A Novel Motion Vector Refinement Algorithm for Spatial Resolution Reduction Transcoding

Goo-Rak Kwon, June-Sok Lee, Jae-Won Kim, Nam-Hyeong Kim, and Sung-Jea Ko
Department of Electronics and Computer Engineering, Korea University, Sungbuk-Ku, Seoul, 136-701, Korea
Tel.: +82-2-3290-3672, Fax.: +82-2-925-5883
E-mail : sjko@dali.korea.ac.kr

Abstract: For the low bit-rate video communications, video contents have to be transcoded according to network bandwidth and the performance of clients. The heterogeneous low bit-rate transcoding, however, suffers from quality degradation caused by the drift error. In this paper, we propose a novel motion estimation refinement algorithm to compensate drift errors. Simulation results show that the proposed algorithm not only preserves original image quality, but also reduces the computational complexity of spatial resolution reduction transcoding.

1. Introduction

With the rapid advance of multimedia technologies, multimedia services, such as video conferencing, video-and-demand (VOD) have become increasingly more common. In these applications, it is often needed to adapt the bit-rate of a coded video bit-stream to the available bandwidth over heterogeneous network environment [1]. Previous research on spatial resolution reduction (SRR) transcoding has been focused on transcoding to produce a low bit-rate (LBR) video stream. In earlier works, bit-rate reduction techniques were explored to meet available channel capacity [2]. This technique was refined in [3] to reduce the complexity of the transcoding architecture. Another transcoding scheme for LBR video is a reduced temporal resolution transcoding. It is used to adapt the bit-rate requirements of a network by using frame skipping method [4], [5]. LBR video transcoding can be also achieved by reducing the spatial resolution of a video [6], [7].

Throughout this article, we concentrate on the solution for the drift error which is the major problem of SRR transcoding. The drift error occurs when the reconstructed pictures in the encoder and the decoder are not exactly the same. Since the current reconstructed picture is used for the future prediction, the drift error is propagated to the future frames and the video quality decreases. The drift error can be minimized by the I-frame insertion or intra macroblock (MB) refresh methods [7]. Although these methods remove drift error successfully, they are not proper to the LBR video transcoding since intra MB and I-frame increase the bit-rate of the encoded bitstream. In [6], an adaptive motion vector (MV) resampling method is proposed to compensate the drift error. The MV mapping method is a useful technique for LBR video transcoding to enhance picture quality without bit-rate increasing. However, the inaccurate resampled MVs lead to the degradation of transcoded picture quality as well as prediction errors causing higher transmission bit rates. In [8], [9], several MV refinement (MVR) methods are proposed to improve video quality by re-estimating MV. However, MVR methods increase the computational complexity and time.

In this paper, we propose a novel MVR algorithm to compensate drift errors with reduced computational complexity for the SRR transcoding.

The organization of this paper as follow. In section 2, we review the proposed MVR algorithm is described in detail. In section 3, the experimental results are discussed. Finally, section 4 concludes this paper.

2. The Proposed Motion Vector Refinement Algorithm

The proposed MVR consists of the MV mapping (MVM) and MVR processes as shown in Fig. 1. In the MVM process, the base MV (BMV) is obtained by selecting the downscaled MV from the four MVs of the original four MBs. We propose a new MVM method, called a twofold MV mapping (TMVM), to find precise BMV which minimizes the prediction error of an MB. In the MVR process, the BMV is refined within small search range by using the motion estimation. In the following subsections, these processes are explained in detail.

2.1 Proposed motion vector mapping

In the case of SRR transcoding, the MVs extracted from the incoming bitstream need to be downscaled to form a BMV as shown in Fig. 1. The BMV can be obtained by using the MVM method which reuses the existing motion information in original video sequence to avoid the computational expenses. However, the inaccurate BMV may result in unacceptable picture quality. In addition, the bit-rate of the transcoded stream increases with the variance of prediction error caused by the inaccurate BMV. Therefore, an effective MVM algorithm is required to find a precise BMV.
Before introducing the proposed method, we briefly review some existing MVM methods. The simplest method is to average the original MVs in the four MBs:

\[ \check{v} = \frac{1}{2N} \sum_{i=1}^{N} v_i, \]  

where \( v_i \) is the MV of \( i^{th} \) MB in original video, \( \check{v} \) denotes the BMV, and \( N \) is the total number of MVs in four MBs in original video. The median vector is defined as the MV with the least distance from all MVs. This method calculates the sum of the distance \( d_i \) between each vector and its neighbors as follows:

\[ d_i = \sum_{j=1,j\neq i}^{4} \| v_i - v_j \|, \]  

then the \( v_i \) at which \( d_i \) has minimum value is selected as the BMV. These methods, however, can not be applicable for the video sequences such as MPEG-2, which provides very rich MB types and motion types to achieve high coding efficiency, because they do not consider any information about MB and MV types. Specifically, the conversion of MB and MV types is a prominent problem at transcoding from the MPEG-2 to other coding formats such as MPEG-4 or H.263, since H.263 and MPEG-4@SP (with simple profile) have relatively simple MB types and motion types than MPEG-2.

The proposed TMVM method selects the BMV that minimizes the prediction errors is selected among the multiple MVs in each MB. The representative MV is obtained by

\[ \check{V}^k = \arg \min_{i \in S} \frac{1}{N} \sum_{j=0}^{N-1} |P^k_j - \hat{R}^k_j(V_i)|^2, \]

where \( \check{V}^k \) is the representative MV of \( k^{th} \) MB, \( S \) is the set of MV candidates, \( V_i \) is \( i^{th} \) MV in \( S \), and \( N \) is the number of pixels in the MB. The \( P^k_j \) is the pixel in the current frame and \( \hat{R}^k_j(V_i) \) is the pixel compensated with \( V_i \) in the reference frame.

Table 1 shows the MV candidates for the different MB and MV types, where \( V \) represents the MV of each MB. In the MB with field MV type, the representative MV is selected from the candidate set \( S = \{ \check{V}_0, \check{V}_T, \check{V}_B, \check{V}_{avg} \} \), where \( \check{V}_0 \) is zero MV, \( \check{V}_T \) is the top-field MV, \( \check{V}_B \) is the bottom-field MV, and \( \check{V}_{avg} \) is the average of all the candidate MVs in the set. In the MB with bidirectional prediction type, the representative MV is selected from \( S = \{ \check{V}_0, \check{V}_f, \check{V}_b, \check{V}_{avg} \} \), where \( \check{V}_f \) and \( \check{V}_b \) represent the forward and backward MV, respectively.

For the MB with backward type, the MVs of the MB are reversed and scaled to obtain a forward MV, and then the representative MV is selected from the candidates. The zero MV is used for the intra-coded and skipped MB regardless of MV type.

After selecting the representative MV for each MB in the original frame, the TMVM is performed to find the BMV in the downsized frame as follows:

\[ \check{v}^k = \arg \min_{i \in S_L} \frac{1}{N} \sum_{j=0}^{N-1} |P^k_j - \hat{R}^k_j(\check{V}_i/2)|^2, \]  

where \( \check{v}^k \) is the BMV of \( k^{th} \) MB, \( S_L \) is the set of candidates, \( \check{V}_i \) is the representative MV obtained in the original image, and \( N \) is the number of pixels in the MB. Note that \( \check{V}_i/2 \) is used for the prediction process since \( \check{P} \) and \( \check{R} \) are the down-sized current and reference pictures, respectively. \( \hat{R}^k_j \) is the pixel of the \( k^{th} \) MB in the current frame and \( \hat{R}^k_j(\check{V}_i/2) \) represents the pixel predicted with \( \check{V}_i/2 \) in the reference frame.

### 2.2 The novel motion vector refinement algorithm

Most of the MVR methods re-estimate the MV by using the small search range and fast search algorithm to reduce the computations. However, there are still heavy computational complexity since the all MVs in the frame are re-estimated in the refinement process.

In the proposed MV re-estimation method, the MVR is performed only for some MBs with large pixel variance in a frame instead of refining all MVs in the frame. We use the following decision algorithm to determine whether or not the BMV of an MB needs to be refined.

\[ \text{if } \sigma^2_i > \sigma^2_t \]

(\text{refine the BMV}_i)

\[ \text{else} \]

(\text{no MV refinement}),

where \( \sigma^2_i \) is the pixel variance of \( i^{th} \) MB, \( \text{BMV}_i \) is the base MV of \( i^{th} \) MB, and \( \sigma^2_t \) is the averaged variance of all MBs in the frame consisting of \( N \) MBs that is given by

\[ \sigma^2_t = \frac{1}{N} \sum_{i=0}^{N-1} \sigma^2_i. \]
When the MB is predicted with the BMV, the MB with larger variance can produce larger distortion than the other MBs. Therefore, the BMV with $\sigma_i^2 > \sigma_f^2$ needs to be refined to enhance the picture quality while reducing the transcoded bit-rate.

For $\sigma_i^2 > \sigma_f^2$, the search range for re-estimation is adaptively calculated as follows:

$$R = r\left(\frac{\sigma_i^2}{\sigma_f^2} - 1\right), \quad (1 \leq r \leq 2, \quad R \leq 4),$$

where $R$ is the calculated search range, $r$ is the scaling factor for the search range, and $\sigma_i^2/\sigma_f^2$ is the distortion rate of the $i^{th}$ MB. Note that the $R$ increases with the scaling factor because the wider search range is required for the MB with larger distortion. Since the MVPM process provides precise BMV for the MVR process, the $R$ is restricted as $R \leq 4$. Thus, the search window size is still small and the computational complexity is much less than the full-scale motion estimation.

Finally, the proposed MV re-estimation method refines the BMV obtained by the TMVM method within the calculated search range.

### 3. Experimental Results

The SRR transcoding from MPEG-2 to H.263 is conducted to evaluate the performance of the proposed MVR method. The test sequence “Foreman” with CIF resolution ($352 \times 288$) is encoded to the MPEG-2 bitstreams as 1.15 M bits/s with 12 frames in GOP ($N = 12$) and 3 frames in sub group ($M = 3$). In the transcoding process, the intra frames are downsized by 2 and encoded to H.263 with the intra mode. The decoded inter frames (P- and B-frames) are downsized in the spatial domain and the prediction is performed with the MV obtained by the MVR. Then, the prediction residues are transformed, quantized, and encoded into a H.263 bitstream. Fig. 2 shows the SRR transcoding architecture using the proposed MVR method.

Before investigating the performance of the MVR, the performance of the proposed MV resampling method is compared with the two MV resampling methods, the average and the median, which were introduced in Section 2.1. Fig. 3 shows the transcoded picture quality with regard to each MV resampling method. The periodic peaks represent the peak signal to noise ratio (PSNR) of I-frames. The PSNR values of the average and the median methods decrease until the I-frame is presented, since the prediction is performed with the inaccurate MVs. On the other hand, the proposed method achieves higher picture quality than the other methods. Note that the PSNR values of the proposed MV resampling method fluctuate with small variance, whereas the values of the average and the median methods are decreasing continuously until the I-frame is refreshed. Therefore, the proposed MV resampling method can provide more precise base MV for the MVR than the other methods.

<table>
<thead>
<tr>
<th>Frame number</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
</tr>
<tr>
<td>TMVM</td>
<td></td>
</tr>
</tbody>
</table>

Table 2 illustrates the picture quality of each MV refinement method according to the different transcoding bit-rates. The PSNR values of 300 frames are averaged to evaluate the performance of the MV refinement methods. For the average and median MV resampling methods, the full-scale motion estimation with small search range is applied as the MV refinement method. The “MVR(2)” in Table 2 means that the MV refinement is performed with a fixed search range of $\pm 2$. As shown in Table 2, the MVR generates higher picture quality than the other methods on every transcoding bit-rate. Specifically, the transcoded picture quality of the MVR increase significantly along with the increasing target bit-rate, whereas the median+MVR(2) and the average+MVR(2) methods do not show impressive quality enhancement.
Table 2. PSNR results with respect to the different bit-rate (Unit: dB)

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>Average+MVR(2)</th>
<th>Median+MVR(2)</th>
<th>MVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>64K b/s</td>
<td>23.92</td>
<td>24.31</td>
<td>26.56</td>
</tr>
<tr>
<td>96K b/s</td>
<td>23.93</td>
<td>24.41</td>
<td>27.34</td>
</tr>
<tr>
<td>128K b/s</td>
<td>23.97</td>
<td>24.80</td>
<td>28.11</td>
</tr>
</tbody>
</table>

Table 3. Transcoding time ratio according to the MV refinement methods (Unit: second)

<table>
<thead>
<tr>
<th>Transcoding time</th>
<th>Average+MVR(2)</th>
<th>Median+MVR(2)</th>
<th>MVR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.25</td>
<td>2.10</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3 shows the transcoding time ratio of each MV resampling method. The transcoding time is measured by using the Pentium-4 2.4Ghz computer. For easy comparison, the processing times of average+MVR(2) and median+MVR(2) are divided by the processing time of the MVR. This result shows that the MVR method significantly reduces the computational complexity for the transcoder, since the proposed method only re-estimates the small number of MVs and calculates search range dynamically according to the MB status.

4. Conclusions

In this paper, we proposed an selective MV Refinement algorithm for the LBR transcoder. The proposed method consists of MVM and MVR process. For the MVM process, we proposed TMVM method to make a precise BMV. Starting with the BMV, the proposed MVR method searches for the closer MV to the optimal MV within the small search range that is adaptively changed according to the MB statistics. By using the proposed MV mapping method and the adaptive calculation of the search range, the picture quality of the transcoded image be significantly improved. Moreover, to reduce the computation, the proposed MV re-estimation method selects some MBs with large pixel variance in the frame. This scheme significantly reduces the computational complexity of the transcoder because the refinement process is not applied to the all the MBs in the frame.

Experimental results show that the proposed algorithm not only preserves original image quality, but also reduces the computational complexity of SRR transcoding.

References