A Novel Model-Based Rate-Control Method for Portrait Video Coding

Keman Yu, *Member, IEEE*, Jiang Li, *Senior Member, IEEE*, Cuizhu Shi, and Shipeng Li, *Member, IEEE*

***Abstract*—The rapid development of wireless networks and mobile devices has made mobile video communication a particularly promising service. We previously proposed an effective video form, scalable portrait video. In low-bandwidth conditions, portrait video possesses clearer shape, smoother motion, and much cheaper computational cost than discrete cosine transform (DCT)-based schemes. However, the bit rate of portrait video cannot be accurately modeled by a rate-distortion function as in DCT-based schemes. How to effectively control the bit rate is a hard challenge for portrait video. In this paper, we propose a novel model-based rate-control method. Although the coding parameters cannot be directly calculated from the target bit rate, we build a model between the bit-rate reduction and the percentage of less probable symbols (LPS) based on the principle of entropy coding, which is referred to as the LPS-rate model. We use this model to obtain the desired coding parameters. Experimental results show that the proposed method not only effectively controls the bit rate, but also significantly reduces the number of skipped frames. The principle of this method can also be applied to general bit plane coding in other image processing and video compression technologies.**

***Index Terms*—Bi-level video, entropy coding, portrait video, rate control.**

I. Introduction

RECENT years have witnessed the rapid development of wireless networks and mobile devices. General Packet Radio Service (GPRS) and Code Division Multiple Access (CDMA 1X) as 2.5G solutions to wide areas are available in increasingly more regions. Wireless LAN 802.11 and Bluetooth have also grown quickly for local-area wireless networks. At the same time, mobile devices have increasingly gained in processing power, storage capacity, and battery lifetime. These evolutions have made mobile video communication a particularly promising service.

After reviewing existing discrete cosine transform (DCT)-based video technologies, such as MPEG1/2/4 [1]–[3] and H.261/263 [4], [5], we find that they perform well in the bandwidth range greater than about 40 kb/s for quarter common intermediate format (QCIF) size. However, the bandwidth of 20–40 kb/s is the current range that is stably provided in 2.5G wireless networks. Moreover, the conventional MPEG/H.26x is still computationally expensive and practically infeasible for real-time coding on mobile devices.

In very low-bandwidth conditions, the video generated by MPEG/H.26x usually looks like a collection of color blocks and the motion becomes discontinuous. However, in video communications, facial expressions that are represented by the motions of the outlines of the face, eyes, eyebrows, and mouth deliver more information than the basic colors of the face. We previously proposed bi-level video [6] to represent these facial expressions and achieve very high compression ratios. In low-bandwidth conditions, such as below 20 kb/s, bi-level video possesses clearer shape, smoother motion, shorter initial latency, and much cheaper computational cost than DCT-based schemes. If more bandwidth is available, we take advantage of it by using portrait video [7], which is an extension of bi-level video. It is composed of more gray levels and, therefore, possesses higher visual quality. Portrait video is so named because the coding is ordered from outlines to details, and videos at lower detail levels appear similar to portraits. The bit rate of 3–4-level portrait videos fits well into the bandwidth range of 20–40 kb/s.

Rate control is a significant component of a video coding scheme. In many applications, especially in real-time video communication, video data must be transmitted over limited bit-rate channels. The rate-control scheme is then responsible for adjusting the coding parameters to adapt the bit rate to the given target. In general DCT-based video coding schemes, coding parameters can be directly calculated by a rate-distortion function. However, in the portrait video coding scheme, the bit rate cannot be modeled by a rate-distortion function as in DCT-based schemes. We previously proposed a semi-empirical rate-control method. In this method, some semi-empirical modes are obtained offline by encoding numerous sequences with vastly varying content. Each mode corresponds to a pair of coding parameters, i.e., a dissimilarity threshold and a half-width of threshold band. During encoding, an appropriate mode is selected according to the available channel bandwidth and output buffer fullness. Because the picture complexity of different sequences and different frames in the same sequence may be vastly diverse, the predefined modes cannot effectively control the bit rate, and, as a result, many frames are skipped.

The problem is that there is not a rate-distortion function that can model the bit rate for portrait video. Thus, the characteristic of this video form leads to a hard challenge: *How can the bit rate be estimated and controlled as accurately as possible*? The obvious difficulty is that the picture complexity is hard to know and the coding parameters are not easily derived.

Manuscript received July 22, 2004; revised November 8, 2004. This work was carried out at Microsoft Research Asia. This paper was recommended by Associate Editor H. Sun.

K. Yu, J. Li, and S. Li are with Microsoft Research Asia, Beijing 100080, China (e-mail: kmyu@microsoft.com; jiangli@microsoft.com; spli@microsoft.com).

C. Shi is with the Institute of Image Communication and Information Processing, Shanghai Jiaotong University, Shanghai 200030, China (e-mail: cuizhushi@hotmail.com).

Digital Object Identifier 10.1109/TCSVT.2005.857310

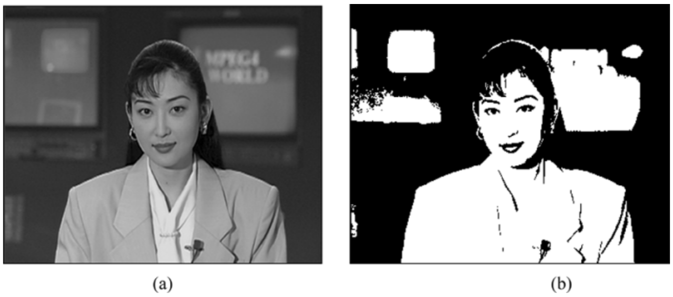


Fig. 1. (a) Grayscale image and (b) its bi-level image.

In this paper, we present a method that not only effectively controls the bit rate but also requires little additional computational overhead. First, we use entropy to estimate the picture complexity; second, we calculate the less probable symbol (LPS) percentage using an LPS-rate function; third, we determine coding parameters according to the LPS percentage. Experimental results show that the proposed method not only effectively controls the bit rate, but also significantly reduces the number of skipped frames. The principle of this method can also be applied to general bit plane coding in other image processing and video compression technologies.

The rest of the paper is organized as follows. Section II reviews the generation, coding, and rate-control of bi-level video, which is the base of portrait video. We describe the principle and details of the proposed rate-control method in Section III. In Section IV, the generation and coding of multiple-level video is first reviewed, and then the rate-control method for multiple-level video is presented. Experimental results are shown and discussed in Section V. We summarize this paper in Section VI.

II. Bi-Level Video Coding and Rate Control

Bi-level video coding is the base of portrait video coding. In designing the rate-control technique, we begin with an examination of bi-level video coding. Before introducing our approach, we first review the architecture of the generation and coding of bi-level video in this section, and then briefly summarize the semi-empirical rate-control method.

A. Architecture of Bi-Level Video Coding

In video communications, the facial expression represented by the motions of the outlines of the face, eyes, eyebrows, and mouth delivers more information than the basic colors of the face. Since the representation of outlines needs only two colors, such as black and white, we developed a video form in which each pixel is represented by only 1 bit. We call it bi-level video. Fig. 1 shows an example of generating a bi-level image from a grayscale image using a simple thresholding method. From this figure, we can clearly identify the person. Experiments [6] show that it is also very easy to perceive facial expressions of a person in a bi-level image sequence.

Conversion of a color video to a bi-level image sequence achieves a significant step in bit-rate reduction. To compress the bi-level image sequence, we designed a bi-level video coding architecture.

As shown in Fig. 2, $G(n)$ is an input grayscale image frame. If this frame is an intra-frame (I-frame), it is directly sent to the adaptive context-based arithmetic encoding (CAE) module; otherwise, it is compared with its previous frame, i.e., $G'(n-1)$, in the Static Region Detection and Duplication module. For each pixel in the same location, we calculate the sum of absolute difference (SAD) in a 3×3 block surrounding the pixel. If the SAD is under a dissimilarity threshold T_d, the pixel is duplicated from its corresponding pixel in the previous frame. In the CAE module, each pixel is coded in a raster order. The process of encoding a given pixel is given here.

1) Computing a context number. If the frame is an I-frame, a 10-bit context $C = \sum_k c_k \cdot 2^k$ is computed based on the coded pixels in the current frame, as illustrated in Fig. 3(a); otherwise, a 9-bit context number is built from the current frame with respect to the previous bi-level image $B(n-1)$, as illustrated in Fig. 3(b).
2) Determining the bi-level value for the pixel in terms of a threshold T and ΔT. For a pixel with its grayscale value within the threshold band $(T-\Delta T, T+\Delta T]$, its bi-level value is determined according to the probability table indexed by the context number. If the probability of 0 is larger than that of 1, the bi-level value is 0, vice versa. For pixels with grayscale values out of the threshold band, their bi-level values are simply determined in terms of T.
3) Using the indexed probability value to drive an arithmetic coder. After the entire frame is processed, the compressed bits and the bi-level image are then sent to the Rate Control module. This module adjusts the dissimilarity threshold T_d and the half-width of threshold band ΔT, and outputs a flag t to indicate if it is transmitted.

B. Semi-Empirical Rate Control

In a DCT-based coding scheme, bit-rate control is achieved by adjusting the quantization parameter. In the bi-level video coding scheme, the bit rate can be adjusted by the following two approaches.

1) *Threshold Band*: The wider the threshold band is, the more pixels are coded according to the probability table of the adaptive context-based arithmetic coder. Therefore, few bits are generated.
2) *Dissimilarity Threshold*: The higher the dissimilarity threshold is, the more pixels are viewed as being similar to their corresponding pixels in the previous frame, and therefore a higher compression ratio is achieved.

Thus, the goal of rate control in the bi-level video scheme is to adapt the bit rate to the target by adjusting the width of the threshold band and the value of the dissimilarity threshold.

The difficulty is that unlike DCT-based coding, bi-level video coding does not possess a rate-distortion function that can be modeled by a simple analytic function. Therefore, a semi-empirical method is adopted. In this method, ten semi-empirical modes are obtained offline by encoding numerous sequences with vastly varying content. Each mode corresponds to a combination of the half-width of the threshold band (ΔT) and the dissimilarity threshold (T_d). Then, during video coding, after a frame is encoded, an appropriate mode for the next frame is

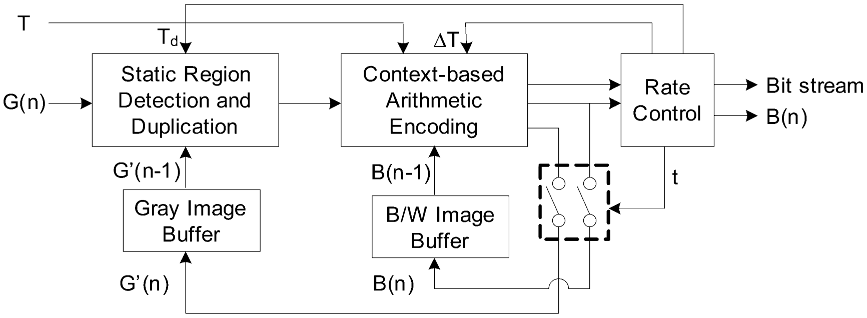


Fig. 2. Bi-level video encoder structure.

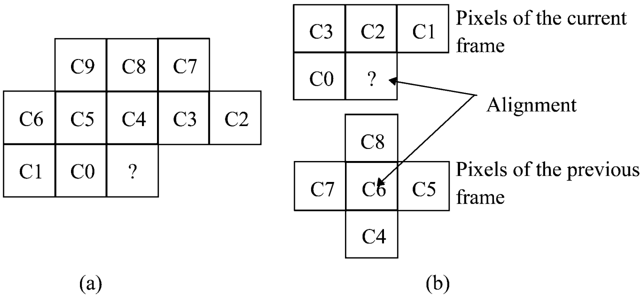


Fig. 3. (a) Intra template and context construction. (b) Iinter template and context construction. The pixel to be coded is marked with "?."

selected according to the target rate and output buffer fullness. However, because the picture complexity of different sequences and different frames in the same sequence may be vastly diverse, applying the same mode to different images is likely to achieve vastly different performances. The semi-empirical method cannot effectively control the bit rate.

The problem is due to the unavailability of a rate-distortion function that can model the bit rate. In Section III, we will introduce how we build another kind of model to accurately estimate and control the bit rate.

III. Model-Based Rate Control for Bi-Level Video

The general rate-control algorithm of DCT-based video coding schemes is derived from classic rate-distortion (R-D) theory, whereas our rate-control model is based on the principle of entropy coding. This section starts by introducing how the compression ratio can be improved using entropy coding, and then elaborates how the rate-control model is built based on the principle of entropy coding.

A. How is the Compression Ratio Improved?

Given a source S that generates random symbols, $s_1, s_2, \ldots, s_N$, and the probability p_i of the occurrence of symbol s_i, according to Shannon's information theory [8], the average information, say entropy, of the source is defined as

$$H(S) = \sum_i p_i \log_2 \frac{1}{p_i}. \tag{1}$$

From information theory, if the symbols are distinct, then the average number of bits needed to encode them is always bounded by their entropy.

Binary arithmetic coding [9] is one of the coding methods that can approach the entropy bound. There are only two kinds of input symbols in a binary arithmetic coder. The symbols that occur with high probability are usually referred to as the more probable symbol (MPS), and the symbols that occur with low probability are referred to as the less probable symbol (LPS). The key idea of binary arithmetic coding is recursively subdividing the probability interval.

Assume that the LPS occurs with a probability of Q_e. The probability of MPS is $(1 - Q_e)$. The subinterval for LPS is ordered above the subinterval for MPS. If the width of the probability interval is represented by A and the code register is defined as C, the coding process is:

for MPS coding

$$C = C \tag{2}$$
$$A = A(1 - Q_e) \tag{3}$$

and for LPS coding

$$C = C + A(1 - Q_e) \tag{4}$$
$$A = AQ_e. \tag{5}$$

We can see that, whenever the MPS is coded, the code register is left unchanged. Whenever the LPS is coded, the value of $A(1 - Q_e)$ is added to the code register. Obviously, the smaller the number of LPS is, the higher the compression ratio is.

Actually, the two approaches that we applied to bi-level video coding to improve the compression ratio exactly comply with the above conclusion. First, for a pixel being originally regarded as LPS according to the single threshold T, if its grayscale value is within the threshold band $(T - \Delta T, T + \Delta T]$, it will be converted to MPS. Thus, enlarging the threshold band reduces the number of LPS's. Second, the higher the value of the dissimilarity threshold is, the more pixels are duplicated from the previous frame, and therefore fewer LPSs occur.

The above study inspires us to find a model between the number of LPSs and the compression ratio or bit-rate reduction.

B. Bit-Rate Estimation

As mentioned in the introduction, one difficulty in achieving our rate-control goal is that the picture complexity is not known before encoding. In DCT-based schemes, the picture complexity is generally represented by the mean absolute difference (MAD) of the current frame after motion compensation. However, in the bi-level video scheme, there is no motion estimation and compensation module. A possible approach is to use the amount of information as the picture complexity. As entropy is the average information of a source, the picture complexity is computable before encoding. Moreover, because entropy is the lower bound of lossless compression and it is measured in bits per symbol, the picture complexity here is also an estimate of the encoding bit count.

In adaptive context-based arithmetic encoding, the entropy cannot be simply obtained by using (1) with the occurrence probabilities of symbol 0 and symbol 1. To compute the complexity of a picture, all pixels are classified into N groups according to their corresponding context values. N is equal to the number of the contexts. For a template containing 9 bits, $N = 2^9$. Each group here actually acts as a source. The entropy of each group is calculated respectively, and then the complexity of the picture can be obtained as follows:

$$H(i) = p_{i0} \log p_{i0}^{-1} + p_{i1} \log p_{i1}^{-1} \tag{6}$$

$$E = \sum_{i=1}^{N} H(i) N_i \tag{7}$$

where p_{i0} and p_{i1} represent the probability of symbol 0 and 1, respectively, in the ith group, N_i is the number of symbols belonging to the ith group, and E is the picture complexity, namely the estimate of the encoding bit count. Note that to achieve an accurate estimate, the source in complexity computation and encoding should be the same and therefore picture complexity should be carried out after Static Region Detection and Duplication.

As the number of bits needed to encode a source is bounded by its entropy, if we use the same context template in computing E and encoding, E should always be smaller than the actual encoding bit count. On the other hand, computing the context values is computationally expensive. If the same template is adopted, the cost of computing E is almost comparable to that of encoding a bi-level image. Therefore, a context template that has fewer context bits is desired for computing E.

Similar to other rate-control methods, only inter-frame coding is considered in portrait video rate control. From the inter template illustrated in Fig. 3(b), we remove the context bits C7 and C8, which are likely the same as C0 and C2, and remove C1 to further reduce the number of context bits and then compose a 6-bit template as shown in Fig. 4. We use this template in computing E. In our experiments on many sequences, this template can relatively accurately estimate the actual encoding bit count, as shown in Fig. 5. In addition, this template significantly reduces the computational cost since here $N = 64$.

As we mentioned in the preceding subsection, the number of LPSs has a significant meaning to bit-rate reduction. The number of LPSs can be easily obtained while computing the picture complexity. For example, given a pixel that is classified to the ith group, if its grayscale value is not larger than the threshold T but $p_{i0} < p_{i1}$ or its grayscale value is larger than T but $p_{i0} > p_{i1}$, this pixel is regarded as an LPS. At the same time, the number of LPS pixels within a threshold band of different widths is also countable. We define LPS_{all} to be the number of all LPS pixels in the whole picture, LPS_i to be the number of LPS pixels which have grayscale values within the threshold band $(T - i, T + i]$. If we set the half-width of the threshold band to be i during encoding, all LPSs within the band will be converted to MPS, then the LPS reduction ratio is

$$\text{LPS}_{ri} = \frac{\text{LPS}_i}{\text{LPS}_{\text{all}}}. \tag{8}$$

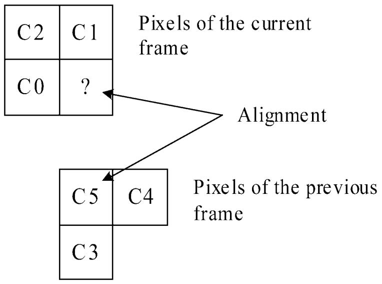
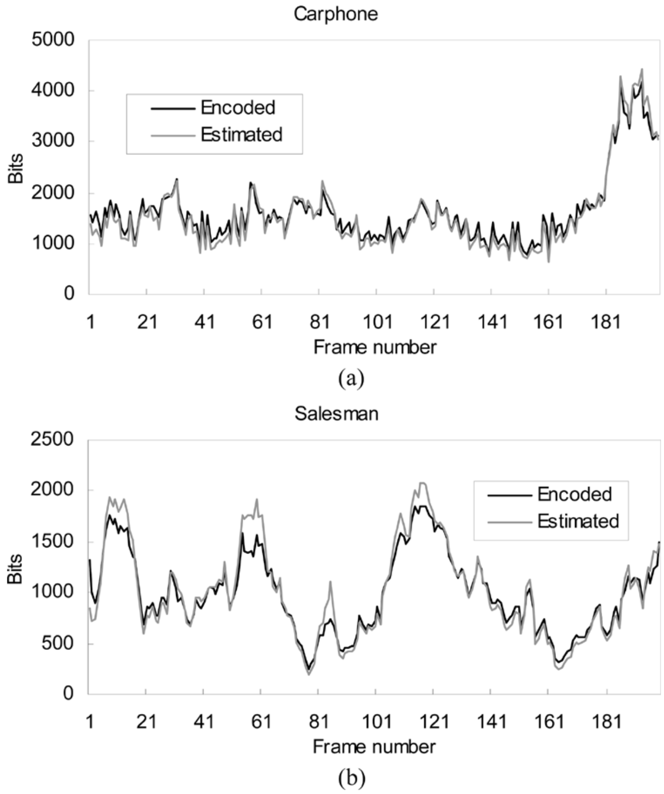


Fig. 4. Inter template for computing picture complexity. The current pixel is marked with “?.”



(a)

(b)

Fig. 5. Encoding bit-rate estimation.

C. How Wide is the Control Range?

It is known that adjusting the width of the threshold band and the value of dissimilarity threshold can control the bit rate. In the previous rate-control method, the range of T_d and ΔT is $[1.0, 3.0]$ and $[0, 5]$, respectively. In practice, we found that, in large motion scenes, such as the *Carphone* sequence, small regions are regarded as static and duplicated from the previous frame. In small motion scenes, such as the *Akiyo* sequence, although increasing T_d can obviously reduce the bit rate, it introduces a significant loss in visual quality at the same time. On the other hand, the relationship between T_d and LPS reduction cannot be formulated as that of ΔT. Therefore, we fixed T_d

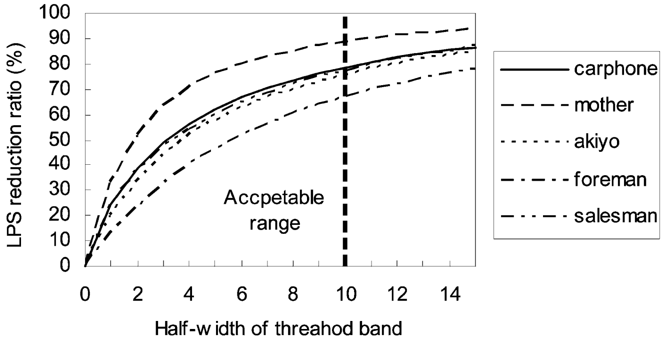


Fig. 6. Trend of LPS reduction ratio.

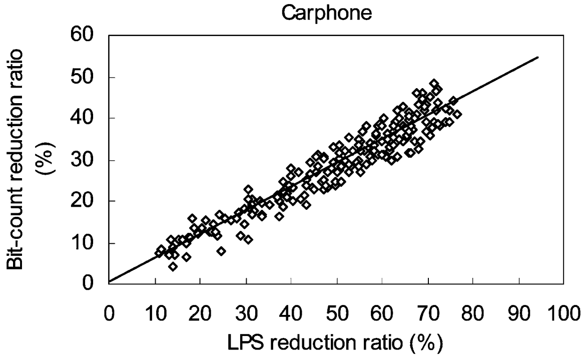
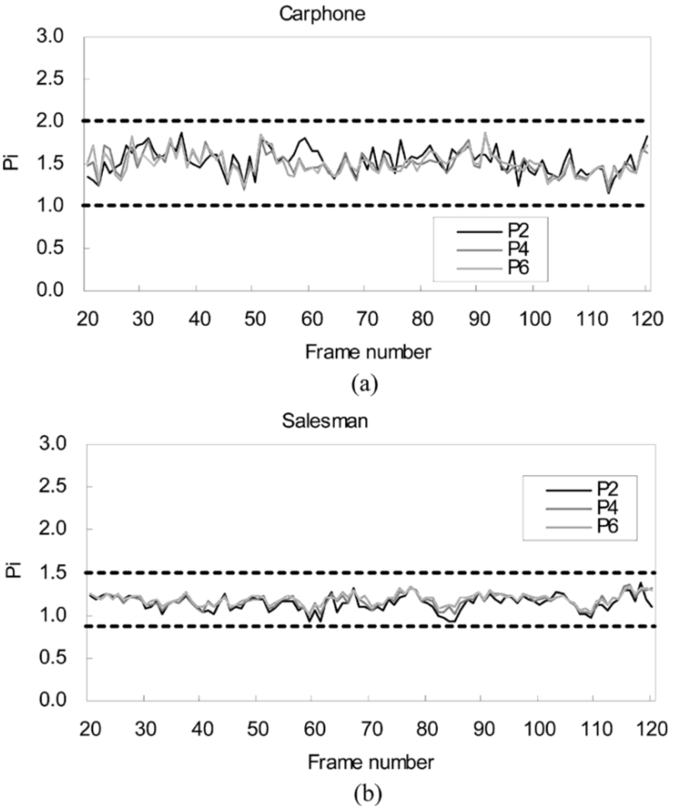


Fig. 7. Scatter plot for bit-count reduction ratio and LPS reduction ratio.



(a)

(b)

Fig. 8. LPS-rate model's stability and uniformity verification.

to a value that does not introduce visible loss in visual quality. $T_d = 0.8$ seems to work well on various sequences.

We also found that enlarging the threshold band can effectively reduce the number of LPSs with virtually no obvious loss in visual quality. However, as shown in Fig. 6, the rise of the LPS reduction ratio slows down along with the enlarging of the threshold band. When $\Delta T = 10$, the degradation of visual quality is acceptable while the number of LPSs is also significantly reduced. Therefore, we enlarge the range of the threshold band to $(T - 10, T + 10]$.

D. Bit-Rate Reduction Model

From the foregoing descriptions, we know that reducing the number of LPSs decreases the encoding bit count. Now it can be verified using the LPS reduction ratio calculated with (8), as illustrated in Fig. 7. The 200 points in the figure are obtained by encoding the first 20 frames of the *Carphone* sequence with ΔT varying from 1 to 10.

Fig. 7 not only illustrates that the bit-count reduction ratio increases along with the LPS reduction ratio, but also indicates that they basically obey a linear relationship. If the linear relationship is relatively stable, uniform, and accurate with different picture contents and different widths of the threshold band, it is exactly the model that we are seeking. Thus, we define the following model and examine it on many sequences with vastly varying content:

$$\frac{\mathrm{LPS}_{ri}}{\mathrm{Rate}_{ri}} = P_i, \qquad \text{for } i = 1, \ldots, 10 \tag{9}$$

where LPS_{ri} represents the LPS reduction ratio and Rate_{ri} represents the bit-count reduction ratio when the half-width threshold band $\Delta T = i$, P_i is the model parameter.

The examination results of the *Carphone* and *Salesman* sequences are shown the Fig. 8. In this figure, the three curves represent the models with ΔT equal to 2, 4, and 6, respectively. In the *Carphone* figure, all three curves are between 1.0 and 2, while they slightly fluctuate around the line of 1.2 in the *Salesman* figure. It is verified that the model is relatively stable. This characteristic also matches the results of all other test sequences. Moreover, the three curves substantially coincide in both figures. It is also verified that the model is relatively uniform with different ΔT and that we can use a uniform P to replace P_i.

To examine whether the model is accurate, one approach is to use it to estimate the encoding bit count with different ΔT. It is known that the picture complexity E is an accurate estimate of the encoding bit count when ΔT is zero, and the bit-count reduction ratio can be obtained by using (10). Then, the estimated bit count L_e for $\Delta T = i$ can be calculated as follows:

$$\mathrm{Rate}_{ri} = \frac{\mathrm{LPS}_{ri}}{P} \tag{10}$$

$$L_e = E \times (1 - \mathrm{Rate}_{ri}) \tag{11}$$

where the model parameter P is initialized to 1.5 as it is the average value in our experiments on many sequences.

After encoding a frame, the model is updated based on the encoding results of the current frame as well as the results of the past frames. First, given the encoding bit count of the current

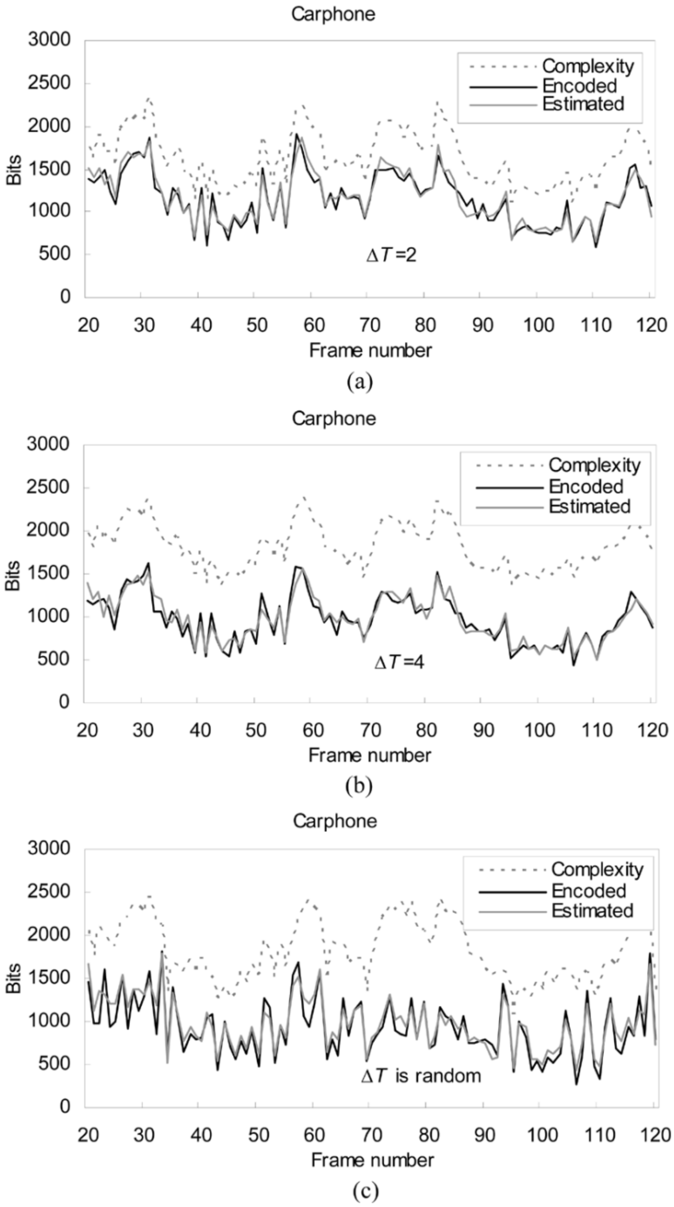


Fig. 9. Bit-count estimation with different widths of threshold band.

frame L_a, the actual bit-count reduction ratio, say Rate_r', is calculated; second, the new model parameter P' is updated; third, the new parameter is restricted in a specified range and then smoothed by a low-pass filter to avoid oscillation. The model update process is defined by the following equations:

$$\text{Rate}_r' = \frac{E - L_a}{E} \tag{12}$$

$$P' = \frac{\text{LPS}_{ri}}{\text{Rate}_r'} \tag{13}$$

$$P' = \max(1, \min(5, P')) \tag{14}$$

$$P = \alpha P + (1 - \alpha)P'. \tag{15}$$

Note that, for the infrequent case that the calculated complexity is smaller than the encoding bit count, i.e., $\text{Rate}_r' < 0$, the model update process is skipped. Increasing α increases the influence of the previous values whereas decreasing α results in a higher influence of the new value. Because $\alpha = 0.7$ seems to work well, we performed all of the following experiments with this value.

Fig. 9 illustrates how the encoding bit count is accurately estimated using the model. The half-width of the threshold band is set to be 2, 4, and a random value in the range of $[1, 10]$, respectively. This figure also clearly shows that increasing the width of the threshold band effectively reduces the bit rate.

It has been verified that the LPS-rate model is relatively stable, uniform, and accurate under different conditions. It is certainly the model that we are seeking. In Section III-E, we will introduce how to apply it to rate control.

E. Model-Based Rate Control

The goal of rate control in the bi-level video scheme is just to select an appropriate half-width of the threshold band and dissimilarity threshold that can adapt the bit rate to the target. Based on the observations and verifications in the previous subsection, we developed a model-based rate-control method. Fig. 10 shows the bi-level video encoder structure with the new rate-control module. In the new structure, the rate-control module is composed of two submodules that are performed before and after the encoding respectively. The rate-control process can be summarized in Fig. 11. There are six major steps in this scheme, as discussed in detail in the rest of this subsection. The target bit calculation, buffer control, and skip-frame control methods in our scheme are based on that of MPEG-4 Q2 rate-control scheme [3].

1) *Initialization*: At this stage, the model parameter P is initialized to 1.5. The buffer size B_s is initialized to half of the target bit rate, say R, and the buffer level B is initialized to half of the buffer size.
2) *Picture complexity and LPS reduction ratio calculation*: The picture complexity E can be obtained using (6) and (7). The LPS reduction ratio with ΔT varying from 1 to 10 can be calculated using (8).
3) *Target bit calculation*: The target bits for each new P-frame are determined as follows:

$$L_t = \frac{R}{f} \times \frac{2B_s - B}{B_s + B} \tag{16}$$

where f is the target frame rate.

4) *Threshold-band width selection*: First, calculate the desired bit-count reduction ratio Rate_r. Second, obtain the desired LPS reduction ratio LPS_r. Then, select the minimal ΔT that has the LPS reduction ratio not smaller than LPS_r.

$$\text{Rate}_r = \frac{E - L_t}{E} \tag{17}$$

$$\text{LPS}_r = \text{Rate}_r \times P \tag{18}$$

$$\Delta T = \min\{i | i \in \{1, \ldots, 10\}, \text{LPS}_{ri} \geq \text{LPS}_r\}. \tag{19}$$

Note that, if there is not a ΔT that satisfies the requirement, then the maximal value, i.e., 10, is adopted.

5) *Updating LPS-Rate model*: The parameter P of the LPS-rate function is continually updated using (12)–(15).
6) *Skip-frame control*: After encoding a frame, the buffer level B is updated by adding the total number of the bits generated from the current frame L_a and subtracting the

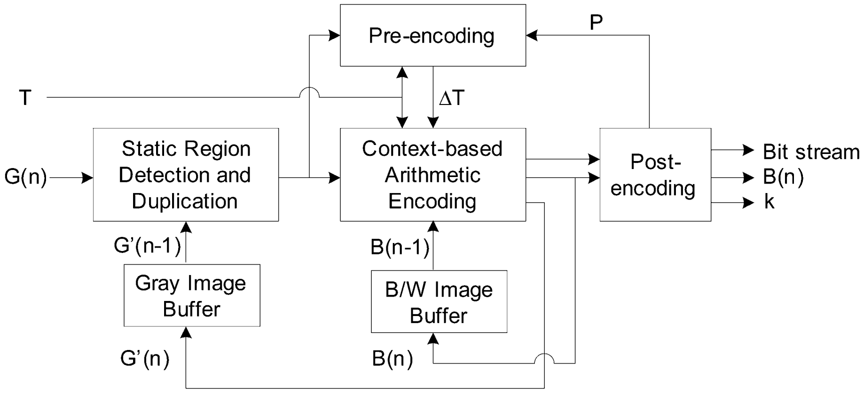


Fig. 10. New bi-level video encoder structure.

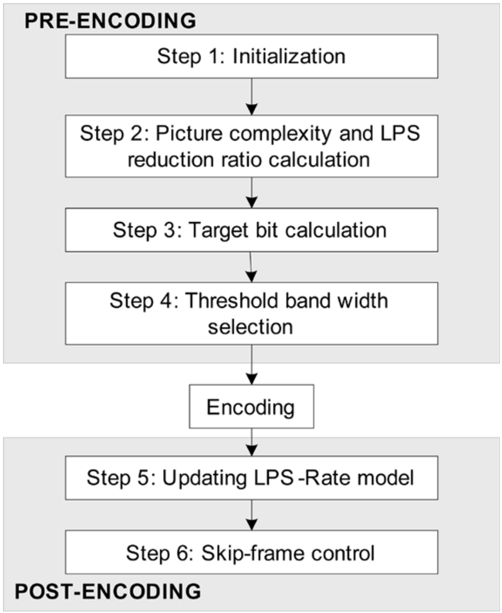

PRE-ENCODING
Step 1: Initialization
Step 2: Picture complexity and LPS reduction ratio calculation
Step 3: Target bit calculation
Step 4: Threshold band width selection
Encoding
Step 5: Updating LPS-Rate model
Step 6: Skip-frame control
POST-ENCODING

Fig. 11. Model-based bi-level video rate-control scheme.

number of bits removed from the buffer. If B exceeds 80% of the buffer size, the encoder skips the upcoming k frames, until the buffer occupancy is reduced to a safe level

$$B = B + L_a - \frac{R}{f}. \tag{20}$$

We have described the whole process of the model-based rate-control method from observation and verification to realization. The method is designed for bi-level portrait video only. Section IV will introduce how to control the rate of multiple-level video based on this method.

IV. Multiple-Level-Video Rate Control

Before introducing the rate-control method for multiple-level video, we first briefly review the generation and coding of multiple-level video and then present how the relationship between bi-level video and multiple-level video can be utilized in rate control.

As the bit rate of 3–4-level portrait videos fits well into the bandwidth range of 20–40 kb/s, and if the number of levels of a multi-level video is greater than four, its compression ratio is no longer competitive with that of DCT-based methods [7], the multilevel video we consider in this paper includes three- and four-level videos only.

A. Multilevel Video Generation and Coding

Fig. 12 illustrate the generation of a four-level image. In a four-level image, each pixel is represented by two bits, so a four-level image can be regards as two bit planes. If we use three thresholds $T_{22} < T_1 < T_{21}$ to convert a grayscale image [Fig. 12(a)] into three bi-level images, respectively, the bi-level image [Fig. 12(b)] generated by the middle threshold T_1 is exactly the first bit plane of the four-level image, while the second bit plane [Fig. 12(e)] is composed of two bi-level images [Fig. 12(c) and (d)] generated using the higher threshold and the lower threshold respectively. The two bit planes comprise a four-level image [Fig. 12(f)].

During coding of a four-level image, the two bit planes are encoded respectively using the aforementioned bi-level image coding method. However, because the second bit plane [Fig. 12(e)] appears to be complex, where some regions even appear like random noise, it is inefficient to compress such a noisy image directly. As the two components [Fig. 12(c) and (d)] of the second bit plane are relatively simple, coding of the second bit plane can be replaced by the combination of coding two mutually exclusive bi-level image components. In addition, we do not need extra bits to describe the regions of the image components, since they are indicated by the first bit plane. Encoding the second bit plane using such a method is more efficient than coding it directly, as was concluded in [7].

Generating and coding three-level video is much simpler than that of four-level video, because only two bi-level images are generated, and the image [Fig. 12(d)] generated by the lower threshold is exactly the second bit plane of the three-level image.

We adopt an empirical method to select thresholds. The threshold of the first bit plane, called the principal threshold, can be adjusted by users. The higher threshold and the lower threshold for the second bit plane are always set as the principal threshold plus and minus 16, respectively.

B. Multilevel-Video Rate Control

Since multilevel-video coding is exactly the coding of multiple bit planes, the previously introduced model-based rate-

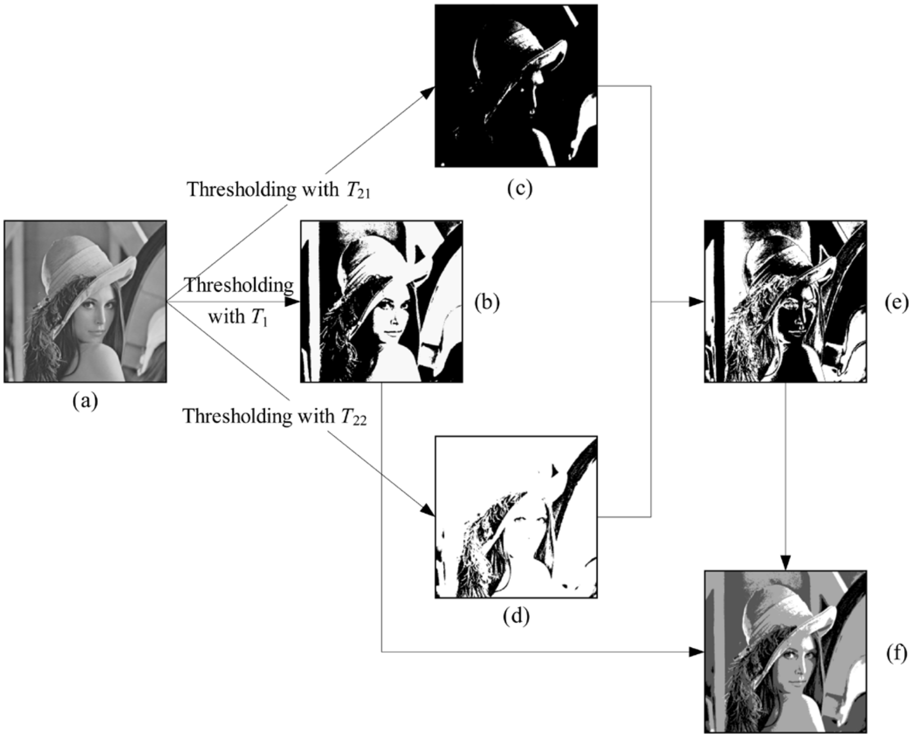


Fig. 12. Generation of a four-level image.

control method can be directly applied to the coding of each bit plane of a multilevel video. However, the computational cost is too expensive if the picture complexity of each bit plane is computed. This problem can be solved if the picture complexity of the multilevel image can be estimated from that of a bi-level image. This is possible because the multiple bi-level images generated by the thresholds with a distance of 16 are likely to have the same complexity. After a large number of experiments, we found that the bit rates of a three-level video and a four-level video are about 1.7±0.3 and 2.7±0.3 times, respectively, of that of a bi-level video. The above assumption and observation motivates us to define a function to model the relationship between the complexity of bi-level video E_2 and the complexity of multilevel video E_i as follows:

$$\frac{E_i}{E_2} = Q, \qquad \text{for } i = 3, 4. \tag{21}$$

In order to examine whether this model is accurate before applying it to rate control, we added this model to the bit-rate estimation process described in Section III-D to predict the coding bit count of multilevel videos. The picture complexity calculation is performed only on the first bit plane. The L_a in (12) is replaced by the bit count of the first bit plane L'_a. The model parameter Q is updated as

$$Q = \frac{L_a}{L'_a} \tag{22}$$

$$Q = \beta Q + (1-\beta)Q' \tag{23}$$

where L_a represents the bit count of the whole image and Q is initialized to 1.7 if the number of levels is three or 2.7 if the number of levels is four; $\beta = 0.7$ works well.

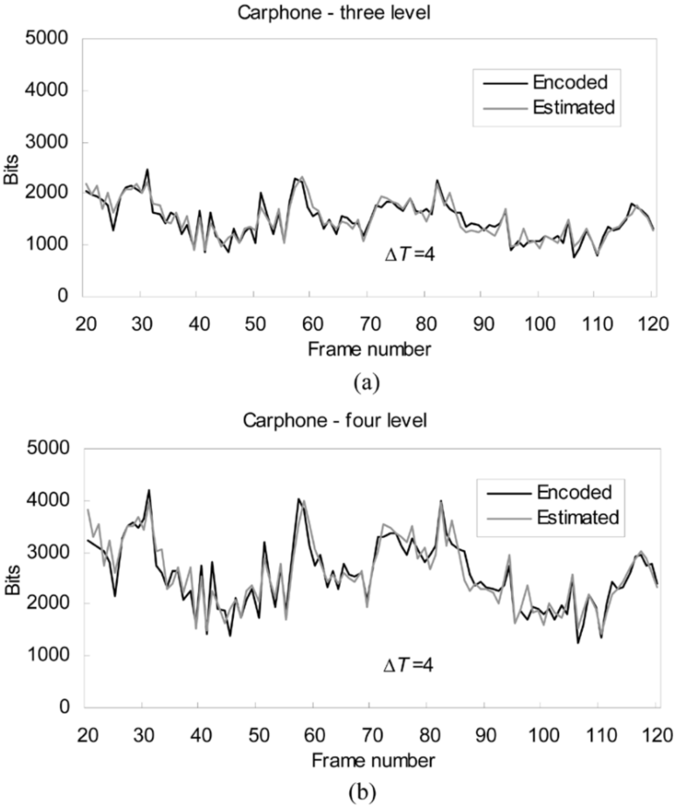


Fig. 13. Bit-count estimation of multilevel video.

As illustrated in Fig. 13, the encoding bit count of multilevel video is accurately estimated by the picture complexity of the first bit plane. The method also works well on a large number of sequences in our experiments. It implies that, given the bit count of a multilevel image, the bits of the first bit plane can be estimated using the model. This is exactly the essential condition of extending the model-based rate-control method to multilevel video.

We integrate the function into the previously described model-based rate-control method to obtain the desired rate-control scheme for portrait video. First, in the *Initialization* stage, Q is initialized to 1, 1.7, or 2.7 for bi-level, three-level, and four-level videos, respectively. Second, in the *Threshold-band width-selection* stage, the number of target bits used to calculate the desired bit-count reduction ratio in (17) is replaced by L_t' to yield

$$L_t = \frac{L_t}{Q}. \tag{24}$$

The selected threshold-band half-width is applied to the coding of all bit planes. Third, in the *Updating LPS-Rate model* step, the actual encoding bit count of the first bit plane L_a' is used in (12) to update the model parameter P, and (22) and (23) are added to update the parameter Q.

V. Experimental Results

To evaluate the performance of the proposed rate-control algorithm under different conditions, we chose six sequences with vastly varying content from standard MPEG-4 test video clips. They are the *Akiyo*, *Mother*, *Salesman*, *Carphone*, *Foreman*, and *Coastguard* sequences. The first three sequences represent scenes with relatively small motion and a fixed background, where movements in *Akiyo* are restricted to the face area of the speaker, *Mother* has two speakers with head and shoulder movements, and *Salesman* possesses a complex background. The other three sequences represent scenes with relatively large motion, where *Carphone* possesses large facial motion and a fast moving background, *Foreman* has large motion in all directions and camera panning, and there is large but uniform motion in *Coastguard.* All sequences are in QCIF format and encoded at 15 frames per second. All frames except for the first frame are encoded as P-frames. Note that the sequences are encoded with different target bit rates, as shown in Table I. Although the sequences with large motion, such as *Foreman*, can also be encoded with a very low bit rate, it is achieved at the expense of the output frame rate. On the other hand, assigning more bits to small motion scenes cannot improve visual quality. Therefore, the target bit rates are assigned according to the motion activities.

TABLE I
Target Bit Rate for Different Sequences and Different Numbers of Levels (kb/s)

	Bi-level	Three-level	Four-level
Akiyo	4.8	9.6	14.4
Mother	9.6	14.4	24.0
Salesman	9.6	14.4	24.0
Carphone	14.4	24.0	33.6
Foreman	19.2	28.8	45.0
Coastguard	19.2	28.8	45.0

We compare the performance of the proposed model-based rate-control algorithm and the previous semi-empirical method. Fig. 14 shows the encoding bit count curves of *Mother* and *Carphone*. The two sequences are encoded into bi-level, three-level, and four-level video, respectively. It can be seen that, in videos with different numbers of levels, with the proposed algorithm, the number of bits produced by each frame matches the target value (represented by the dashed lines) much better than that with the semi-empirical method. In most cases, the encoding bit count generated by the proposed algorithm slightly fluctuates near the target value. In some cases, for example, between frames 150–160 in the bi-level *Carphone* video and frames 160–170 in the bi-level *Mother* video, as indicated by $\langle 1 \rangle$ in Fig. 14(a) and (b), the bit-count curve of the current work is obviously lower than the target line. However, in some other cases, for example, between frames 180–200 in the bi-level *Carphone* video and frames 50–60 in the bi-level *Mother* video, as indicated by $\langle 2 \rangle$ in Fig. 14(a) and (b), the bit-count curve of the current work is notably higher than the target line. These two cases can be explained by the threshold band curves of the bi-level *Carphone* video and the bi-level *Mother* video shown in Fig. 14(g) and (h). First, for the frames for which the encoding bit count is obviously less than the target, their corresponding ΔT has already been set to the minimal value (marked by $\langle 1 \rangle$). This indicates that the picture complexity of these frames is very low and the target bit count is overfull for these frames. Second, for the frames for which the number of bits notably exceeds the assignment, ΔT is at the upper bound of the control range (marked by $\langle 2 \rangle$). This means that the picture complexity in these scenes is significantly higher than the target. In order to preserve the image quality, we cannot enlarge the width of the threshold band. Some frames are skipped by both methods to achieve the target bit rate.

To be concrete, we calculate the rate-control error ratio (RCER) of each frame and show the average results of six sequences in Table II. It can be seen that, in all sequences with different numbers of levels, with the proposed algorithm the rate-control error ratio is significantly reduced. It also verifies that the number of bits produced by each frame matches the target value much better than that of previous work

$$L_o = \frac{R}{f} \tag{25}$$

$$\text{RCER} = \frac{|L_a - L_o|}{L_o}. \tag{26}$$

Because of the skip-frame control, both rate-control methods can output a final bit rate that closely matches the target bit rate shown in Table I. However, for the previous method, this is achieved at the expense of the output frame rate. As the proposed method can effectively control the encoding bit count of

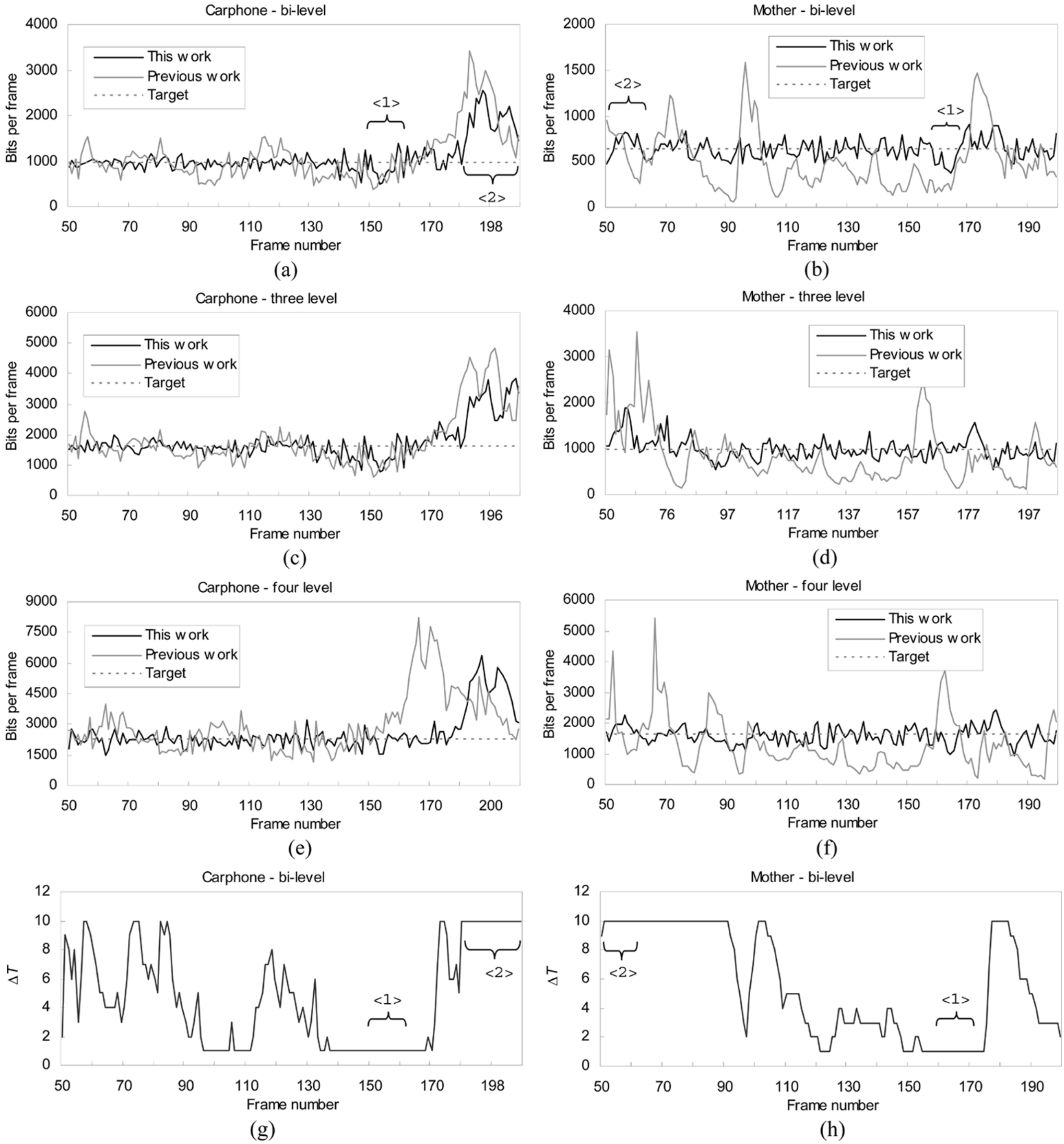


Fig. 14. Encoding bit count of each frame of (a), (b) bi-level video, (c), (d) three-level video, and (e), (f), four-level video, and (g), (h) the half-width of threshold band used in bi-level video.

each frame to the target in most cases, it can reduce the number of skipped frames. As shown in Table III, the output frame rates are obviously improved by the proposed method in most cases.

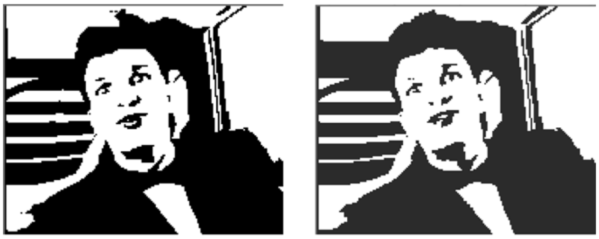
The proposed method not only effectively controls the bit rate and reduces the number of skipped frames, but also improves the visual quality of the decoded video compared to the previous method. As shown in Fig. 15, there is an obvious error in the hair region in image (a) which was generated by the previous method, whereas it is invisible in image (b) which was generated by the proposed method. There are two reasons. First, as mentioned in Section III-C, increasing the value of dissimilarity threshold T_d might introduce a much more significant loss in visual quality than enlarging the threshold band does. In the proposed method, T_d is fixed to a value that does not introduce visible loss in visual quality while the range of the threshold band is enlarged to $(T - 10, T + 10]$. Second, the proposed method can derive the coding parameters more accurately than the previous method does.

TABLE II
COMPARISON OF AVERAGE RATE-CONTROL ERROR RATIO BETWEEN PREVIOUS WORK AND THIS WORK (%)

	Bi-level		Three-level		Four-level	
	Old	New	Old	New	Old	New
Akiyo	42.45	25.28	36.67	21.62	34.98	21.14
Mother	59.44	25.10	49.96	27.24	59.39	24.78
Salesman	39.39	21.20	50.23	22.16	38.00	18.94
Carphone	62.21	21.23	57.50	22.17	73.49	24.68
Foreman	57.02	20.99	78.15	24.96	68.16	23.77
Coastguard	44.96	16.14	92.77	19.77	76.49	18.12

TABLE III
COMPARISON OF FRAME RATE BETWEEN THIS WORK AND PREVIOUS WORK (fps)

	Bi-level		Three-level		Four-level	
	Old	New	Old	New	Old	New
Akiyo	15.00	14.85	15.00	15.00	15.00	15.00
Mother	13.44	13.91	12.83	13.45	13.05	13.91
Salesman	14.60	14.63	13.26	13.46	14.00	14.40
Carphone	11.58	13.08	11.51	12.88	10.25	12.41
Foreman	11.80	13.25	10.50	12.50	10.40	12.60
Coastguard	12.25	13.65	8.10	13.10	9.10	13.45



(a) (b)

Fig. 15. Decoded image generated by (a) the previous method and (b) the proposed method.

VI. CONCLUSION

In this paper, we review the portrait video coding scheme and its rate-control algorithm and propose a novel model-based rate-control algorithm for portrait video.

In low-bandwidth conditions, portrait video possesses clearer shape, smoother motion, and much cheaper computational cost than DCT-based schemes. However, the bit rate of portrait video cannot be accurately modeled by a rate-distortion function as in DCT-based schemes. The previously proposed empirical rate-control method cannot effectively control the bit rate.

The main contribution we make in this paper is that we propose a model-based method that can effectively control the bit rate of portrait video. From the entropy coding principle, we know that reducing the number of LPSs reduces the bit rate, while increasing the width of threshold band reduces the number of LPSs. In addition, the picture complexity calculated by using entropy can accurately predict the maximal encoding bit count of a picture. Although the width of the threshold band is not directly related to the encoding bit count, the LPS reduction ratio that corresponds to the threshold-band half-width is computable, while the LPS reduction ratio is linear related to the bit-count reduction ratio. Therefore, we build an LPS-rate model to obtain the desired width of the threshold band inversely. Before coding a frame, the desired bit-count reduction ratio is obtained in terms of the picture complexity and the target bit count, and then the desired LPS reduction ratio is calculated using the LPS-rate model, hence the desired width of the threshold band is selected according to the LPS reduction ratio. This method is firstly proposed for bi-level video and then extended to multilevel video by using the relationship between bi-level video and multilevel video. Experimental results show that the proposed method not only effectively controls the bit rate, but also improves the output frame rate.

The principle of this method can also be applied to general bit plane coding in other image processing and video compression technologies.

REFERENCES

[1] *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mbit/s—Part 2: Video*, ISO/IEC 11 172-2, 1993.

[2] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13 818-2, 2000.

[3] *Coding of Moving Pictures and Audio*, ISO/IEC JTC1/SC29/WG11 N3312, Mar. 2000.

[4] "H.261 Video Codec for Audiovisual Services at $p \times 64$ kbit/s," ITU-T Recommendation, 1993.

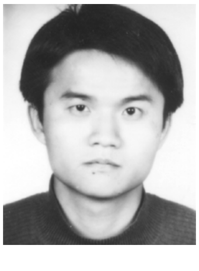
[5] "H.263 Video Coding for Low Bit Rate Communication," ITU-T Recommendation, 1998.

[6] J. Li, G. Chen, J. Xu, Y. Wang, H. Zhou, K. Yu, K. T. Ng, and H. Y. Shum, "Bi-level video: Video communication at very low bit rates," in *Proc. 9th ACM Int. Conf. Multimedia*, 2001, pp. 392–400.

[7] J. Li, K. Yu, T. He, Y. Lin, S. Li, and Q. Zhang, "Scalable portrait video for mobile video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 5, pp. 376–384, May 2003.

[8] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.

[9] *Coding of Still Pictures, 07/99*, ISO/IEC JTC1/SC29/WG11 (ITU-T SG8) N1359.

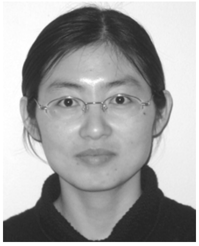


Keman Yu (M'03) received the B.S. and M.S. degrees in computer science and engineering from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively.

He joined Microsoft Research Asia, Beijing, China, as an Assistant Researcher in 2001 and became an Associate Researcher in 2003. His research interests include image/video compression and multimedia communication.

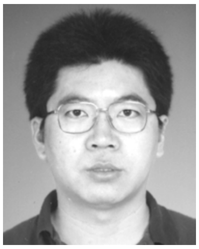
Jiang Li (M'99–SM'04) received the B.S. degrees in applied physics and applied mathematics from Tsinghua University, Beijing, China, in 1989, and the M.S. degree in optics and the Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, Zhejiang, China, in 1992 and 1998, respectively.

He is currently a Research Manager with the Media Communication Group, Microsoft Research Asia, Beijing. He joined the faculty of Zhejiang University as an Assistant Professor in 1992 and became a Lecturer and an Associate Professor in 1994 and 1998, respectively. He joined Microsoft Research Asia in 1999 and became the Project Leader of the Internet Media Group in 2002. He is now leading research on mobile video communication, multiparty conferencing, push-to-talk on IP across various platforms, peer-to-peer networking, and interactive multiview video.



Cuizhu Shi received the B.S. degree in information engineering, communication, and signaling engineering and the M.S. degree in electronic information and electric engineering from Shanghai Tongji University, Shanghai, China, in 2001 and 2004, respectively.

Her research interests include image processing and video compression.



Shipeng Li (M'97) received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 1988 and 1991, respectively, and the Ph.D. degree from Lehigh University, Bethlehem, PA, in 1996, all in electronic engineering.

He was with the Electrical Engineering Department, USTC, from 1991 to 1992. He was a Member of Technical Staff with Sarnoff Corporation, Princeton, NJ, during 1996–1999. He has been a Researcher with Microsoft Research Asia, Beijing, China, since May 1999, and now is a Research Manager with the Internet Media Group. His research interests include image/video compression and communications, digital television, multimedia, and wireless communication. He has contributed several technologies to the MPEG-4 international standard.