Drift Compensated Coding Optimization for Fast Bit-rate Reduction Transcoding

Peng Zhang*[ac], Xiangyang Ji[a], Wen Gao[b], Qingming Huang[c]
[a] Institute of Computing Technology, Chinese Academy of Sciences;
[b] School of Electronics Engineering and Computer Science, Peking University
[c] Graduate School, Chinese Academy of Sciences

ABSTRACT

Bit rate adaptation is one of the basic problems in universal multimedia access. Therefore, in many cases, there exists a strong requirement for a very fast transcoding implementation. Usually, open-loop transcoding architecture is the desired solution with the lowest computational complexity. However, severe drift error makes it infeasible to most applications. In this paper, we propose a drift-compensated coding optimization scheme, by which generated bitstreams can be effectively transcoded into lower bitrate by open-loop transcoder with little drift propagation. The encoder integrates a virtual open-loop transcoder, in which drift error is effectively compensated by suitably adjusting the RD based mode selection and motion estimation for each macroblock. Simulation results show that compared with traditional coding, the proposed mode selection scheme can yield better coding efficiency when rate reduction transcoding to low bitrate occurs. And meanwhile, it does not degrade the coding efficiency in comparison with the normal single layer coding in H.264/AVC.

Keywords: transcoding, drift-compensation, coding optimization, H.264/AVC, open-loop

1. INTRODUCTION

There are various terminals including lap top computers, set-top boxes, and smart phones which may have different accesses to the media sources, including local access network (LAN), DSL, dial-up and wireless network. When transmitting a compressed video bit-stream over heterogeneous networks and to different terminals, it is often needed to dynamically adapt the coded video stream to meet the constraints of channel bandwidth or terminal processing capability. Transcoding is one of the key technologies to fulfill this challenging task, which converts a video from one format to another format. Bitrate downscaling of a pre-compressed video stream to a channel bandwidth is one of the basic transcoding problems [1~4]. For example, a video is compressed at a high bit rate for DVD applications, but later needs to be accessed by a mobile user at a much lower bitrate.

The open-loop architecture might be the fastest transcoding architecture which include selective transmission [1,2], where the high-frequency DCT coefficients are discarded, and re-quantization [3], where the DCT coefficients are re-quantized. The open-loop transcoders are computationally efficient, since they operate directly on the DCT coefficients. However, the reconstructed references in the transcoded stream are different from that used in the original stream. Such mismatches would bring severe drifting error. In order to reduce drifting error in video transcoding, the close-loop video transcoder was proposed [4]. By reusing motion vectors in the original stream, it can merge the decoding and encoding loops in the cascaded transcoder together. So the drifting errors can be accumulated in the prediction loop and used to compensate the re-quantization errors during the transcoding process. L. Yuan [5] proposed a hybrid transcoding scheme by take advantages of both open-loop and close-loop architectures.

As H.264/AVC [6] appears, things become different. Among all features adopted in H.264, variable block-size motion compensation, quarter-sample-accurate motion estimation, and multiple reference frames are most important as they play a dominant role in the coding efficiency enhancement as well as the computational complexity increase. Although these features improve the coding efficiency, they usually become obstacles for H.264/AVC transcoding. As the prediction mode and motion vectors are optimized for high bit rate, they may not be optimal at lower bit rate, the open-loop and close-loop transcoders which reuse these information will suffer an additional quality loss [7].

*peng.zhang@jdl.ac.cn; phone 86 10 5885 8300; fax 86 10 58858301;

As many applications still require a very fast transcoding implementation. Open-loop transcoding architecture is the fastest transcoding implementation and it will supply acceptable quality if drifting error can be controlled. As Lagrangian rate-distortion optimization (RDO) is used to find the optimal motion vectors and block prediction mode for each macroblock。We propose a drift-compensated coding optimization architecture, by which generated bitstreams can be transcoded into lower bitrate by open-loop transcoder with little drift propagation. The proposed encoder integrates a virtual open-loop transcoder, and keeps tracking the drift error at low bit rate, which can be compensated during the RD based mode selection and motion estimation of each block. Our simulation results show that compared with traditional transcoding, the drift-compensation in both mode decision and motion estimation can yield better coding efficiency when rate reduction transcoding to low bitrate occurs. And meanwhile, drift-compensation in mode decision does not degrade the coding efficiency in comparison with the normal single layer coding in H.264/AVC, while drift-compensation in motion estimation degrades the coding efficiency.

The rest of this paper is organized as follows. In section 2, we describe the proposed drift compensated methods in detail. In section 3, the simulated results are presented. Finally, section 4 concludes this paper.

2. DRIFT-COMPENSATED CODING OPTIMIZATION

In order to understand how the drifting error comes, we first review the architectures of the conventional cascaded decoder-encoder (Fig.1) and open-loop transcoder with re-quantization scheme (Fig.2), and discuss what drift error is.

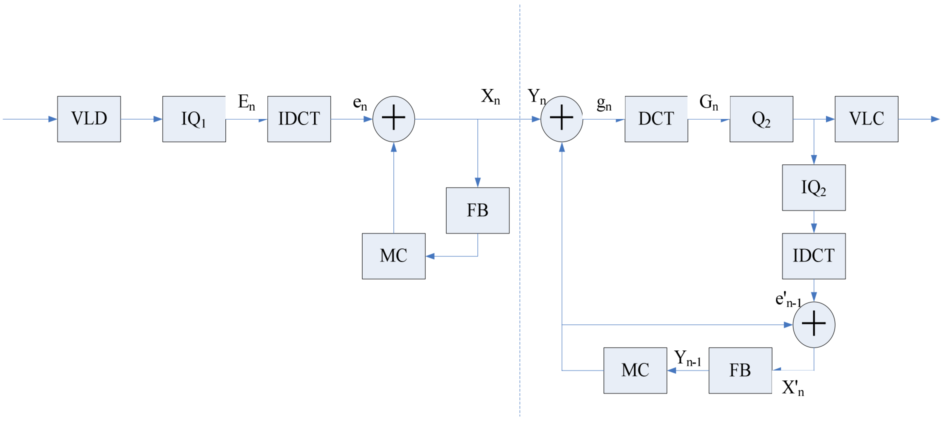


Fig. 1 cascade decoder-encoder

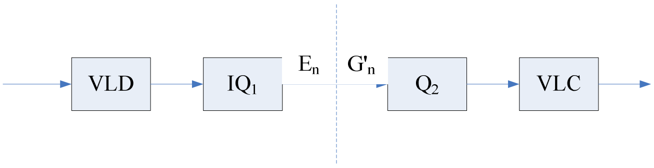


Fig. 2 open-loop transcoder

Fig.1 illustrates the framework of a cascaded decoder-encoder. In the decoder part, X_n denotes the decoded picture, E_n is the DCT coefficients after requantization, and e_n is the residual, while Y_n is the input picture of the encoder part, g_n and G_n are coefficients before and after DCT transform, respectively. M denotes motion compensation operation. In this figure, we can get:

$$X_n = e_n + M(X_{n-1}) \tag{1}$$

and

$$g_n = Y_n - M(Y_{n-1}) = e_n + M(X_{n-1}) - M(X'_{n-1}) \tag{2}$$

$$
\begin{aligned}
G_n &= \mathrm{DCT}(g_n) = DCT(e_n + M(X_{n-1}) - M(X'_{n-1})) \\
&= \mathrm{DCT}(e_n) + \mathrm{DCT}(M(X_{n-1}) - M(X'_{n-1})) \\
&= E_n + \mathrm{DCT}(\Delta\mathrm{MC}) \qquad (3)
\end{aligned}
$$

where

$$\Delta\mathrm{MC} = M(X_{n-1}) - M(X'_{n-1}) \qquad (4)$$

In Fig.2, the open-loop transcoder starts with the dequantization of the DCT coefficients using the original quantizer levels (Q1). These coefficients are re-encoded with a new target quantizer (Q2) for output bit rate reduction. We get

$$G'_n = E_n \qquad (5)$$

Compared (3) with (5), the drifting error of the *n*th frame, d_{drift} can be expressed as

$$d_{drift}(n) = G_n - G'_n = \mathrm{DCT}(\Delta\mathrm{MC}) \qquad (6)$$

d_{drift} represents an error in the reference picture that is used for motion compensation accumulated to current picture. This error may be caused by re-quantization, elimination of some nonzero DCT coefficients, or by integer truncation [3].

Open-loop transcoders are relatively simple since frame memory is not required and there is no need of inverse transform transform. However, severe drift propagation usually makes it unacceptable in real application. To eliminate drift propagation, [4] proposed a close-loop architecture, which contains a feed-back loop to compensate the drift. If the encoder could compensate most part of the drift, open-loop transcoder would work well in a wide range environment.

In this paper, we propose a drift-compensation coder system as illustrated in fig.3, in which a virtual open-loop transcoder is integrated, and keeps tracking the drift error at low bit rate.

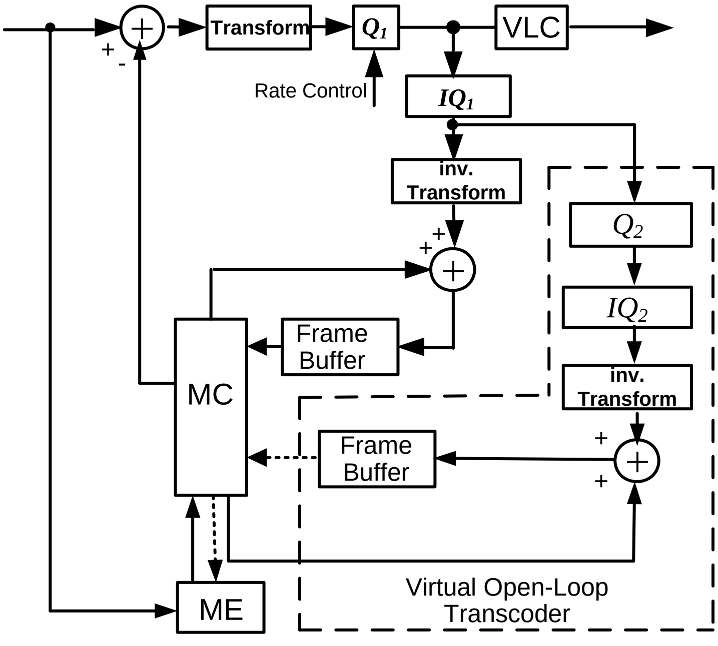


Fig. 3 Drift-compensated encoder system

As we know, the coding mode of a macroblock in H.264/AVC can vary from the set {INTRA4x4, INTRA16X16, INTER16x16, INTER16x8, INTER8x16, INTER8x8, INTER8x4, INTER4x8, INTER4x4, SKIP, DIRECT}. The

Lagrangian method is used to find the optimal motion vector for inter coded block and the optimal coding mode for any block. It provides high performance in solving the optimal bit allocation compromise between the motion vectors and the residual coding in the encoder.

For a inter block in an inter frame, the rate constraint motion estimation is first done to find the optimal motion vector by minimizing:

$$J(mv,\lambda_{MOTION}) = D(s,c(mv)) + \lambda_{MOTION} \times R(mv - pmv) \tag{7}$$

where mv is the estimated motion vector by motion estimation, pmv is the predicted motion vector, and λ_{MOTION} is the Lagrange multiplier. The rate term $R(mv - pmv)$ represents the motion information to be coded. $D(s,c(mv))$ is the sum of absolute difference (SAD) between the original picture s and the coded picture c.

Afterwards, the rate-constrained mode selection is performed to choose the optimal coding mode by minimizing:

$$J(s,c,MODE \mid QP,\lambda_{MODE}) = D(s,c,MODE \mid QP) + \lambda_{MODE} \times R(s,c,MODE \mid QP) \tag{8}$$

where the distortion $D(s,c,MODE \mid QP)$ is measured as the sum of squared errors between the original block s and the reconstructed block c, and QP is the quantization parameter. $R(s,c,MODE \mid QP)$ is the rate obtained after run-level variable-length coding.

2.1 Drift compensation in mode decision

First, we apply drift compensation in the mode decision procedure. Drift error of virtual transcoder is taken into account to get a drift resilient mode for each block. Thus, the overall distortion of the mth MB in the nth frame is represented by:

$$D(s,c,MODE \mid QP) = D_s(s,c,MODE \mid QP) + \theta \times D_{drift}(QP_{vt}) \tag{9}$$

where θ is the weight factor, $D_s(s,c,MODE \mid QP)$ and $D_{drift}(QP_{vt})$ denote the source and the drifting distortions, respectively, and QP_{vt} denotes the quantization parameter used in the virtual transcoder. Fig. 4 illustrates the proposed drift compensated RDO procedure. After the current frame is encoded, the reconstructed transcoded image is stored as the reference frame for the future frames.

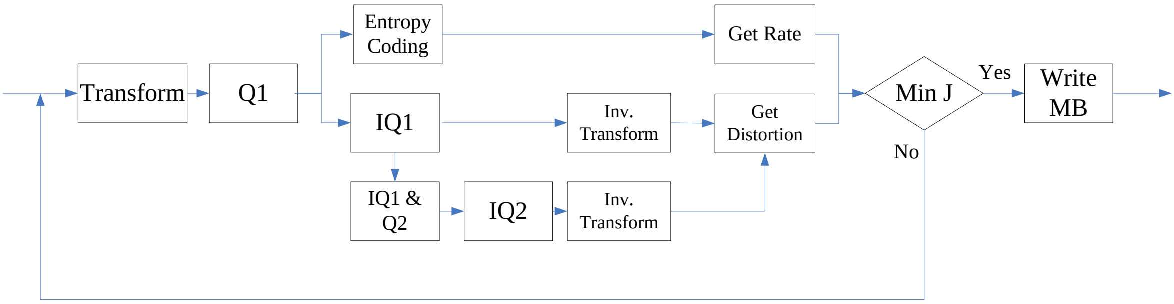


Fig. 4 Proposed drift compensated RDO procedure

2.2 Drift compensation in motion estimation

It is intuitively obvious that since motion vectors have a central role in the prediction loop, they are directly and critically involved in the drift error propagation mechanism encountered in open-loop transcoding. Hence, motion compensation may have a considerable impact on drift compensation.

In general, most reference models of existing video coding standards select the motion vector so as to minimize some measure of the prediction error. For example, the minimum sum of absolute difference (SAD) is given by

$$\min \sum_{i \in MB} | f_n^i - \hat{f}_n^i | = \min \sum_{i \in MB} |e_n^i| \tag{10}$$

where, f_n^i and $\hat{f}_n^i$ denote the original and encoder reconstruction values of pixel i in frame n, respectively. mv is searched over all the possible motion vector choices of a MB, and e_n^i is the prediction error.

In H.264/AVC, RD optimization of motion compensation was considered as shown in (2), in order to better allocate bitrate among motion vector and prediction residue coding, and thereby improve the overall coding efficiency.

Similarly, we extend the above RD approach to the case of drift-compensation. The only change in formulation is the sum of source-coding distortion D (SAD) and the drift error

$$D(s,c(mv)) = D_s(s,c(mv)) + \varphi \times D_{drift}(c(mv)) = \sum_{i \in MB} (|e_n^i| + \varphi * |d_{drift,n}^i|) \tag{11}$$

where, $d_{drift,n}^i$ and e_n^i denote the drift error and the prediction error of pixel i in frame n. φ is weighted factor.

3. EXPERIMENTAL RESULTS

Experiments have been carried out to verify the performance of the proposed algorithm. The testing platform is the H.264 reference software JM9.4. A cascaded decoder-encoder is taken as references in the comparison and the coded streams will then be transcoded by an open-loop transcoder. Because intra-prediction in H.264/AVC also has drift effect, Intra blocks are transcoded with close-loop architecture for simplicity, and inter blocks are transcoded with open-loop architecture.

In the experiments, we used three sequences, foreman, mobile, and news, which represents active, medium, and still motion, respectively. All sequences were coded as CIF, 30fps, QP24. As intra frame and B frame do not lead drift error, for simplicity, each GOP of coded sequences contains 30 frames, and no B frames in the coded sequences.

Fig. 5 illustrates transcoding results of stream (foreman) coded with the proposed drift compensation in mode decision with different θ value. And Fig. 6 illustrates transcoding results of stream (foreman) coded with the proposed drift compensation in motion estimation with different φ value. From the results, we can see that drift error can be effectively compensated by the proposed method both in mode decision and motion estimation procedure. And meanwhile, the larger θ and φ be, the better drift-error compensation achieved.

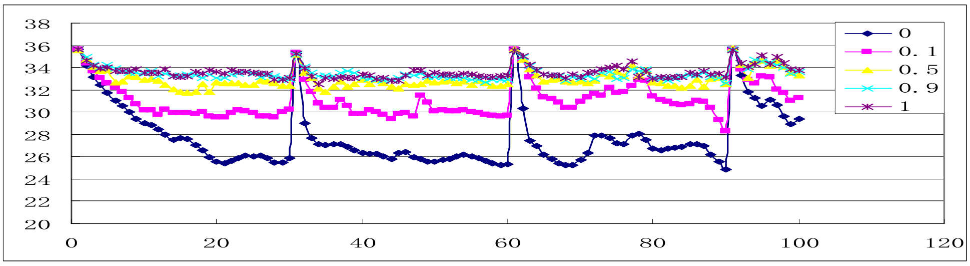


Fig. 5 Transcoding results with different θ value, foreman

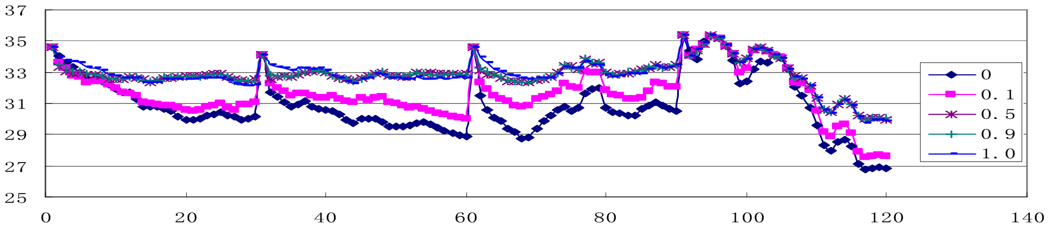


Fig. 6 Transcoding results with different φ value, foreman

Fig. 7 shows results of coding performance of drift compensation in mode decision and motion estimation, with θ and φ range from 0 to 1, respectively. From the results, we can see that drift compensation in mode decision performs much better than drift compensation in motion estimation. And a proper weighting factor θ or φ is the key to get a trade-off between coding efficiency and transcoding performances. Fig. 8 shows the results of the generated streams were transcoded to lower bit-rate. From the figure, we can see that drift error propagation is effectively eliminated. Compared with no drift-compensated streams, our proposed method performed superior to the JM. Fig. 9 illustrates R-D curves of the three sequences which are coded by the proposed method and JM encoder, respectively. From the results, we can see that the performance of our proposed method able to yield comparable coding efficiency with JM. Fig. 10 shows that when the drift-compensated streams are transcoded to different target rates over the assumed low bitrate point at encoder, there exist a linear relationship between PSNR and QP.

4. CONCLUSIONS

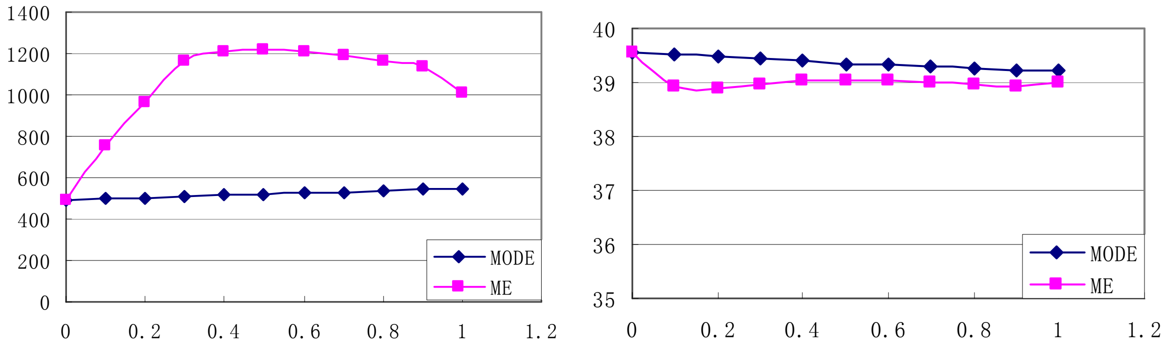
In this paper, we designed a drift-compensated coder, which compensates the possible drift-error propagation by integrating a virtual open-loop transcoder in the H.264/AVC encoder. Drift-compensation can be applied both in motion estimation procedure and mode decision procedure. Simulation results show that the proposed drift-compensation encoder optimization scheme improves the performance of open-loop transcoder while has little penalty on encoder performance.

5. ACKNOWLEDGEMENT

This work was supported in part by Beijing NSFC under contract No. 20040120.

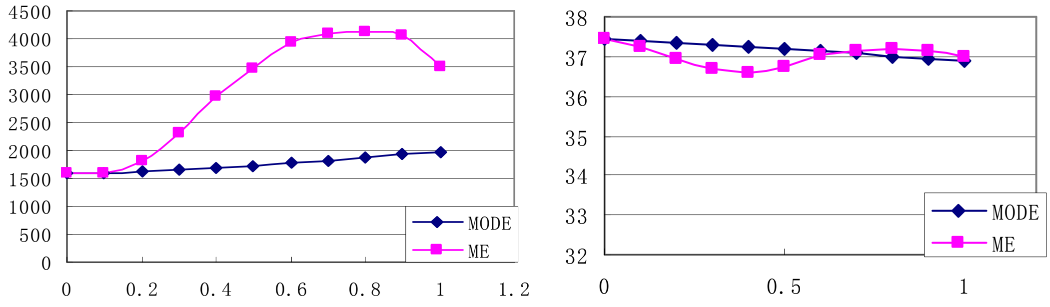
REFERENCES

1. Y. nakajima, H. Hori, and T. Kanoh, “Rate conversion of MPEG coded video by requantization process,” in Proc. IEEE Int. Conf. Image Processing, pp. 408-411, Washington, DC, 1995.
2. H. Sun, W. Kwok, and J. Zdepski, “Architectures for MPEG compressed bitstream scaling,” IEEE trans. Circuits Syst. Video Technol., vol. 6, pp. 191-199, Apr. 1996.
3. P. Assumcno and M. Ghanbari, “Post-processing of MPEG-2 coded video for transmission at lower bit-reates,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Atlanta, GA, 1996, pp. 1998-2001.
4. P. Assuncao and M. Ghanbari, “A frequency-domain video transcoder for dynamic bit rate reduction of MPEG-2 bit streams,” IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 8. pp. 953-967, Dec. 1998.
5. L. Yuan, F. Wu, Q. Chen, S. Li, W. Gao, ”The fast close-loop video transcoder with limited drifting error,” Proc. of the 2004 Int. Symposium on Circuits and Systems, vol. 3, pp. 23-26 May, 2004
6. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 2003.
7. P. Zhang, “key technology for Key Techniques of bitrate reduction transcoding for H264 streams,” PCM’2004



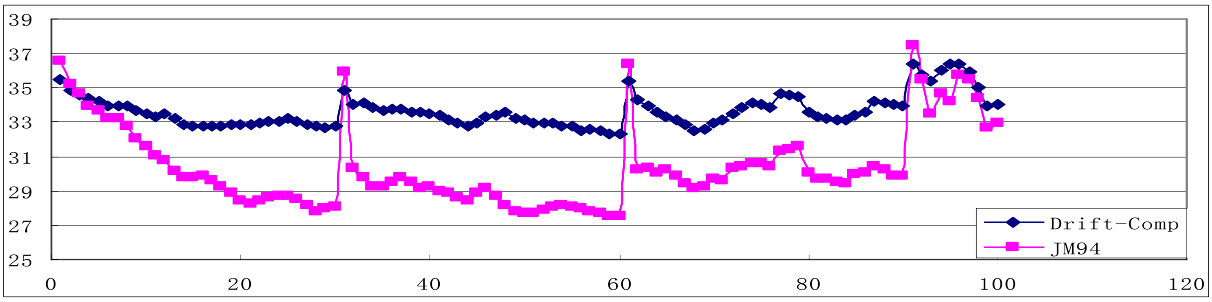
(A) RATE, FOREMAN

(B) PSNR, FOREMAN

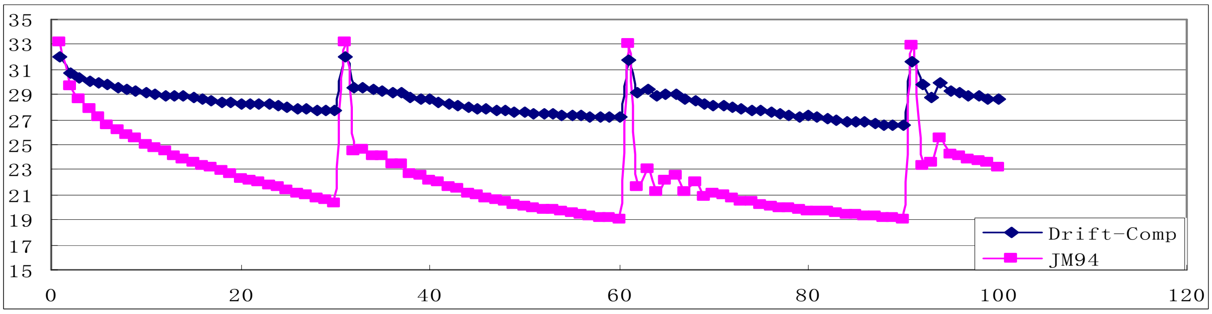


(C) RATE, MOBILE (D) PSNR, MOBILE

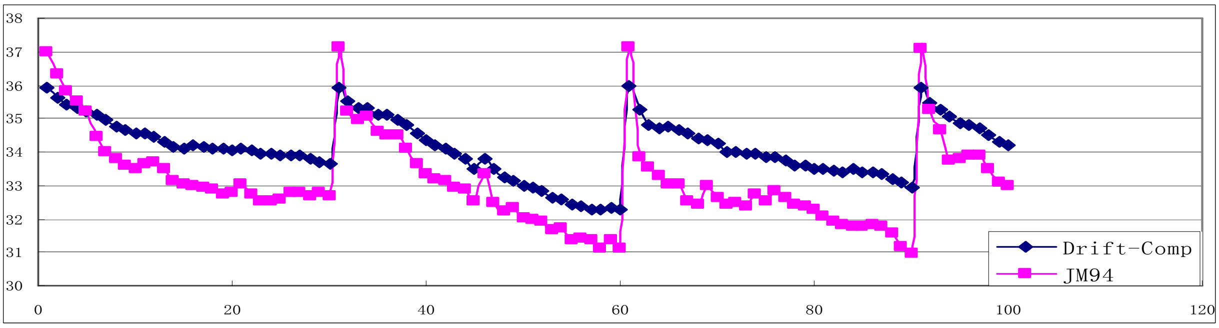
Fig. 7 Coding performance comparison of drift compensation by mode decision and motion estimation (θ and φ range from 0 to 1, respectively)



(A) foreman

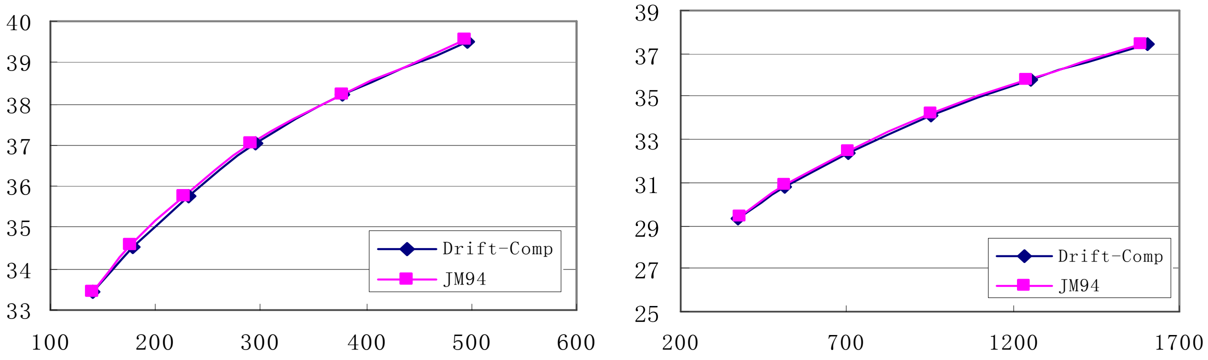


(B) Mobile



(C) News

Fig. 8 PSNR curve of of *foreman*, *mobile*, and *news* transcoded by open-loop transcoder.



(A) Foreman (B) Mobile

Fig. 9 R-D curve of *foreman and mobile* coded by proposed method, JM reference software, respectively.

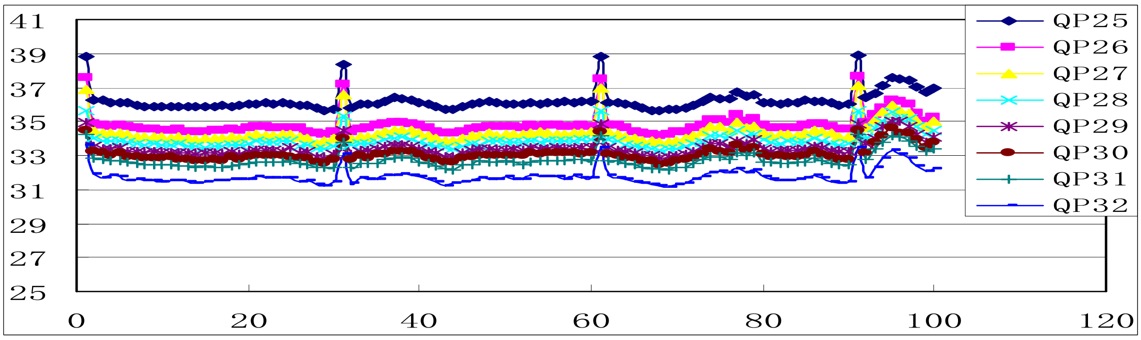


Fig. 10 Transcoding results with different target QP, *foreman*