ABSTRACT
We propose a region-of-interest (ROI) coding framework based on leaky prediction (LP) for robustly transporting H.264 video over error-prone network. The LP-based ROI coding can remove error drift caused by the destruction in the decoded background from the proposed system, with guaranteed ROI quality throughout. Experimental results show that this scheme enables the decoder to reconstruct the ROI with better quality and the global video frame with improved quality even if the background bitstream cannot be correctly received or completely lost.

Index Terms—leaky prediction, region-of-interest, error resilience, H.264/AVC

1. INTRODUCTION
Video coding applications over the Internet and wireless networks have gained significant interest. In such an environment, it is very important to make the coder adaptive to varying network throughputs to obtain good visual quality with the available rate resources [1]. An enable approach to cope efficiently with this challenge is the layered or scalable coding scheme. Region-of-interest (ROI) scalability is of crucial interest in application scenarios where some visual regions are more important or interesting than the other parts of the video image. Although H.264/AVC is finalized without support of scalable coding except for temporal scalability with hierarchical B frames [2], ROI scalable coding can be easily implemented by using flexible macroblock order (FMO) [3]. Fig.1 illustrates how ROI scalable coding can be achieved in H.264/AVC. ROI is first defined in a video frame to divide the frame into ROI area and non-ROI area, and then each of the area is mapped to a distinct slice. Here ROI area is mapped to slice 0 as ROI layer, while non-ROI mapped to slice 1 as background layer. The two slices can be treated differently in transportation [4]. Since ROI layer is prior to background layer, it can be strongly protected by unequal error protection or using a reliable transport mechanism. However, the receipt of the background layer data packets is not guaranteed because they may be discarded partially or completely due to bandwidth variation.

For ROI scalable stream transmission over error-prone network, there is no guarantee that the ROI layer can be correctly decoded. The reason is that due to transmission error, the background layer at the decoder may mismatch for that at the encoder in motion-compensation (MC) loop, and may be employed as reference in decoding the current ROI layer. As shown in Fig. 2, when the ordered stream carrying the update information of background layer for \( f(n-2) \) is truncated, the reconstruction of background layer in \( f(n-1) \) will be falsified (marked by shaded area). Since \( f(n) \) might refer to background layer of \( f(n-1) \) in both ROI and background layer decoding, the errors spread in the decoded \( f(n) \) (marked by shaded area in both layers). Even if all the following bitstream can be correctly received, the errors will propagate spatially and temporally to the following frames \( f(n+1), f(n+2)\cdots \) until the next intra-coded frame is received.

To refrain background errors propagation, error concealment techniques[5]-[6] can be employed to hide visible distortion. However, the error remaining after concealment propagate to successive frames and remain visible for a longer period of time, which makes the resulting artifacts particularly annoying[7]. Another possible method to avoid background
error propagation is to constrain the motion search into ROI area, preventing inter-frame dependency between ROI and background layers. Despite the coding robustness of constrained motion estimation (ME), it is scarcely used due to the inefficiency.

In this paper, we propose a novel approach for ROI scalable coding framework by introducing leaky factor in MC loop, so that any error propagated from non-ROI area to the ROI is damped. The proposed scheme has two options for the prediction loop: the whole frame and the ROI layer. It partially employs a reference block from a constrained ME in ROI layer, and partially from unrestricted ME in the whole reference. A leaky factor is deployed to weigh the two predictions to trade off the two reference blocks. If leaky factor \( \alpha = 1 \), the ROI layer is completely excluded from MC loop, yielding to the best efficiency but the worst error resilience performance. When \( \alpha = 0 \), it restricts the ME range into ROI area only and thus results in poor prediction with strong robustness.

The rest of the paper is organized as follows: Section 2 describes the proposed ROI scalable coding with leaky prediction. In Section 3 we analyze the principle of error mitigation in ROI layer with the proposed scheme. Section 4 gives the simulation results with analysis, which demonstrate the robustness of the proposed scheme. And Section 5 concludes the paper.

2. ROI SCALABLE CODING FRAMEWORK WITH LEAKY PREDICTION

2.1. Encoder Structure

Leaky prediction is a well-known technique to increase error robustness by balancing coding efficiency and robustness [8]. Since the leakage introduced by spatial filtering in a motion compensated predictor is not strong enough for error resilience [9] in ROI region, which is considered much more important than the rest of the frame, we apply leaky factor \( \alpha \) (\( 0 < \alpha < 1 \)) to scale the whole reference employed by ROI layer to accelerate the error degradation. The encoder framework of the proposed scheme is shown in Fig. 3, by modifying the hybrid block-based coding structure in H.264/AVC. The difference lies in ME and MC loop. For a macroblock (MB) in ROI slice, there are two prediction reference sources: the whole reference frame and the ROI layer. An unrestricted ME is done in the whole frame to obtain the optimal prediction block. At the same time it conducts constrained ME within the decoded ROI layer to get the second prediction block. The two blocks are then scaled by \( \alpha \) and \( 1 - \alpha \) respectively, so as to generate a weighted sum as the final prediction for the coding block.

The prediction of ROI layer MB in \( f(n) \) denoted \( f_{ROI}^{(p)}(n) \) is formulated as:

\[
f_{ROI}^{(p)}(n) = \alpha MC_1(f^{(r)}) + (1 - \alpha) MC_2(f_{ROI}^{(r)})
\]

\( f^{(r)} \) is the prediction block from the whole frame reference, \( f_{ROI}^{(r)} \) is the prediction from ROI layer reference, \( MC_1(\cdot) \) denotes block motion compensation. It is worth noting \( MC_1 \) and \( MC_2 \) deploy two distinct motion vectors, \( MV1 \) and \( MV2 \) respectively, and both motion vectors have to be sent to the decoder.

Basically, we try to put higher priority to the decoded ROI layer quality over background layer than the background. In this paper, no leaky prediction is used in coding background slice, i.e. only the usual coding framework with unrestricted ME in the whole reference frame is carried out for background slice coding. Accordingly, the error mitigation speed in background layer is slower than that in ROI layer due to the lack of extra leakage. Of course, both ROI and background slices can be coded with leaky prediction. In this sense, the encoder framework in Fig. 3 is not restricted to ROI slice use, hence we do not distinguish the coding scheme utilized by the two slices in the proposed scheme.

2.2. Decoder Structure

The leaky prediction decoder structure, shown in Fig. 4 employs both the decoded overall frame and ROI layer in the MC loop. The decoder receives two different sets of side information. The first set of block mode with \( MV1 \) gets prediction from the whole frame, and the second mode with \( MV2 \) is to obtain the prediction from the ROI layer. Then the same gain factors \( \alpha \) and \( 1 - \alpha \) are introduced to scale the two predictions respectively to get a combined prediction block for current decoded MB. Both ROI slice and background slice MB can be decoded by this decoder if the MB is coded with leaky prediction at the encoder. For those MB not coded with leaky prediction, the conventional MC loop without ROI layer reference is utilized in decoding instead.

3. ERROR MITIGATION WITH LEAKY PREDICTION

In ROI scalable video coding in H.264, the mismatch of background layer causes error to spread spatially across ROI layer, yielding annoying artifacts in ROI. A factor \( \alpha \) less than 1 can severely attenuate the erroneous prediction signal from background, to mitigate the effect of the error in the decoded
following frames, while still preserving efficiency. In this section, we analyze the error robustness of ROI scalability with leaky prediction. And the efficiency problem will be discussed in Section 4.

For a clear formulation of error propagation with leaky prediction, we simplify the ROI scalable coding by the following three assumptions:

1) Particular care is only given to quality of the decoded ROI layer, while error propagation in the non-ROI region can be tolerated. Thus only macroblocks in ROI slice deploy leaky prediction.

2) The erroneous case is specialized so that the background layer of only one frame is influenced by transmission error. Suppose it is \( f(n-1) \), and the bitstream for frames that follow \( f(n-1) \) is spared from error.

3) Single reference is used, and each frame only predicts from its previous one frame.

Suppose \( \Delta \) is the error in the background layer of \( f(n-1) \). For each MB in ROI slice of \( f(n) \), the prediction is

\[
 f_{ROI}^{(p)}(n) = \alpha MC_1(f^{(r)}(n-1) + \Delta) + (1 - \alpha) MC_2(f_{ROI}^{(r)}(n-1))
\]  

(2)

The optimal prediction \( f^{(r)}(n-1) \) obtained by the unrestricted ME can be either inside, partially inside or entirely outside the previous ROI layer. Here we assume the worst case, in which the optimal prediction is entirely in the non-ROI area, i.e.

\[
 f^{(r)}(n-1) = f_{non-ROI}^{(r)}(n-1)
\]  

(3)

Therefore

\[
 f_{ROI}^{(p)}(n) = \alpha MC_1(f_{non-ROI}^{(r)}(n-1) + \Delta) + (1 - \alpha) MC_2(f_{ROI}^{(r)}(n-1)) = \alpha MC_1(f_{non-ROI}^{(r)}(n-1)) + \alpha \Delta + (1 - \alpha) MC_2(f_{ROI}^{(r)}(n-1))
\]  

(4)

For simplicity, 4 can be rewritten as

\[
 i(n) = \alpha o(n-1) + e(n) + (1 - \alpha)i(n-1)
\]  

(5)

where

\[
 i(n) = f_{ROI}^{(p)}(n) \quad o(n) = MC_1(f_{non-ROI}^{(r)}(n))
\]

As an iterative expression of \( i(n) \) and \( i(n-1) \), and \( e(n) \) is the ROI layer error in \( f(n) \), here \( e(n) = \alpha \Delta \). Since the leakage introduced by 1/4-pel accuracy in the MC loop, the transmission error in non-ROI area mitigates with time step quantized by \( \delta(t) \) according to the analytical model in [7], where \( \delta(0) = 1 \) and \( 0 < \delta(t) < 1 \) when \( t > 0 \). Therefore, the background error that spreads to ROI layer in \( f(n+1) \) can be expressed as
\[ e(n + 1) = \alpha \Delta \delta(1) + (1 - \alpha)\alpha \Delta \]  
\[ e(n + t) = \alpha \Delta \delta(t) + (1 - \alpha)\alpha \Delta \delta(t - 1) + \cdots + (1 - \alpha)(t-2)\alpha \Delta \delta(1) + \cdots \]

Clearly,
\[ \lim_{t \to \infty} \Delta (1 - \alpha)^t \to 0 \]  

Hence the error introduced from background decays over time and finally converges to 0 in the ROI layer of the video sequence, faster than that in the background layer which decreases with \( \delta(t) \). Though in true application, continuous errors in a bistream is more likely instead of a single frame error, the extra leakage proposed in the scheme also attributes to better error mitigation performance, as is demonstrated in Section 4.1.

4. SIMULATION RESULTS AND ANALYSIS

We present experimental results in this section to illustrate the robustness and efficiency of the proposed ROI scalable coding with leaky prediction by incorporating the proposed algorithm into H.264/AVC software H.264/AVC software JM9.8[10]. Packet loss simulation is conducted to compare the error resilience of the proposed scheme with that without the leaky prediction. The parameters related to the proposed scheme is investigated to analyze their respective influence on the performance.

4.1. Overall Performance of Leaky Prediction in ROI scalable Coding

Based on the theoretical analysis in Section3, the error from background impairment decays over time and can finally converge to 0 in the ROI layer of the video sequence, if the three assumptions have been satisfied. In Fig.5, we use the test sequence foreman (QCIF, 12.5Hz, 100 frames, QP=28, IPPP mode, single ROI) in the simulation. In this paper, we simulate a simple scenario that the background slice packets of the first P frame is discarded completely during transmission. We replace the corrupted image content by the corresponding pixels from previous frame as a simple approach for error concealment, which yields good results for sequences with little motion[6]. It can be clearly observed from Fig5 that loss \( \Delta \text{PSNR} \) in ROI layer gradually decays to near 0 with time in the proposed scheme, while there is no evident \( \Delta \text{PSNR} \) increase if no leaky prediction is employed. In addition, as shown in Fig. Fig5, the proposed scheme recovers the PSNR faster, since better ROI layer quality has upgraded the overall frame quality by the proposed scheme, though the background slice is coded without leaky prediction.

In Fig.6, continuous errors occur every 50 frames in the background slice transmission in 250 framesforeman. And the proposed scheme achieves gains, up to more than 2dB, in both ROI layer and overall picture compared to that without leaky prediction.

Since the leaky factor aims to get a tradeoff between robustness and efficiency, it would inevitably introduces losses in coding efficiency in error-free case due to two reasons:

1) The coded MB referring to the previous ROI layer is not always likely to find a good match in the constrained ME, and this would cause larger residue, thus leading
2) It is required to transmit two sets of block mode and motion vectors for each MB within ROI (coded with leaky prediction). And additional bitrate has to be consumed for the extra set of the side information.

Fig.7 shows the RD curve of the proposed scheme, in both low bitrate case in foreman QCIF (Fig.7(a)) and high bitrate application in stefan CIF (Fig.7(b)). On average, 1dB PSNR loss in the whole frame and the ROI is observed in slow motion sequence foreman, but the PSNR gap is narrowed with bit rate increase. Compared to the enhanced robustness in error-prone environments, the small loss in coding efficiency can be accepted. While in stefan sequence that featured fast motion, it is more difficult to find the match block by the constrained ME in ROI than in slow motion video, making the RD performance even worse, up to 2dB PSNR loss in error-free case.

4.2. Influence of Parameters on Leaky Prediction Performance

Leaky factor represents the evolvement of the ROI layer reference in MC, hence it serves as a parameter to adjust the importance between coding efficiency and robustness. In this experiment, we fix the quantization steps QP and vary the leaky factor $\alpha$. We use 250 frames foreman with continuous background error every 50 frames, and only ROI slice is code with leaky prediction. It is seen in Fig.8 that the error resilience performance of the reconstructed ROI layer and overall frame is closely related to the leaky factor. Generally, the smaller the leaky factor, the faster the errors decay, at a greater cost of bitrate, as illustrated in Tab.1.

The number of MB using leaky prediction in ROI scalable coding has a considerable effect on the proposed scheme. As mentioned in Section.2.1, leaky prediction is not restricted to ROI slice coding. Here we introduce a parameter $BK$ to indicate the size of background slice involved in leaky prediction. Specifically, $BK$ is the size of ROI neighboring area, e.g. the extended 1 MB area encircling ROI would use leaky prediction if $BK = 1$. Again, we fix QP and leaky factor in 250 frames foreman with continuous error every 50 frames, and vary $BK$ from 0 (no MB in background slice using leaky prediction) to 4 (all the MB in background slice are using leaky prediction).

Fig.9 shows the error robustness performance with $BK$ for the overall frame only, in that $\Delta$PSNR curves of the ROI layer will not have much noticeable difference with varying $BK$. Generally, larger $BK$ corresponds to faster error recovery. Note that when $BK = 4$, i.e leaky prediction is applied to all macroblocks in one frame, $\Delta$PSNR decays almost to 0 for a periodical period of 4 seconds (error occurs every 50 frames and frames rate is 12.5Hz). The analysis of ROI layer error mitigation in Section.3 is also applicable here, except to replace MB in ROI slice with MB in either ROI or background.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>bitrate(kbps)</th>
<th>increase(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>62.96</td>
<td>0</td>
</tr>
<tr>
<td>0.9</td>
<td>83.77</td>
<td>33.05</td>
</tr>
<tr>
<td>0.8</td>
<td>93.03</td>
<td>47.76</td>
</tr>
<tr>
<td>0.5</td>
<td>116.48</td>
<td>85.01</td>
</tr>
</tbody>
</table>

Table 1. Bitrate varying with leaky factor $\alpha$ in foreman QCIF in error periodically every 50 frames (QP=28,12.5Hz) and bitrate increase compared to coding without leaky prediction.
slice. However, the bitrate would increase substantially with $BK$, as is illustrated in Tab.2, because larger $BK$ indicates that more background MB adjacent to ROI region would be involved in leaky prediction coding.

5. CONCLUSION

We have presented a novel scheme based on leaky prediction for ROI scalable coding to enhance the quality of ROI layer reconstruction. It makes a tradeoff between coding efficiency and error robustness in error-prone network. ROI layer reference is also involved in MC loop for prediction in addition to the overall frame. Error propagation due to the prediction mismatch is effectively controlled by leaky prediction in the ROI slice.

Simulation results show that the error in ROI layer of the video can be removed much faster from the decoded frame after a short period of time compared to ROI scalable coding without leaky prediction, and the overall frame error degradation is accelerated as well. And a significant gain of up to 2dB in PSNR is reported in both ROI layer and overall frame. Meanwhile it is observed the overall rate-distortion performance is close to that without leaky prediction in slow motion video, though a bit lower in high motion sequence.

6. REFERENCES