s=1}^{b} N_P(b_s,1,i)}$ with a predefined threshold, T; set configuration by complexity prediction model; Go to State 0.

In the control process, there is a target complexity, C_T. The complexity control method can adaptively change the target complexity of P-GOP, $C_{\mathrm{P-GOPT}}$, which has linear relation with C_T. The relation between C_T and average target complexity of P-GOP, $\overline{C_{\mathrm{P-GOPT}}}$, is mainly affected by the frame rate and GOP structure. As an approximation, it is assumed that there is no complexity for I frames and complexities of P and B frames are viewed as the same. Thereafter, the relation can be calculated by

$$C_T = \eta \cdot \overline{C_{\mathrm{P-GOPT}}}, \tag{21}$$

$$\eta = \frac{(p-1) \cdot f}{p \cdot p'}, \tag{22}$$

where p is the period of one GOP, p' is the period of one P-GOP and f is the frame rate. In conducting the experiments, the GOP structure is $IBBPBBPBBPBB_{(12)}$, thus $p = 12$, $p' = 3$ and $f = 25$ (frames/s), so $C_T = 7.6389 \cdot \overline{C_{\mathrm{P-GOPT}}}$. In practice, the relation is also slightly affected by C_T and the limited number of settings that can be chosen. Thus $C_{\mathrm{P-GOPT}}$ should be adaptively controlled according to C_T for the test sequences. To achieve this, the target P-GOP complexity after coding the most recent frame, i_{th} frame, is

$$C^{(i)}_{\mathrm{P-GOPT}} = \frac{C_T \cdot M - f \cdot \sum_{j=1}^{j=i} [C_{PF}(b(j), r(j), j) + C_{BF}(b(j), r(j), j)]}{\eta \cdot (M - i)}. \tag{23}$$

3. EXPERIMENTAL RESULTS

The adaptive complexity control method was applied to PAL (720×576 pixels, 25 fps) test sequences. The GOP structure is $\text{IBBP}_{(12)}$. R-D optimization is enabled in the reference software. Direct mode is tested in spatial mode.

The commonly used test sequences Mobcal, Barcelona and Table tennis are utilized (each with 100 frames). The accuracy of the complexity control is shown in Table 2. On the average the proposed scheme controls the integer motion estimation complexity with an accuracy of 0.43% of the target complexity. The complexity is measured in millions of weighted search positions per second (MWSP/s).[4]

The coding performance for Mobcal, Table tennis and Barcelona are shown in Figure 2, where the R*-C curves obtained with the proposed complexity control are compared to those obtained in previous work[4] by using the *best average settings*, the *best individual settings* and the *complexity worst case settings* (for details refer to Appendix A). *best average settings* and *best individual settings* are all obtained using *off-line* optimization of the parameters, and they are acquired by coding the test sequences with all the settings. The *complexity worst case settings* obtained represents a conservative design approach based on the analysis of the complexity worst case. We note that the adaptive scheme controlling complexity on-line outperforms the off-line optimization of fixed settings. The main improvements are seen for the low target complexities, which is our main objective. (At the higher complexity levels, the methods basically converge to searching the full configuration b_{max} and r_{max}) For target complexities, $C_{\mathrm{T}} < 210$ MWSP/s, the three test sequences' average improvement of the R^* compared with *best individual settings* is 0.76 %; compared with *best average settings* is 1.70% and improvements up to 7.48% are obtained for the sequence Mobcal and low complexities; compared with the *complexity worst case settings*, it is 6.3% in average and improvements up to 18%.

Table 2. The measured complexity per second, C_s, achieved by the proposed control scheme compared to the target complexity C_{T}. Complexities are measured in MWSP/s. The range is roughly $50 - 450$ MWSP/s, depending on the sequence.

Sequence	Barcelona	Mobcal	Tab.ten.	MobBar	MobTab	BarTab
mean($\lvert C_{\mathrm{T}} - C_s \rvert$)	0.94	0.76	0.84	0.78	0.36	0.51
mean($\frac{\lvert C_{\mathrm{T}} - C_s \rvert}{C_s}$)	0.42%	0.44%	0.44%	0.32%	0.17%	0.19%

Scene changes occur frequently in video. Three test sequences with a scene change are constructed, each by concatenating two of the sequences above. The new sequences are named MobBar (Mobcal + Barcelona), MobTab (Mobcal + Table) and BarTab (Barcelona + Table). The R*-C results for MobBar and BarTab are shown in Figure 3. When $C_{\mathrm{T}} < 210$ MWSP/s, the three test sequences' average improvement of R^* compared with *best individual settings* is 2.38%, compared with *best average settings*, it is 2.62%, compared with the *complexity worst case settings*, it is 7.19%. The average complexity control accuracy is 0.2% (shown in Table 2).

4. CONCLUSIONS

A new adaptive complexity control algorithm is presented. Besides accurately controlling the computational load of motion estimation during the encoding process aimed at real-time implementations, it also provides slight average improvements in R-D performance compared to the previously found off-line optimized settings.[4] These improvements are considerable in some cases, up to 6.3% compared with the off-line solution. The improvements is up to 18% compared with the *complexity worst case settings*.

APPENDIX A. RESULTS TO BE COMPARED WITH

The following measures are used for comparison. Let r_M and b_M refer to a fixed setting of $r(i)$ and $b(i)$ for all i in the window of M frames.

The ***best individual settings*** are found among all the individually-tested settings for each sequence based on convex optimization of $\mathrm{R}^* - \mathrm{C}$.

The ***best average settings*** (S_{avg}) are found among all tested settings that the three initial test sequences (training datasets) have in common and averaging both the R^* and the C values. They are denoted by R^*_{avg} and C_{avg}. The best average settings are decided by convex optimization. The set based on these **sorted** (by complexity) settings is denoted as S_{avg}.

The ***complexity worst cases settings*** are decided by the following process. First, find the worst-case Complexity by testing the individual settings $(r_M, b_M)^4$ for sequences, giving 15 different combinations of (r_M, b_M) for each, out of these find the largest complexity of different sequences for each setting as the worst-case complexity.

$$C_{\text{worst case}}(r_M, b_M) = max(C_{\text{train dataset 1}}(r_M, b_M), C_{\text{train dataset 2}}(r_M, b_M)...) \tag{24}$$

Second, for a given $\mathrm{C_T}$, calculate all possible settings with smaller complexity,

$$S_1 = \{(r_M, b_M) | C_{\text{worst case}}(r_M, b_M) \leq C_T, \forall (r_M, b_M)\} \tag{25}$$

Finally, the R^* of the complexity worst cases settings are calculated by

$$R^*\{argmin[R^*_{avg}(r_M, b_M)]\}, (r_M, b_M) \in S_1 \bigcap S_{avg} \tag{26}$$

REFERENCES

1. T. Wiegand and G. J. Sullivan (ed.), “Draft ITU-T Rec. and FDIS (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC),” *H.264/MPEG-4 AVC (JVT-doc. JVT-G050)*, Mar. 2003.
2. T. Wiegand et al.,“Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 13, no. 7, pp. 1-19, July 2003.
3. A. M. Tourapis,“Enhanced predictive zonal search for single and multiple frame motion estimation,” in *Proc. SPIE VCIP 2002*, San Jose, CA, USA, 2002, pp. 1069-1079.
4. J. S. Andersen, S. Forchhammer and S. M. Aghito, “Rate-Distortion-Complexity optimization of Fast Motion Estimation in H.264/MPEG-4 AVC,” in *Proc. ICIP 2004*, Singapore, Oct. 2004, pp. 111-114.
5. D. Alfonso et al.,“Detailed rate-distortion analysis of H.264 video coding standard and comparison to MPEG-2/4,” in *Proc. SPIE VCIP 2003*, Lugano, Switzerland, 2003, pp. 891-902.
6. C. Kannangara, and I. Richardson, “Computional Control of An H.264 Encoder Through Lagrangian Cost Function Estimation”, in *Proc. VLBV'05*, Sardinia, Italy, Sept. 2005, p. 4.
7. I. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd, 2003.
8. K. Sayood. *Introduction to Data Compression.* Morgan Kaufmann, second edition, 2000.

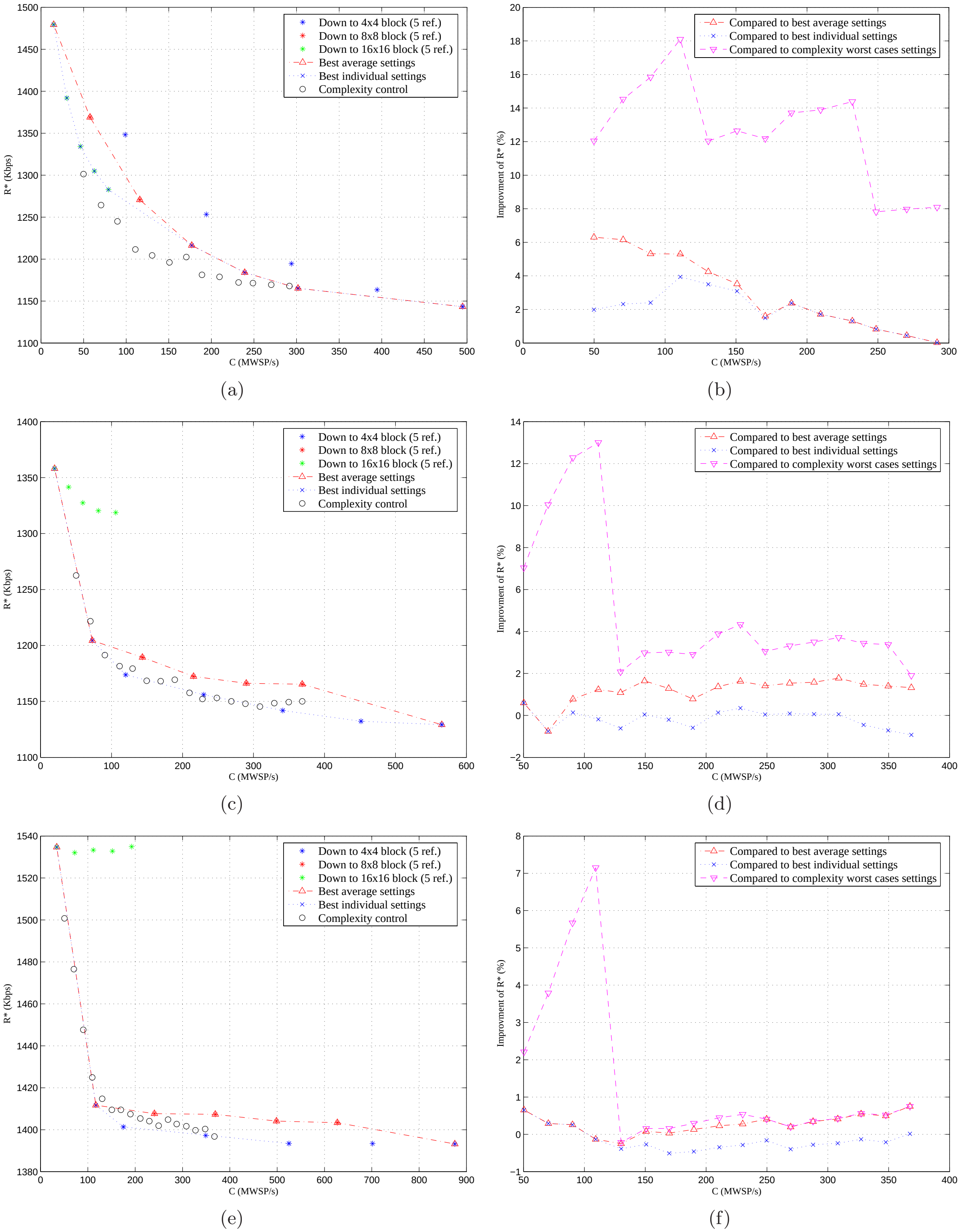
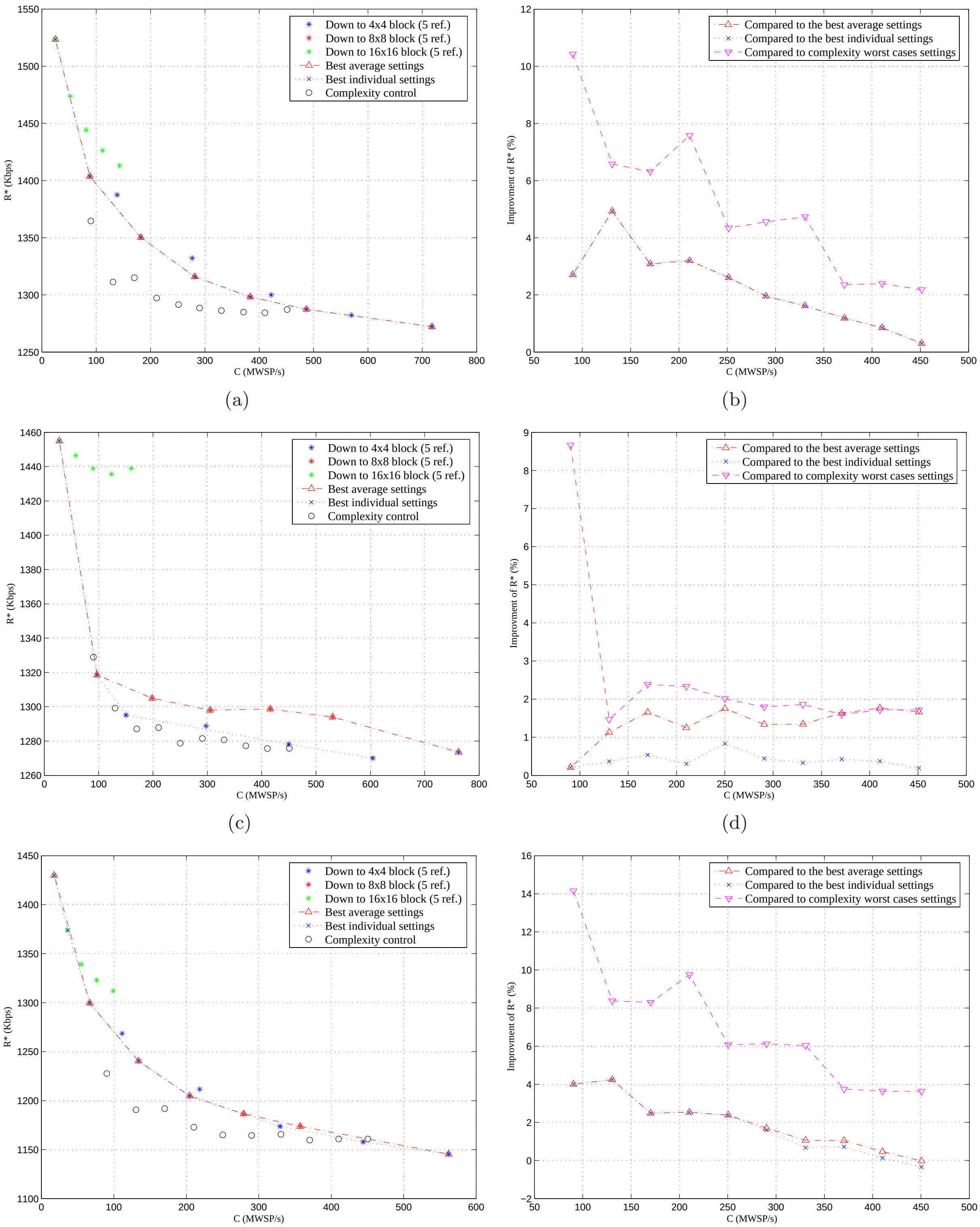


Figure 2. Evaluation of the R-D-C (R*-C) performance: The proposed control scheme compared to the *best average settings*, the *best individual settings* and the *complexity worst case settings*, for Mobcal (a) , Table tennis (c) and Barcelona (e). The percentage of improvement in terms of R^*, for Mobcal (b), Table tennis (d) and Barcelona (f).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3. Evaluation of R-D-C (R*-C) performance for sequences with scene change. The proposed scheme compared to the *best average settings*, the *best individual settings* and the *complexity worst case settings*, for MobBar (a), BarTab (c) and MobTab (e). The percentage of improvement in terms of R^*, for MobBar (b), BarTab (d) and MobTab (f).