SIP (2013), vol. 2, e3, page 1 of 13 © The Authors, 2013.

doi:10.1017/ATSIP.2013.4

ORIGINAL ARTICLE

Constant frame quality control for H.264/AVC

CHING-YU WU, PO-CHYI SU, LONG-WANG HUANG AND CHIA-YANG CHIOU

A frame quality control mechanism for H.264/AVC is proposed in this research. The research objective is to ensure that a suitable quantization parameter (QP) can be assigned to each frame so that the target quality of each frame will be achieved. One of the potential application is consistently maintaining frame quality during the encoding process to facilitate video archiving and/or video surveillance. A single-parameter distortion to quantization (D–Q) model is derived by training a large number of frame blocks. The model parameter can be determined from the frame content before the exact encoding process. Given the target quality for a video frame, we can then select an appropriate QP according to the proposed D–Q model. Model refinement and QP adjustment of subsequent frames can be applied by examining the coding results of previous data. Such quality measurements as peak signal to noise ratio (PSNR) and structural similarity (SSIM) can be employed. The experimental results verify the feasibility of the proposed constant quality video coding framework.

Keywords: H.264/AVC, quality control, PSNR, SSIM, quantization parameter

Received 25 November 2012; Revised 13 March 2013

I. INTRODUCTION

H.264/AVC [1] is widely adopted in many applications these days due to its advanced coding tools. Under the same quality constraint, the bit-rate saving of H.264/AVC is significant when compared with such predecessors as MPEG-2 and MPEG-4. It should be noted that video frame quality is considerably affected by the quantization parameter (QP) assigned to each frame. Owing to varying contents in video frames, the quality may fluctuate a lot and careless assignment of QP may result in serious distortion in certain frames. This negative effect may not be acceptable in such applications of video surveillance and/or video archiving, since we require that the quality of each frame should be equally preserved well under these scenarios, in which the recorded video frames may be critically viewed afterwards. The objective of this research is to develop a distortion–quantization (D–Q) model so that a suitable QP can be assigned to each frame efficiently according to the frame content to help achieve constant quality video coding.

The measurement of quality has long been a research focus of video processing. The most commonly used metric is peak signal to noise ratio (PSNR), which is defined as

$$PSNR(x, y) = 10log_{10}\frac{255 \times 255}{MSE(x, y)}, \qquad (1)$$

where $MSE(x, y)$ is the mean squared error between two contents x and y, e.g., the original/reference frame and the coded/processed frame, respectively. Simplicity is the major advantage of PSNR and comparing different algorithms based on PSNR is easy. Although PSNR is sometimes questioned for its lack of representing subjective or perceptual quality, when the original video is available, PSNR still serves as a pretty good indicator of quality degradation from the process of lossy compression. To further reflect the subjective quality in measurement, many researchers [2–5] tried to take human visual systems (HVS) into account. Structural SIMilarity index, SSIM [2], is one of the well-known metrics. SSIM of two contents x and y is defined as

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \qquad (2)$$

where μ_x (μ_y) and σ_x (σ_y) are the local mean and standard deviation of x (y), respectively. σ_{xy} is the local correlation coefficient of x and y. C_1 and C_2 are small constants to avoid instability when the denominator is close to zero. Considered being more related to HVS, SSIM is also suggested these days to evaluate the quality of processed frame in video coding. The encoding algorithms explicitly employing SSIM have also been proposed [6, 7]. Since PSNR and SSIM are commonly used in video codec designs, we adopt them as examples to demonstrate the idea of constant quality coding. Other quality metrics that have a higher correlation with human perceptual quality can be better choices but their computational complexities may be too high to be used in a real-time encoding system.

To achieve constant quality video coding, one may think that using a fixed QP value to encode the entire video may work. Figure 1 shows an example of encoding the video "Foreman" with a fixed QP equal to 30. We can see that

Department of Computer Science and Information Engineering, National Central University, Jhongli City, Taoyuan 32001, Taiwan. Phone: +886-3-4227151 ext.35314

Corresponding author: Po-Chyi Su
Email: pochyisu@csie.ncu.edu.tw

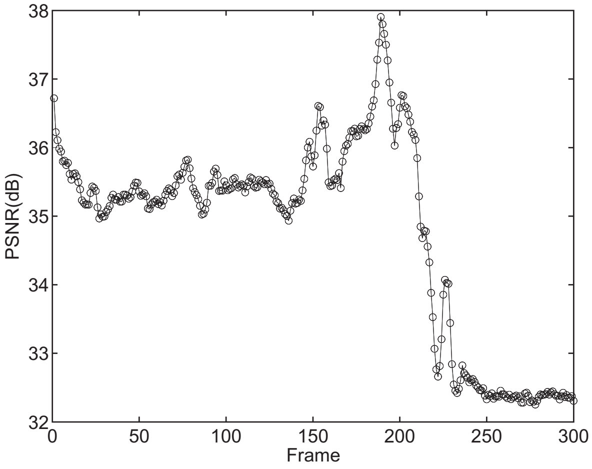

Fig. 1. PSNR variation when the fixed $QP = 30$ is used to encode the video Foreman.

the PSNR values vary and that smaller QP values should have been assigned in the latter part of this video. A similar problem exists if SSIM is used as the measurement. Therefore, constant quality video coding is not a trivial issue and extra attention should be paid to the encoder. Up to now, most of the existing work related to constant quality video coding adopted PSNR as the measurement. Huang *et al.* [8] proposed one of the early researches by encoding the video several times and employing the Viterbi algorithm to pick a suitable QP for each frame. To be more specific, a trellis structure is formed with each node representing a QP value. After encoding the video with different QP's, a few nodes resulting in similar PSNR values are clustered. By connecting nodes (of adjacent frames) in clusters, we can assign a QP value for each frame so that the resulting PSNR values are within a pre-defined range. However, since every frame has to be encoded several times, this scheme is quite time-consuming and only acceptable in offline applications. To attain more efficient quality control, D–Q and/or rate-distortion (R–D) models are developed to facilitate QP assignment. Ma *et al.* determined the relationship between PSNR and QP to develop a rate–quantization (R–Q) model for effectively allocating bit budgets [9]. Kamaci *et al.* made use of Cauchy-density function to depict the distribution of AC coefficients after block discrete cosine transform for developing an effective D–Q model [10].

In [11], sum of absolute transform differences is used to determine the related parameters of a D–Q model, which can accurately predict the PSNR in intra coded frames. De Vito *et al.* assigned or adjusted the QP values according to the difference between the average PSNR of previously encoded frames and the target PSNR [12]. If the difference is small, the QP of previous frame is used. Han *et al.* encoded the video twice and used the information of first-run encoding as the reference to attain constant quality coding [13]. In our opinion, the major drawback of the existing methods is the requirement of encoding the video several times. In addition, a practical D–Q model has not been successfully developed. In this research, we aim at proposing a framework, which can adopt more flexible quality measurements, to achieve constant quality video coding. Before encoding a frame, we will approximately predict its D–Q relationship from frame content to help determine a suitable QP such that the resultant quality is close to the target value. The model parameters should be content adaptive since frames with different characteristics should have varying D–Q relationships. Different from the existing approaches, we do not encode every single frame several times to collect the data points for forming the D–Q curve. A trained content adaptive D–Q model is built for assigning a QP value efficiently and most of the frames will thus be encoded just once. A few frames will be encoded at most twice to pursue the objective of constant quality encoding and to avoid significant increase of encoding time as well.

The rest of the paper is organized as follows. Model training of our proposed scheme is described in Section II and the complete QP assignment procedure is presented in Section III. Section IV demonstrates the experimental results, followed by the conclusion in Section V.

II. D–Q MODEL

As we aim at building a model that links the distortion and QP, the measurement of distortion has to be defined first. The measurement of distortion based on PSNR, D_{PSNR}, is related to MSE and we can simply use the sum of squared errors (SSE) as the measurement. For SSIM, it will be close to one if the contents to be compared are similar, so we define the distortion D_{SSIM} as $1 - SSIM$. It is observed that a power function can reasonably depict the relation between D_{PSNR} and quantization in both intra- and inter-coding. We employ the following function to describe the relationship, *i.e.*,

$$D_{PSNR} = \alpha \times QP^{\beta}, \tag{3}$$

where α and β are the two model parameters. It should be noted that most of the existing algorithms used *Qstep* in the fitting function while we choose *QP* instead. The reason for doing so is to develop a single-parameter model, which will be explained later. To verify the power function, we encode some test CIF videos, including Foreman, Coastguard, Container, Football, Mobile, Paris, and Stefan, each with 100 frames, by using intra coding with QP's ranging from 20 to 40 and record the corresponding D_{PSNR}. The curves from the collected data are matched with the above power function by regression. The R^2 values are all very close to one, which means that the chosen function can fit the data very well. In fact, by replacing D_{PSNR} by D_{SSIM}, we also observe a similar relationship. Again, the R^2 values are almost equal to one. However, we list the parameters, α and β in Table 1 and we can see that the two values vary in each video. Existing work usually chose to train some data in the same video or employ the data in the previously decoded frames to acquire these parameters for subsequent encoding. The major disadvantage is that quality fluctuation may be observed in the

Table 1. The relationship between distortion and QP.

	D_{PSNR} vs. QP		D_{SSIM} vs. QP	
Video	α	β	α	β
Foreman	3.38×10^{-2}	5.15	2.30×10^{-6}	3.00
Coastguard	8.20×10^{-2}	5.07	2.75×10^{-8}	4.37
Container	1.29×10^{-2}	5.48	1.26×10^{-5}	2.57
Football	3.03×10^{-2}	5.23	4.64×10^{-7}	3.52
Mobile	3.76×10^{-4}	6.66	3.39×10^{-10}	5.40
Paris	4.50×10^{-4}	6.50	2.93×10^{-8}	4.18
Stefan	2.09×10^{-4}	6.73	1.74×10^{-10}	5.49

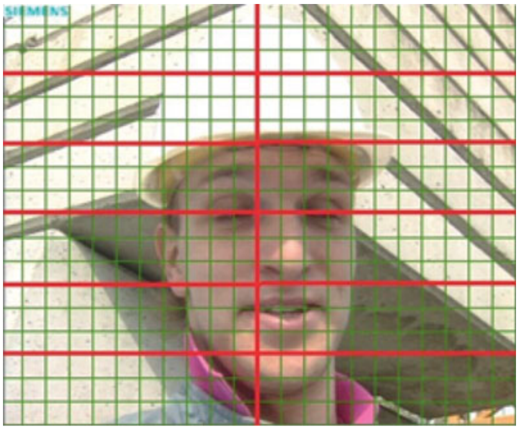

Fig. 2. Partition of a frame into basic units.

first few encoded frames if inappropriate parameters are set. More encoding processes may thus be required. In addition, when the scene changes happen, the parameters have to be determined again or the performance will be affected seriously.

The objective of this research is to appropriately estimate these parameters by using a content-adaptive model. The first step is to collect various data samples for training. To begin with, the frame will be divided into basic units. There are several choices for deciding the size of basic units, e.g., an entire frame, a group of macroblocks (MB's) or a single MB. Designing a frame model, i.e., determining a QP value according to the feature representing the entire frame, sounds a reasonable and straightforward approach. A feature representing the frame is computed to determine α and β in equation (3) for the whole frame. However, we found that a slight model inaccuracy will result in poor determination of QP. Using MB's directly for model training should be more flexible. Nevertheless, according to our experience, when the unit size is too small, it will be difficult to determine a well-defined relationship between the content and the parameters. An obvious example is that we may easily obtain small blocks with uniform colors and encoding such blocks with different QP values may generate unexpected results. It is worth noting that such blocks occupy a large portion in common frames. In other words, there will be a large number of outliers in our training data. Training the model with so many "unusual" blocks will be challenging and the model parameters may not be acquired accurately. Therefore, we choose to use a group of MB's as the basic unit in our framework. For a CIF video frame, we divide it into basic units as shown in Fig. 2. A unit contains 33 MB's so a frame contains 12 basic units. Such division may look a bit awkward but we have a reason for this choice. By dividing the frame across the center as shown in Fig. 2, we can obtain blocks or basic units that contain meaningful content more easily since there are usually important objects at the center of a frame. Besides, the units should be reasonably large too. In other words, we expect that a unit can consist of areas with different characteristics so that the number of outliers can be reduced to facilitate the training process. Furthermore, a larger number of "meaningful" units certainly helps QP determination.

We first deal with the intra-coded frames. Since many frames in a video have similar content, we do not use video sequences for training but select still images. We use 200 images from Berkeley image database [14]. Each image is scaled and cropped properly to the CIF frame size. These images are concatenated into a video, which is encoded with various QP's. The quality distortion of each basic unit and the corresponding QP values are collected. The relationship between the distortion and QP shown in equation (3) still holds. A very important finding is that there exists a linear relationship between $ln(\alpha)$ and β for both PSNR and SSIM as shown in Fig. 3. The R^2 values of using this linear relationship are both as high as 0.99. The fact indicates that equation (3) can be reduced to only one variable. For I-frames, the D–Q model can thus be expressed as

$$D^I_{PSNR} = e^{-2.83\beta+9.06} \times QP^\beta \tag{4}$$

for PSNR, and

$$D^I_{SSIM} = e^{-3.35\beta-3.32} \times QP^\beta \tag{5}$$

for SSIM. These relationships are derived by regression. In fact, according to our tests, a similar relation can also be found in P-frames and the data can be fitted well by

$$D^P_{PSNR} = e^{-2.91\beta+10.06} \times QP^\beta \tag{6}$$

and

$$D^P_{SSIM} = e^{-3.48\beta-2.55} \times QP^\beta. \tag{7}$$

The R^2 values in P-frames can also reach 0.99 in both distortion measurements. In our opinion, since PSNR and SSIM perform quite differently, such a relationship may exist in many different quality metrics. If PSNR is adopted as the quality metric, we can use Equations (4) and (6) to determine mapping between QP and the distortion for a given frame. For SSIM, Equations (5) and (7) will be employed.

The next step is to seek an efficient way to choose suitable β for a basic unit. It is worth noting that β is content-related. According to our observations, if the content can be affected by lossy coding more easily, the value of β will be larger. On the other hand, for the unit with relatively more uniform content, β will be quite small. Therefore, we would like to

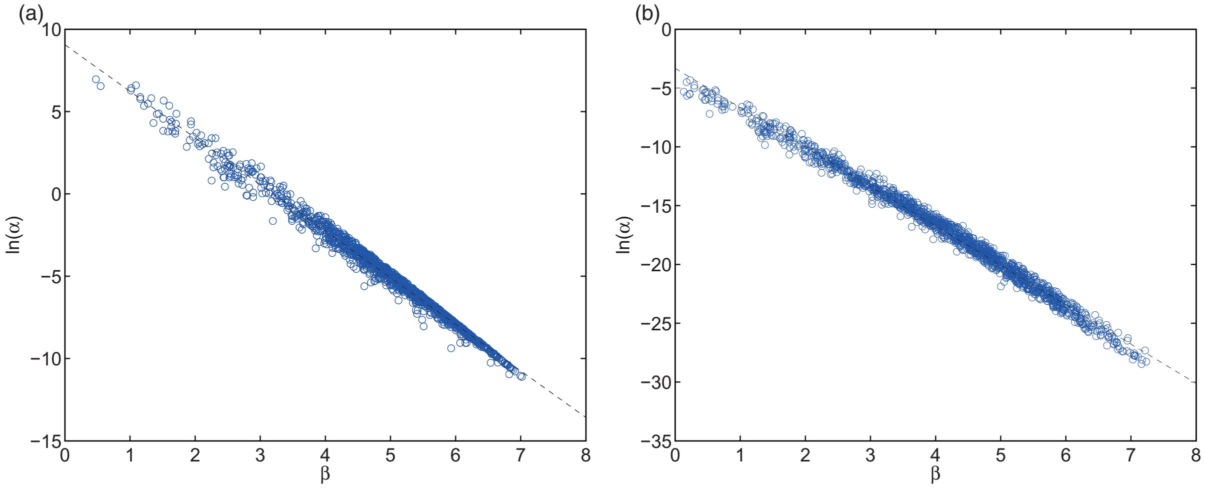

Fig. 3. The relationship between $ln(\alpha)$ and β for using (a) PSNR and (b) SSIM as the measurement in I-frames.

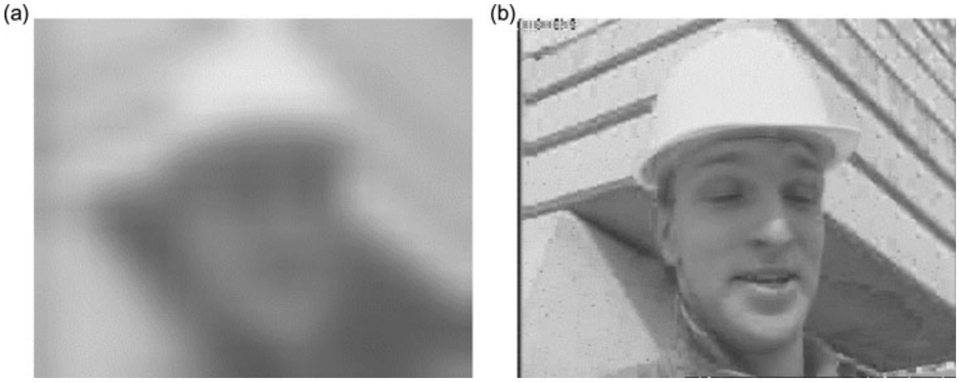

Fig. 4. The preprocessed frames of Foreman by (a) resizing and (b) SVD.

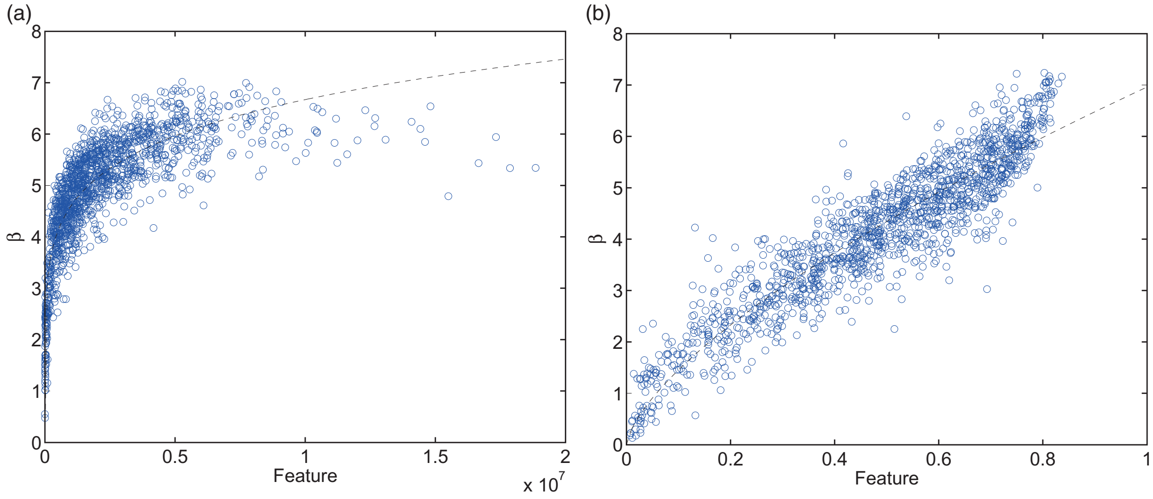

Fig. 5. The relationship between the extracted feature and β for (a) PSNR and (b) SSIM in I-frames.

predict the effects of compression on content so that a reasonably good β can be selected. One way to achieve this is to encode the frame with different QP's to observe the curve but it may be computationally prohibitive. In other words, this "pre-processing" has to be efficient to avoid considerable increase in the load of video coding. Besides, we aim at developing a more general framework for constant quality H.264 video coding, in which the distortion measurement may be different in targeted applications. We thus adopt the following strategy. The pre-processing or, in fact, a process of distortion is applied on the input frame and then the selected quality measurement will be used to evaluate the degradation of these distorted versions. That is, we make use of these degradation measurements to help us select a suitable β.

Again, we collect training data for coding with different QP's to determine β and, at the same time, preprocess these training data to obtain the distortions. By examining β and the degradations, we would like to know whether such a solid relationship exists. After various trials, the preprocessing we consider right now includes two parts: resizing and singular value decomposition (SVD). The resizing process quickly removes high-frequency textures. We simply calculate the 16 × 16 block means to obtain a down-sampled version of an input frame. Then, this small frame is filtered by a 3 × 3 Gaussian low-pass filter. Finally, we linearly interpolate it to form the frame with the original frame size. Figure 4(a) shows a seriously blurred version of Foreman. The reason for removing high-frequency textures is to predict the effects of lossy compression as these parts are affected more. The other process is applying 16 × 16 block SVD after the block mean is removed. We then use the block mean and the important eigenvectors/eigenvalues to reconstruct the block. Such blocks will contain significant content and can serve as reliable references to see what may be left after coding. The first and second eigenvector pairs are used to reconstruct the block as shown in Fig. 4(b). Although the blocky artifacts are seen, the content can still be preserved quite well. In addition, we found that this SVD process performs better in blocks with more textures. Given these two pre-processed or distorted frames, we calculate their quality degradation (D_{PSNR} or D_{SSIM} for now) compared with the raw input frame. Then, the two distortion measurements are combined to form a so-called "content feature" for evaluating the single parameter β in our model. Since it can be shown from Fig. 4 that the degrees of distortions in these two steps are quite different as resizing results in more serious quality degradation, the two evaluations are weighted and summed to form the feature. In our training data evaluated in SSIM, the average distortion for resized frames, D_{SSIM}^{resize}, is around $K = 4$ times that of SVD processed frames, D_{SSIM}^{svd}. We thus calculate the "spatial feature", $F_{SSIM}^{spatial}$, by

$$F_{SSIM}^{spatial} = 0.2 \times D_{SSIM}^{resize} + 0.8 \times D_{SSIM}^{svd}, \tag{8}$$

which will be used to determine β. In the case of using PSNR, K is around 5.5 and the two values are weighted accordingly to obtain $F_{PSNR}^{spatial}$, i.e.,

$$F_{PSNR}^{spatial} = 0.15 \times D_{PSNR}^{resize} + 0.85 \times D_{PSNR}^{svd}. \tag{9}$$

Figure 5 shows the relationship between the extracted feature and β in the training data of I-frames. We also found that the data are clustered and can be fitted well by using regression. For PSNR, the data can be depicted reasonably well by

$$\beta = 0.49 \times (F_{PSNR}^{spatial})^{0.16}. \tag{10}$$

The fitting function for SSIM is

$$\beta = 6.96 \times (F_{SSIM}^{spatial})^{0.68}. \tag{11}$$

Since only intra-coding is applied in I-frames, the feature for I-frames, $F_{PSNR/SSIM}^{I}$, is simply $F_{PSNR/SSIM}^{spatial}$.

In P-frames, temporal information is required. As in regular video coding, we apply motion estimation with 16 × 16 blocks and with the searching range set as ±8 to form a motion compensated frame. Only the integer positions are searched. Similar to what we have done for I-frames, the distortion of this compensated frame is computed to determine the temporal feature, $F_{PSNR/SSIM}^{temporal}$. However, since intra-coding may still be employed on P-frames, we also calculate the spatial feature, $F_{PSNR/SSIM}^{spatial}$, and use the average of the two features to determine most of the P-frame features by

$$F_{PSNR/SSIM}^{P} = 0.5 \times F_{PSNR/SSIM}^{spatial} + 0.5 \times F_{PSNR/SSIM}^{temporal}. \tag{12}$$

The method of calculating the average value to form the feature does look a bit heuristic and one may even think of estimating the percentages of intra and inter coding in a frame to decide a more suitable weighting function. However, whether a block will be intra or inter coded may depend on the QP value. A block may become intra-coded when a smaller QP is used. Applying the block type prediction or classification before the QP assignment is thus less reasonable. In addition, separating the intra and inter coding in the model training process of P-frames is rather complicated. Therefore, we choose to take both spatial and temporal characteristics into account to form a P-frame feature and resort to the simplified model training process on a large number of collected data to achieve good performances. Figure 6 shows the relationship between the feature and β in P-frames in the case of PSNR and SSIM. Although some outliers exist, the fitting can still be good enough to help us choose suitable QP values of P-frames. The fitting curve for PSNR is

$$\beta = 0.34 \times (F_{PSNR}^{P})^{0.17} \tag{13}$$

and that for SSIM is

$$\beta = 17.32 \times (F_{SSIM}^{P})^{0.96}. \tag{14}$$

As mentioned before, we will calculate the feature to determine the single parameter β for each basic unit. Then,

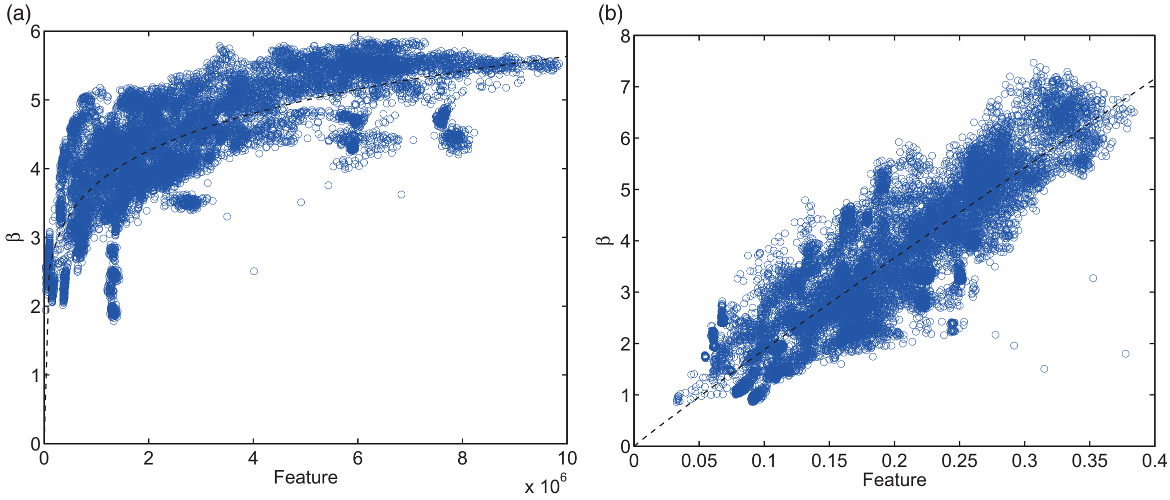

Fig. 6. The relationship between the extracted feature and β for (a) PSNR and (b) SSIM in P-frames.

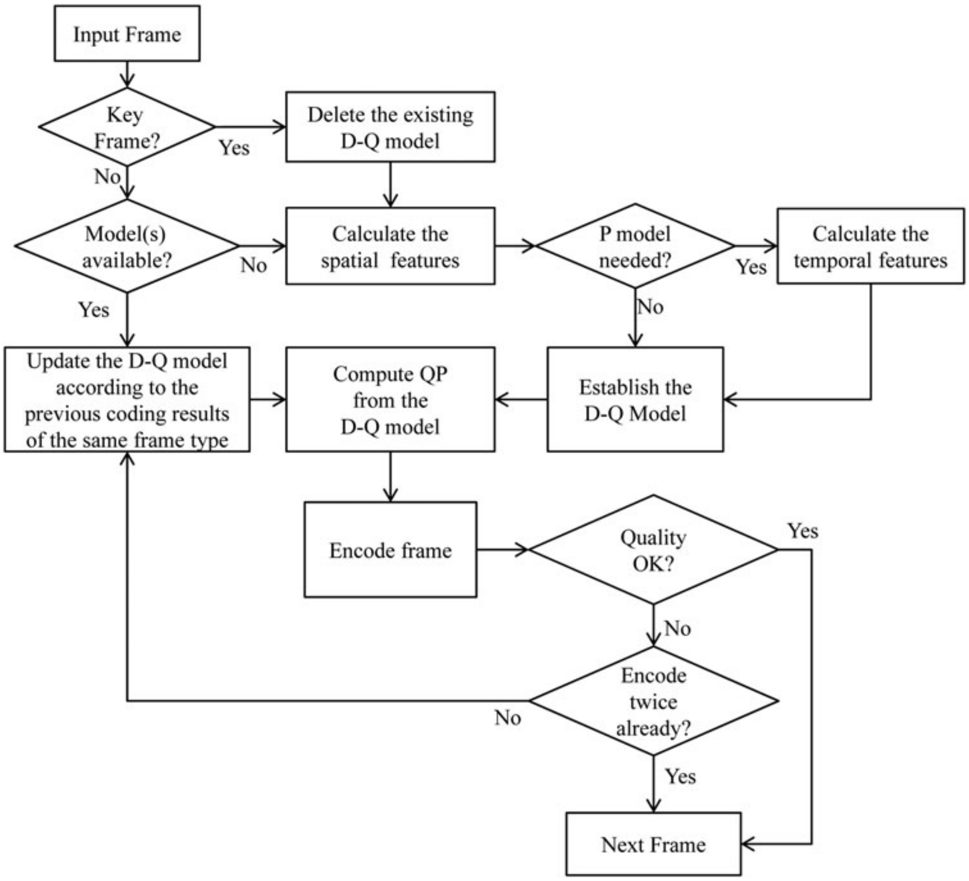
Input Frame
Key Frame?
Yes
No
Delete the existing D-Q model
Model(s) available?
No
Yes
Calculate the spatial features
P model needed?
Yes
No
Calculate the temporal features
Update the D-Q model according to the previous coding results of the same frame type
Compute QP from the D-Q model
Establish the D-Q Model
Encode frame
Quality OK?
Yes
No
Encode twice already?
No
Yes
Next Frame

Fig. 7. The flowchart of the encoding procedure.

the frame QP, QP_F, is determined such that the overall distortion will be as close to target distortion as possible. That is,

$$QP_F = \arg\min_{QP\in[0,51]} \sum_{i=1}^{12} (D^{(i)}(QP) - D^{target})^2, \quad (15)$$

where $D^{(i)}(QP)$ is the distortion of the ith basic unit estimated by Equations (4), (6) or Equations (5), (7), and D^{target} is target distortion. The use of 12 units helps to reduce the negative effects from possible model inaccuracy of a single unit.

III. THE ENCODING PROCEDURE

Our objective is to strictly maintain the quality of each frame. That is, after a target distortion is set, e.g., PSNR equal to 40 dB or SSIM equal to 0.92, the distortion measurement of each decoded frame should reach the target value as close as possible so that constant quality coding can be successfully achieved. With the proposed D–Q model, the selection of QP can be done in a straightforward manner. Given an input frame, the feature is computed to determine the D–Q relations of basic units and the frame QP will be chosen according to Equation (15). There are a few issues that will affect the designs of our proposed encoding procedure. First, adjacent frames in a video usually have similar content, which will result in similar features. Then, calculating features in each frame does not seem that necessary. The spatial feature $F^{spatial}$ is relatively efficient but the temporal feature $F^{temporal}$ is more time-consuming because of motion estimation. Therefore, if we can reuse the feature of a previous frame with similar content for computing the model parameter β, the whole encoding procedure will be more efficient. In other words, the spatial and temporal features of a frame will only be re-calculated if such a feature with the same frame type or similar content is not available. Second, the quality of the reference frame will affect that of the currently encoded frame. Especially when a scene change frame appears and its QP is not appropriately assigned. The quality of the subsequent frames may be poor and larger quality variations may also be observed, especially when a scene change frame appears and its QP is not appropriately assigned. Our strategy is to apply the scene change detection to determine the so-called key frames to build the D–Q model. We will then encode these frames carefully, probably with two runs, so that the quality of subsequent frames can also be maintained. Third, as mentioned before, the content of adjacent frames will be similar. If the frame coding types are also the same, the coding results of the previous frame can serve as a good indication of model accuracy. Therefore, the coding performance of the previous frame of the same type will be examined for model adjustment so that single-run coding may work as well as multiple-run coding. The flowchart of the encoding process is demonstrated in Fig. 7 and explained as follows.

A simple scene-change detection process by examining the luminance histograms of adjacent frames is adopted. The Bhattacharyya distance of two histograms is calculated and compared with a threshold. If the difference is larger than the threshold, a scene change is detected and we call this scene change frame as the key frame. It should be noted that, although the key frame may need to be encoded as a P-frame, we only use the spatial feature $F^{spatial}$ to calculate β, instead of using the P-frame feature F^P shown in Equation (12), because a large number of intra-coded blocks will appear in this frame. After using $F^{spatial}$ to determine the D–Q relation and the frame QP for encoding this frame, we usually encode this frame once again if the resulting quality of this decoded frame is not close to the target value. This two-pass encoding is to ensure that these important scene-change frames have the targeted quality. The model will be slightly adjusted according to the first-run encoding results. We call this process the model update, which actually has an additional adjusting factor θ defined by

$$\theta = \frac{D^p(QP_F)}{e^{(a\times\beta+b)} \times QP_F^{\beta}}, \tag{16}$$

where a and b are the trained variables listed in Equations (4)–(7). That is, the denominator is the predicted distortion by our model and $D^p(QP_F)$ is the resulting distortion by using QP_F to encode the frame in the first run. In the second-run encoding of this frame, the model becomes

$$\theta \times D^{I/P}_{PSNR/SSIM}, \tag{17}$$

where $D_{PSNR/SSIM}$ is defined in Equations (4)–(7), and a better QP_F can then be chosen accordingly. In other words, we simply adjust the parameter α in Equation (3) and this strategy is quite effective. Figure 8 shows the comparison of coding results on Foreman by using the original model and those by using the updated model with θ. In Figs 8(a) and 8(c), we encode all the frames by using intra coding only. In Figs 8(b) and 8(d), only the first frame is an I-frame and the other frames are coded as P-frames. The qualities of P-frames are then averaged. We select Foreman in this test since it contains large content variations and our original model does not perform that well. By using the simple scaling factor θ, the predicted quality, measured in either PSNR or SSIM, will be close to the actual quality after model adjustment in second-run encoding.

For other frames, we will use the coding result of the previous frame with the same frame type as the reference to adjust our D–Q model. That is, θ will be computed by dividing the resulting distortion of the previous frame (i.e., $D^p(QP_F)$ in Equation (16)) by the predicted distortion so that most of the frames will be encoded only once. As mentioned before, only the spatial feature $F^{spatial}$ will be used to find β in the scene-change frames. It should be noted that there will be a couple of special cases for other frames. (1) For the first P-frame after the scene-change frame, since its temporal feature $F^{temporal}$ is not available, we will calculate its own feature F^P. In addition, since the previous P-frames do not have similar content, this P-frame may be encoded twice without referring to the coding results of previous frames. (2) For the first I-frame after the scene-change frame, we will calculate its own $F^{spatial}$ to calculate β and may also encode this frame twice to use its own first-run coding results for model adjustment. To sum up, the features will be computed and the coding may be applied twice in the following three cases: (1) The scene-change or key frame, (2) the first I-frame after the key frame, and (3) the first P-frame after the key frame. For most of the other I/P-frames, we basically employ the existing features and use the coding results of the frames with similar content and with same frame type for model adjustment. Then, the calculation of the features will not be applied repeatedly. Finally, to achieve extremely consistent video quality, the coding result of each frame will be checked. If the result deviates from the target too far, we may encode that frame once more and the model is also adjusted by Equation (17).

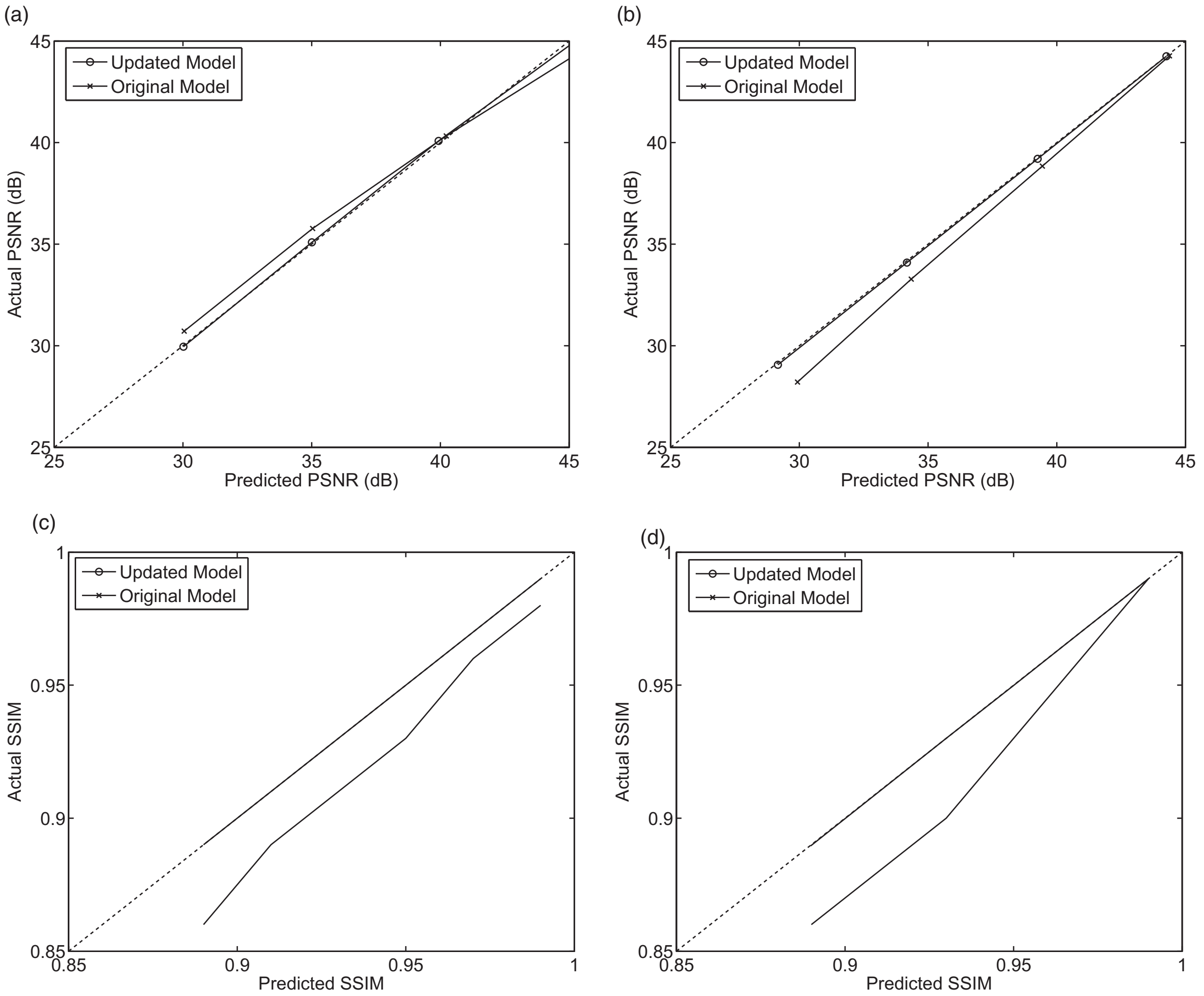

Fig. 8. The comparison of coding results of Foreman by using the original model and updated model in the case of (a) PSNR in I-frames, (b) PSNR in P-frames, (c) SSIM in I-frames, and (d) SSIM in P-frames.

In our scheme, if the absolute difference of target PSNR and the resulting PSNR is larger than 0.25 dB, the frame will be encoded again. In SSIM, the threshold of absolute difference is set as 0.015. A frame will not be encoded more than twice to maintain the efficiency of the proposed method.

IV. EXPERIMENTAL RESULTS

We implemented our scheme in JM 15.1 reference software of H.264/AVC [15] to evaluate the performances of our proposed D–Q model and encoding procedure. The settings are as follows:

(1) Rate distortion optimization is enabled.
(2) Motion search range for coding is ±16.
(3) Fast full search algorithm is used.
(4) CAVLC is used.
(5) De-blocking filter is enabled.

We set the target PSNR as 30, 35, 40, and 45 dB and target SSIM as 0.91, 0.95, and 0.99 to test the feasibility of our scheme on different quality measures. SSIM is calculated in 8 × 8 blocks without overlapping. Eight CIF videos including Coastguard, Monitor, Table, Foreman, Mobile, Stefan, News, and Paris, each with 300 frames, are used in our experiments. Figures 9 and 10 show the performance of constant quality video coding measured in PSNR and SSIM, respectively. We can see that the resulting quality can achieve target quality in all of the cases. When the target quality is set lower, the variations of both PSNR and SSIM become larger because of wider range of QP. The variations are more obvious in the latter part of Foreman because of fast camera motions. Two-pass encoding is not applied very often and the most frequent case happens when the target SSIM is set as 0.91 in Foreman, in which only 16 out of 300 frames are encoded twice. In other videos such as Monitors and Mobile, except for the first two frames, which are the first I- and P-frames, respectively, and do not have any previous coding results, other frames are encoded just once.

Table 2 compares the performance of our quality control algorithm with the one proposed by De Vito *et al.* [12], in which the PSNR and QP values in previous frames are used to maintain constant quality in one pass. Five sequences

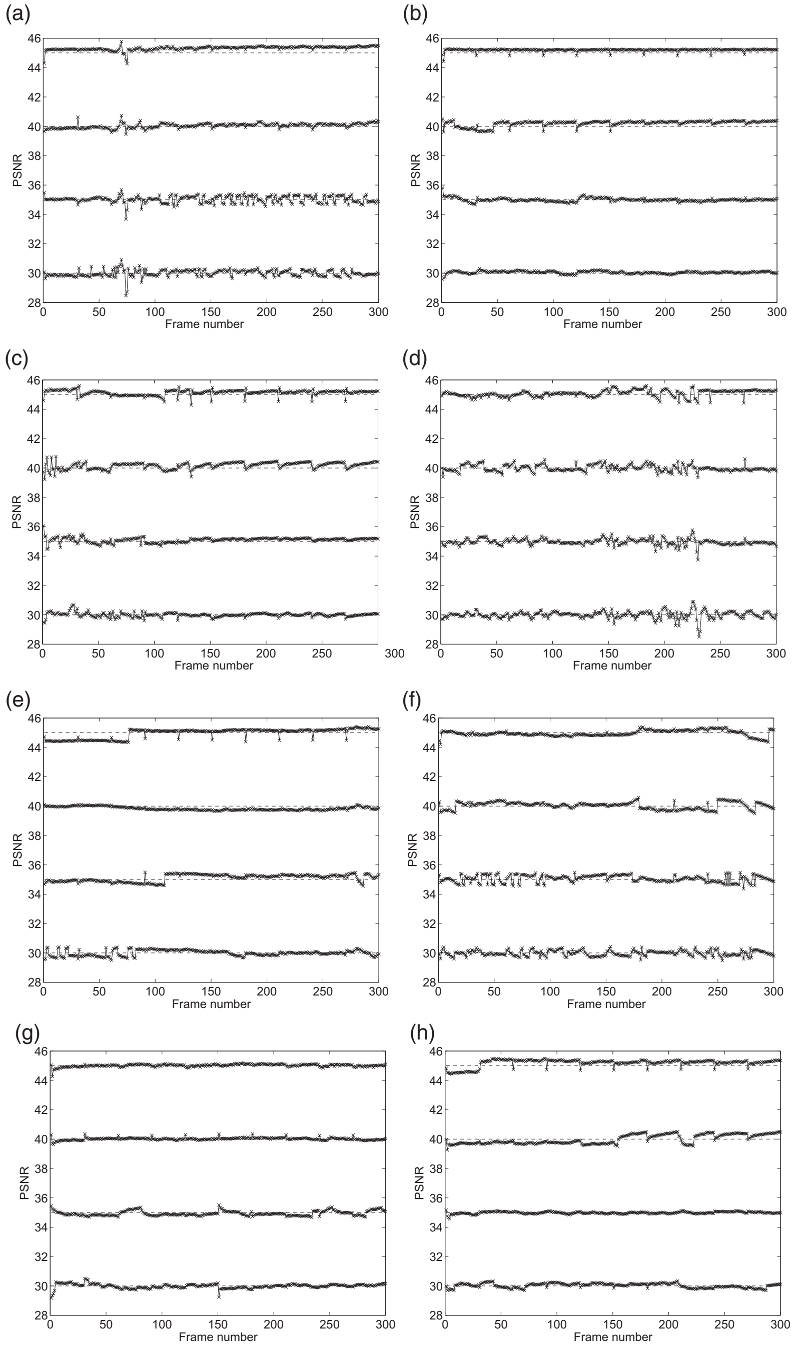

Fig. 9. The performances of constant quality (PSNR) video coding of (a) Coastguard, (b) Monitor, (c) Table, (d) Foreman, (e) Mobile, (f) Stefan, (g) News, and (h) Paris.

at three different target PSNR values are tested. The average absolute deviations of PSNR are 0.42 and 1.02 dB in our method and [12] respectively. The average PSNR variances are 0.06 and 0.25 dB in our method and [12], respectively. Therefore, our scheme can achieve better performances of constant quality coding. Table 3 shows the other comparison of our scheme with [13] and [16], both of which are two-pass schemes. That is, they will apply first pass encoding to estimate the R–D curve and second pass encoding to achieve constant quality coding. We use the resulting PSNR values of [13, 16] as targets to compress the videos by our scheme. We can see that the average PSNR values are close to the targeted ones and the PSNR variances of our scheme are lower than those of the other two methods. It should be noted that our scheme is more efficient since most of the frames are encoded only once.

Furthermore, we demonstrate the performances of proposed quality control in videos with a larger resolution. Four 4CIF (704 × 576) videos are tested and the performances of maintaining SSIM are shown in Fig. 11. The size of basic unit is set as 11 × 3 MB’s, the same with what we have done on CIF videos, and the same model parameters are also employed. The reasonably good performances in Fig. 11 indicate that, as long as most of the basic units used in

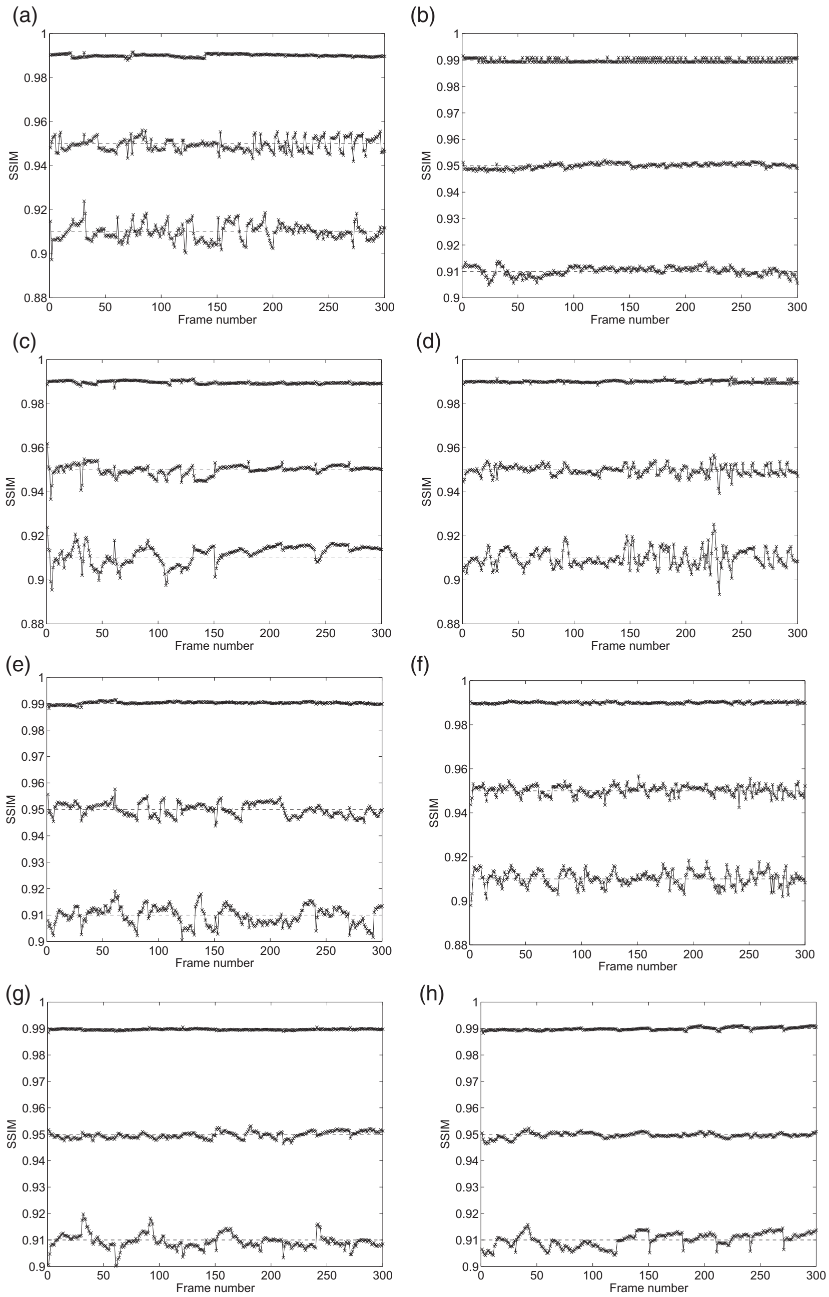

Fig. 10. The performances of constant quality (SSIM) video coding of (a) Coastguard, (b) Monitor, (c) Table, (d) Foreman, (e) Mobile, (f) Stefan, (g) News, and (h) Paris.

Table 2. Performance comparison of our scheme with [12].

	PSNR in dB (σ^2)					
	Target PSNR: 30 dB		Target PSNR: 33 dB		Target PSNR: 36 dB	
Video	Proposed	[12]	Proposed	[12]	Proposed	[12]
Foreman	29.87(0.10)	29.93(0.22)	32.90(0.11)	33.07(0.24)	35.97(0.10)	35.92(0.13)
Paris	29.83(0.02)	30.14(0.25)	32.84(0.02)	32.52(0.11)	36.03(0.02)	35.69(0.09)
News	29.90(0.05)	29.78(0.61)	32.83(0.01)	33.31(0.39)	36.07(0.10)	36.15(0.29)
Table	29.88(0.11)	29.70(0.36)	32.94(0.04)	33.15(0.22)	36.01(0.03)	36.14(0.17)
Stefan	29.92(0.04)	29.85(0.25)	33.02(0.07)	32.66(0.25)	36.02(0.06)	35.84(0.23)

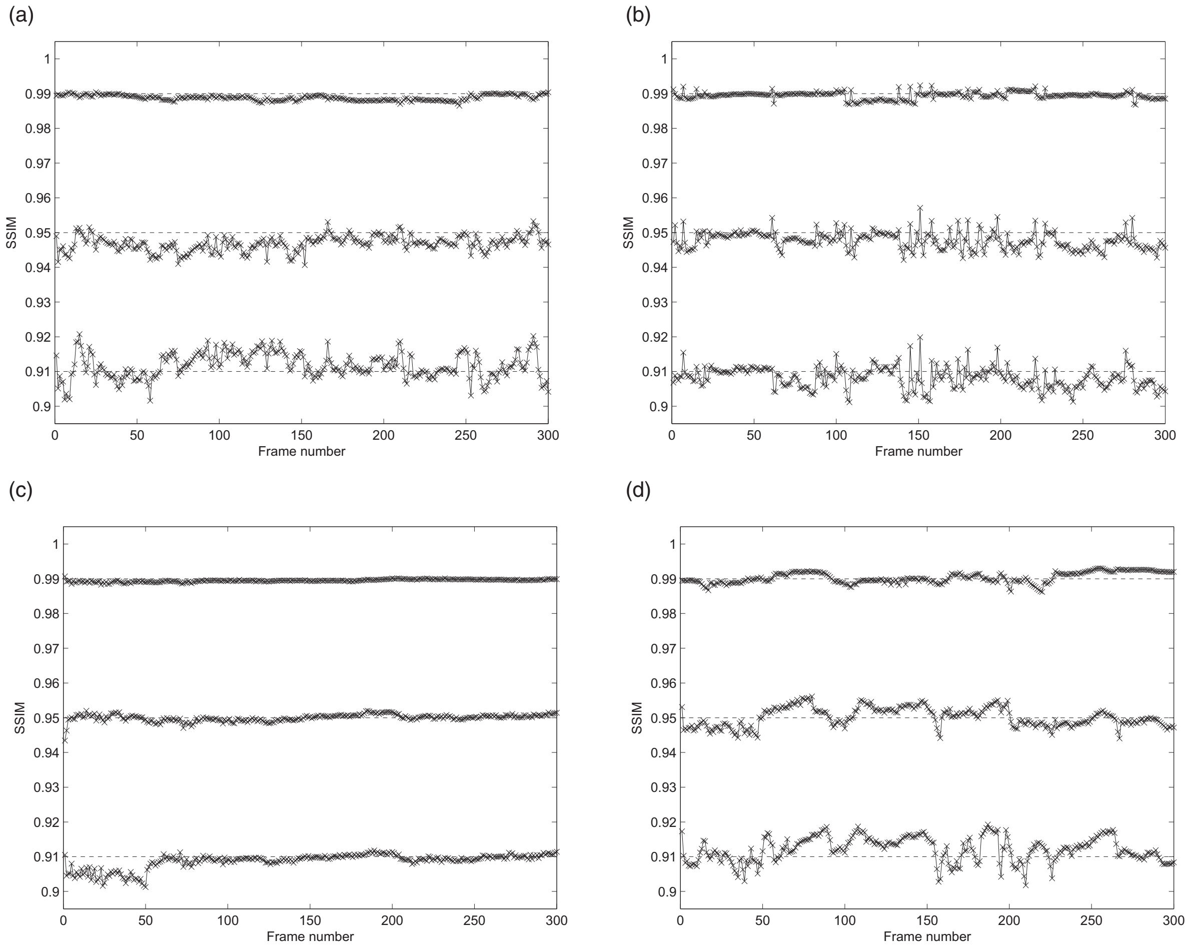

Fig. 11. The performances of constant quality (SSIM) video coding in 4CIF videos: (a) City, (b) Crew, (c) Harbor, and (d) Soccer.

Table 3. Performance comparison of our scheme with [13, 16].

	PSNR in dB (σ^2)			
Video	Proposed	[13]	Proposed	[16]
Foreman	37.04(0.05)	36.98(0.25)	36.73(0.04)	36.72(0.39)
Paris	33.87(0.02)	33.89(0.12)	33.79(0.03)	33.68(0.06)
Mobile	29.69(0.04)	29.65(0.46)	31.22(0.05)	31.15(0.12)
Table	40.75(0.02)	40.78(0.12)	39.09(0.02)	39.07(0.07)
Stefan	32.41(0.05)	32.38(0.32)	31.87(0.03)	31.81(0.08)

the training process contain meaningful contents, the built model can work well in videos with different resolutions.

Finally, we would like to discuss the strategy of video encoding involving B-frames. In our framework, we choose not to train the models of B-frames for the following reasons. First, the number of B-frames (between P-frames) may vary according to the settings of encoders. Training models with different parameter settings is not a flexible approach. Second, several prediction modes can be used in a B-frame, including list 0, list 1, bi-predictive, and direct predictions. As mentioned earlier, we will not perform block classification before the exact encoding process hence it will be difficult to determine reasonable features and the corresponding weighting factors. Therefore, instead of training the models for B-frames, we propose a simple QP determination method by assigning the QP value according to the related P-frames. More specifically, the QP value for a B-frame is set as $\lfloor \frac{QP_{list0}+QP_{list1}}{2} \rfloor$ when both list 0 and list 1 are available and both of them are encoded as P-frames. If one of list 0 and/or list 1 is unavailable or not a P-frame, the QP value of the only reference P-frame will be used to encode the current B-frame. Figure 12 illustrates the performances of the proposed B-frame QP determination. We can see that the target PSNR can be achieved in all of the sequences. Although the quality variations are a bit larger than those shown in Fig. 9, the performances are still satisfactory.

V. CONCLUSION

In this research, a frame quality control mechanism for H.264/AVC is proposed. A suitable QP can be assigned in each frame so that target frame quality can be achieved. A single-parameter D–Q model is derived and the model parameter can be determined from the frame content. The results by using such quality measurements as PSNR and

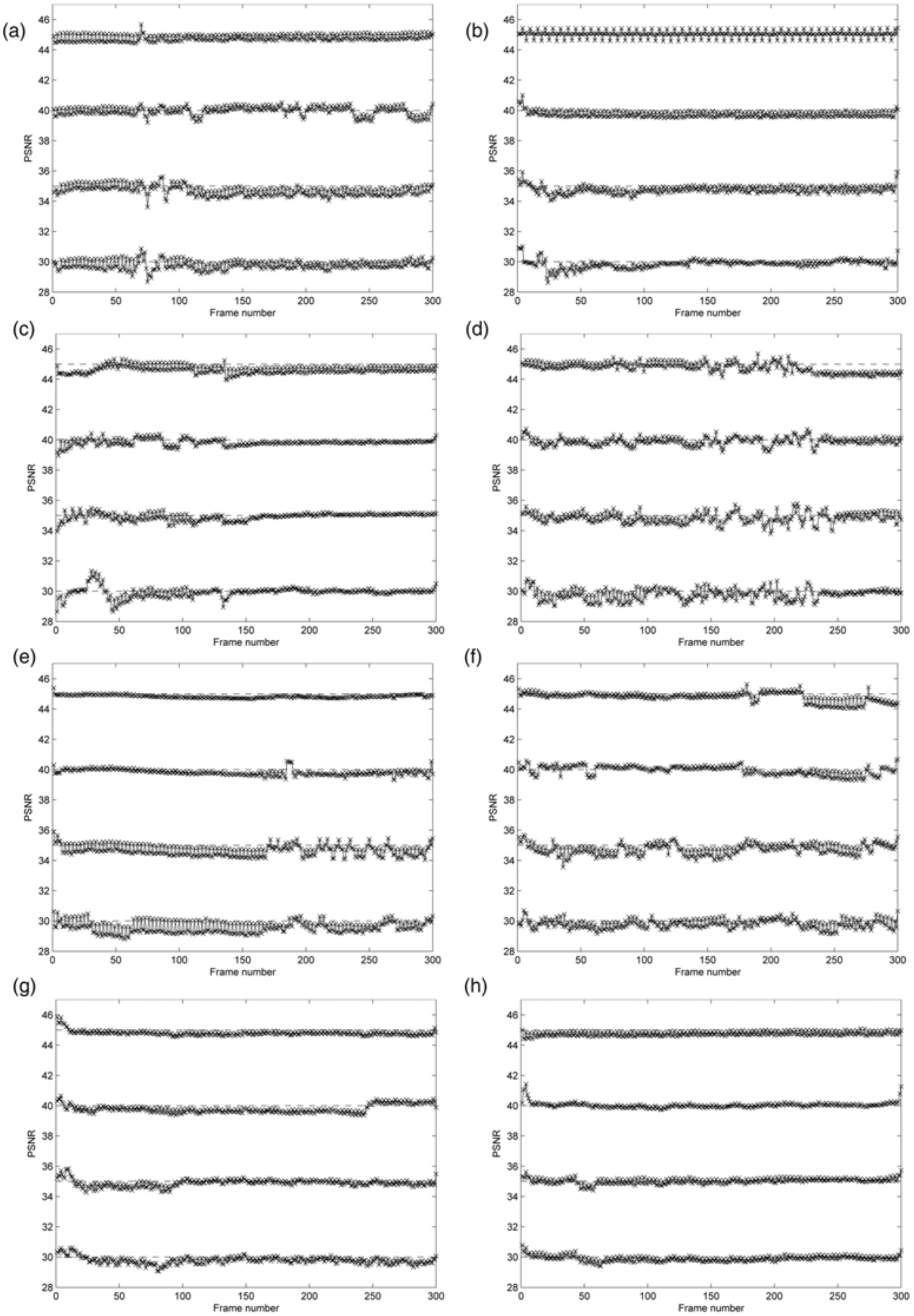

Fig. 12. The performances of constant quality (PSNR) video coding with B-frames (IBBPBBP. . .) of (a) Coastguard, (b) Monitor, (c) Table, (d) Foreman, (e) Mobile, (f) Stefan, (g) News, and (h) Paris.

SSIM verify the feasibility of our proposed method. We will extend them to test more quality metrics to further prove the generality of this framework.

ACKNOWLEDGEMENT

This research is supported by "The Aim for the Top University Project" of the National Central University and the Ministry of Education, Taiwan, R.O.C.

REFERENCES

[1] Wiegand, T.; Sullivan, G.; Bjntegaard, G.; Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, **13**(7) (2003), 560–576.

[2] Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, **13**(4) (2004), 600–612.

[3] Pinson, M.H.; Wolf, S.: A new standardized method for objectively measuring video quality. *IEEE Trans. Broadcast.*, **50**(3) (2004), 312–322.

[4] Liu, T.J.; Liu, K.H.; Liu, H.H.: Temporal information assisted video quality metric for multimedia. in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, July 2010, 697–701.

[5] Moorthy, A.K.; Bovik, A.C.: Efficient video quality assessment along temporal trajectories. *IEEE Trans. Circuits Syst. Video Technol.*, **20**(11) (2010), 1653–1658.

[6] Huang, Y.H.; Ou, T.S.; Su, P.Y.; Chen, H.H.: Perceptual rate-distortion optimization using structural similarity index as quality metric. *IEEE Trans. Circuits Syst. Video Technol.*, **20**(11) (2010), 1614–1624.

[7] Ou, T.S.; Huang, Y.H.; Chen, H.H.: SSIM-based perceptual rate control for video coding. *IEEE Trans. Circuits Syst. Video Technol.*, **21**(5) (2011), 682–691.

[8] Huang, K.L.; Hang, H.M.: Consistent picture quality control strategy for dependent video coding. *IEEE Trans. Circuits Syst. Video technol.*, **18**(5) (2009), 1004–1014.

[9] Ma, S.; Gao, W.; Lu, Y.: Rate-distortion analysis for H.264/AVC video coding and its application to rate control. *IEEE Trans. Circuits syst. Video Technol.*, **15**(12) (2005), 1533–1544.

[10] Kamaci, N.; Altunbasak, Y.; Mersereau, R.M.: Frame bit allocation for the H.264/AVC video coder via cauchy density-based rate and distortion models. *IEEE Trans. Circuits and systems for video technology*, **15**(8) (2005), 994–1006.

[11] Wu, C.Y.; Su, P.C.: A content-adaptive distortion-quantization model for intra coding in H.264/AVC, in *The 20th IEEE Int. Conf. Computer Communication Networks (ICCCN 2011)*, Maui, Hawaii, July 2011, 1–6.

[12] Vito, F.D.; Martin, J.C.D.: PSNR control for GOP-level constant quality in H.264 video coding, in *IEEE Int. Symp. Signal Processing and Information Technology (ISSPIT)*, Athens, Greece, December 2005, 612–617.

[13] Han, B.; Zhou, B.: VBR rate control for perceptually consistent video quality. *IEEE Trans. Consum. Electron.*, **54**(4) (2008), 1912–1919.

[14] Martin, D.; Fowlkes, C.; Tal, D.; Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *Proc. 8th Int. Conf. Computer Vision*, vol. 2, July 2001, 416–423.

[15] H.264/AVC JM reference software [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[16] Zhang, D.; Ngan, K.N.; Chen, Z.: A two-pass rate control algorithm for H.264/AVC high definition video coding. *Signal Process. Image Commun.*, **24**(5) (2009), 357–367.

Ching-Yu Wu received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Central University, Taiwan, in 2006 and 2008, respectively. Currently, he is working toward a Ph.D. degree at the same department. His research interests include video processing, video compression, and multimedia content analysis.

Po-Chyi Su was born in Taipei, Taiwan in 1973. He received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1995 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1998 and 2003, respectively, all in Electrical Engineering. He then joined the Industrial Technology Research Institute, Hsin-Chu, Taiwan, as an engineer. Since August 2004, he has been with the Department of Computer Science and Information Engineering, National Central University, Taiwan. He is now an Associate Professor. His research interests include multimedia security, visual surveillance, digital image/video processing, and compression.

Long-Wang Huang received the B.S. degree from the Department of Electronic Engineering, National Yunlin University of Science and Technology, Taiwan, in 2010, and the M.S. degree from the Department of Computer Science and Information Engineering, National Central University, Taiwan, in 2012. He is currently a software engineer in the Institute for Information Industry, Taipei, Taiwan. His research interests include multimedia compression and processing.

Chia-Yang Chiou was born on September 30, 1985. He received the B.S. degree from Aletheia University, New Taipei City, Taiwan, in 2009, and the M.S. degree from National Central University, Jhongli, Taiwan, in 2011, both in Computer Science and Information Engineering. He is now working as a Firmware Engineer in Etron Technology, Hsin-Chu, Taiwan. His research interests include image/video coding, processing, and firmware design.