Prediction of Transmission Distortion for Wireless Video Communication: Analysis

Zhifeng Chen and Dapeng Wu

Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida 32611

Abstract—Transmitting video over wireless is a challenging problem since video may be seriously distorted due to packet errors caused by wireless channels. The capability of predicting transmission distortion (i.e., video distortion caused by packet errors) can assist in designing video encoding and transmission schemes that achieve maximum video quality or minimum end-to-end video distortion. This paper is aimed at deriving formulae for predicting transmission distortion. The contribution of this paper is two-folded. First, we identify the governing law that describes how the transmission distortion process evolves over time, and analytically derive the transmission distortion formula as a closed-form function of video frame statistics, channel error statistics, and system parameters. Second, we identify, for the first time, two important properties of transmission distortion. The first property is that the clipping noise, produced by non-linear clipping, causes decay of propagated error. The second property is that the correlation between motion vector concealment error and propagated error is negative, and has dominant impact on transmission distortion, compared to other correlations. Due to these two properties and elegant error/distortion decomposition, our formula provides not only more accurate prediction but also lower complexity than the existing methods.

Index Terms—Wireless video, transmission distortion, clipping noise, slice data partitioning (SDP), unequal error protection (UEP), time-varying channel.

I. INTRODUCTION

Both multimedia technology and mobile communications have experienced massive growth and commercial success in recent years. As the two technologies converge, wireless video, such as video phone and mobile TV in 3G/4G systems, is expected to achieve unprecedented growth and worldwide success. However, different from the traditional video coding system, transmitting video over wireless with good quality or low end-to-end distortion is particularly challenging since the received video is subject to not only quantization error but also transmission error. In a wireless video communication system, end-to-end distortion consists of two parts: quantization distortion and transmission distortion. Quantization distortion is caused by quantization errors during the encoding process, and has been extensively studied in rate distortion theory [1], [2]. Transmission distortion is caused by packet errors during the transmission of a video sequence, and it is the major part of the end-to-end distortion in delay-sensitive wireless video communication under high packet error probability (PEP), e.g., in a wireless fading channel.

The capability of predicting transmission distortion at the transmitter can assist in designing video encoding and transmission schemes that achieve maximum video quality under resource constraints. Specifically, transmission distortion prediction can be used in the following three applications in video encoding and transmission: 1) mode decision, which is to find the best intra/inter-prediction mode for encoding a macroblock (MB) with the minimum rate-distortion (R-D) cost given the instantaneous PEP, 2) cross-layer encoding rate control, which is to control the instantaneously encoded bit rate for a real-time encoder to minimize the frame-level end-to-end distortion given the instantaneous PEP, e.g., in video conferencing, 3) packet scheduling, which chooses a subset of packets of the pre-coded video to transmit and intentionally discards the remaining packets to minimize the group of picture (GOP)-level end-to-end distortion given the average PEP and average burst length, e.g., in streaming pre-coded video over networks. All the three applications require a formula for predicting how transmission distortion is affected by their respective control policy, in order to choose the optimal mode or encoding rate or transmission schedule.

However, predicting transmission distortion poses a great challenge due to the spatio-temporal correlation inside the input video sequence, the nonlinearity of both the encoder and the decoder, and varying PEP in time-varying channels. In a typical video codec, the temporal correlation among consecutive frames and the spatial correlation among the adjacent pixels of one frame are exploited to improve the coding efficiency. Nevertheless, such a coding scheme brings much difficulty in predicting transmission distortion because a packet error will degrade not only the video quality of the current frame but also the following frames due to error propagation. In addition, as we will see in Section III, the nonlinearity of both the encoder and the decoder makes the instantaneous transmission distortion not equal to the sum of distortions caused by individual error events. Furthermore, in a wireless fading channel, the PEP is time-varying, which makes the error process a non-stationary random process and hence, as a function of the error process, the distortion process is also a non-stationary random process.

According to the aforementioned three applications, the
existing algorithms for estimating transmission distortion can be categorized into the following three classes: 1) pixel-level or block-level algorithms (applied to mode decision), e.g., Recursive Optimal Per-pixel Estimate (ROPE) algorithm [3] and Law of Large Number (LLN) algorithm [4], [5]; 2) frame-level or packet-level or slice-level algorithms (applied to cross-layer encoding rate control) [6], [7], [8], [9], [10]; 3) GOP-level or sequence-level algorithms (applied to packet scheduling) [11], [12], [13], [14], [15]. Although the existing distortion estimation algorithms work at different levels, they share some common properties, which come from the inherent characteristics of wireless video communication system, that is, spatio-temporal correlation, nonlinear codec and time-varying channel. However, none of the existing works analyzed the effect of non-linear clipping noise on the transmission distortion, and therefore cannot provide accurate distortion estimation.

In this paper, we derive the transmission distortion formulae for wireless video communication systems. With consideration of spatio-temporal correlation, nonlinear codec and time-varying channel, our distortion prediction formulae improve the accuracy of distortion estimation from existing works. Besides that, our formulae support, for the first time, the following capabilities: 1) prediction at different levels (e.g., pixel/frame/GOP level), 2) prediction for multi-reference motion compensation, 3) prediction under slice data partitioning (SDP) [16], 4) prediction under arbitrary slice-level packetization with flexible macroblock ordering (FMO) mechanism [17], [18], 5) prediction under time-varying channel, 6) one unified formula for both I-MB and P-MB, and 7) prediction for both low motion and high motion video sequences. In addition, this paper also identifies two important properties of transmission distortion for the first time: 1) clipping noise, produced by non-linear clipping, causes decay of propagated error, and 2) the correlation between motion vector concealment error and propagated error is negative, and has dominant impact on transmission distortion, among all the correlations between any two of the four components in transmission error. Due to the page limit, we move most of the experimental results to our sequel paper [19], which 1) verify the accuracy of the formulae derived in this paper and compare that to existing models, 2) discuss the algorithms designed based on the formulae, 3) apply our algorithms in practical video codec design, and 4) compare the R-D performance between our algorithms and existing estimation algorithms.

The rest of the paper is organized as follows. Section II presents the preliminaries of our system model under study to facilitate the derivations in the later sections, and illustrates the limitations of existing transmission distortion models. In Section III, we derive the transmission distortion formula as a function of frame statistics, channel condition, and system parameters. Section IV concludes the paper.

II. SYSTEM DESCRIPTION

A. Structure of a Wireless Video Communication System

Fig. 1 shows the structure of a typical wireless video communication system. It consists of an encoder, two channels and a decoder where residual channel and motion vector (MV) channel may be either the same channel or different channels. If residual packets or MV packets are erroneous, the error concealment module will be activated. In typical video encoders such as H.263/264 and MPEG-2/4, the functional blocks can be divided into two classes: 1) basic parts, such as predictive coding, transform, quantization, entropy coding, motion compensation, and clipping; and 2) performance-enhancing parts, such as interpolation filtering, deblocking filtering, B-frame, multi-reference prediction, etc.

Although the up-to-date video encoder includes more and more performance-enhancing parts, the basic parts do not change. In this paper, we analyze the transmission distortion for the structure with the basic parts in Fig. 1.

Note that in this system, both residual channel and MV channel are application-layer channels; specifically, both channels consist of entropy coding and entropy decoding, networking layers2, and physical layer (including channel encoding, modulation, wireless fading channel, demodulation, channel decoding). Although the residual channel and MV channel usually share the same physical-layer channel, the two application-layer channels may have different parameter settings (e.g., different channel code-rate) for different SDP packets under unequal error protection (UEP) consideration.

Table I lists notations used in this paper. All vectors are in bold font. Note that the encoder needs to reconstruct the compressed video for predictive coding; hence the encoder and the decoder have a similar structure for pixel value reconstruction. To distinguish the variables in the reconstruction module of the decoder from those in the reconstruction module of the encoder, we add \( \hat{\cdot} \) on top of the variables at the encoder and add \( \sim \) on top of the variables at the decoder.

B. Clipping Noise

In this subsection, we examine the effect of clipping noise on the reconstructed pixel value along each pixel trajectory over time (frames). All pixel positions in a video sequence form a three-dimensional spatio-temporal domain, i.e., two dimensions in spatial domain and one dimension in temporal domain. Each pixel can be uniquely represented by \( u^k \) in this

\[ \hat{u}^k = u^k + e^k \]

\[ e^k = u^k - \hat{u}^k \]

\[ e^k = \begin{cases} u^k, & u^k \leq 0 \\ u^k, & u^k > 0 \end{cases} \]

\[ T \{ \hat{u}^k \} = \sum_{k=1}^{K} \hat{u}^k \]

\[ Q \{ \hat{u}^k \} = \sum_{k=1}^{K} \hat{u}^k \]

\[ Q^{-1} \{ \hat{u}^k \} = \sum_{k=1}^{K} \hat{u}^k \]

\[ S(r) \]

Fig. 1. System structure, where T, Q, Q\(^{-1}\), and T\(^{-1}\) denote transform, quantization, inverse quantization, and inverse transform, respectively.
three-dimensional time-space, where \( k \) means the \( k \)-th frame in temporal domain and \( u \) is a two-dimensional vector in spatial domain, i.e. position in the \( k \)-th frame. The philosophy behind inter-prediction of a video sequence is to represent the video sequence by virtual motion of each pixel, i.e., each pixel recursively moves from position \( v \) in the \( k-1 \)-frame, i.e. \( v^{k-1} \), to position \( u^k \). The difference between these two positions is a two-dimensional vector called MV of pixel \( u^k \), i.e., \( mv_k^u = v^{k-1} - u^k \). The difference between the pixel values of these two positions is called residual of pixel \( u^k \), that is, \( e_k^u = f_k^u - f_{k-1}^{u+mv_k^u} \). Recursively, each inter-predicted pixel in the \( k \)-th frame has one and only one reference pixel trajectory backward towards the latest I-block.\(^3\)

At the encoder, after transform, quantization, inverse quantization, and inverse transform for the residual, the reconstructed pixel value for \( u^k \) may be out-of-range and should be clipped as

\[
\hat{f}_u^k = \Gamma(f_{u+mv_k}^k + e_k^u),
\]

where \( \Gamma(\cdot) \) is a clipping function defined by

\[
\Gamma(x) = \begin{cases} 
\gamma_l, & x < \gamma_l \\
\gamma_l \leq x \leq \gamma_h, & \gamma_h, & x > \gamma_h
\end{cases}
\]

where \( \gamma_l \) and \( \gamma_h \) are user-specified low threshold and high threshold, respectively. Usually, \( \gamma_l = 0 \) and \( \gamma_h = 255 \).

The residual and MV at the decoder may be different from their counterparts at the encoder because of channel impairments. Denote \( \tilde{mv}_u^k \) and \( \tilde{e}_u^k \) the MV and residual at the decoder, respectively. Then, the reference position for \( u^k \) at the decoder is \( \tilde{v}^{k-1} = u^k + \tilde{mv}_u^k \), and the reconstructed pixel value for \( u^k \) at the decoder is

\[
\hat{f}_u^k = \Gamma(f_{u+\tilde{mv}_u^k}^{k-1} + \tilde{e}_u^k). \tag{3}
\]

In error-free channels, the reconstructed pixel value at the receiver is exactly the same as the reconstructed pixel value at the transmitter, because there is no transmission error and hence no transmission distortion. However, in error-prone channels, we know from (3) that \( \tilde{f}_u^k \) is a function of three factors: the received residual \( \tilde{e}_u^k \), the received MV \( \tilde{mv}_u^k \), and the propagated error \( f_{u+\tilde{mv}_u^k}^{k-1} \). The received residual \( \tilde{e}_u^k \) depends on three factors, namely, 1) the transmitted residual \( \tilde{e}_u^k \), 2) the residual packet error state, which depends on instantaneous residual channel condition, and 3) the residual error concealment algorithm if the received residual packet is erroneous. Similarly, the received MV \( \tilde{mv}_u^k \) depends on 1) the transmitted \( mv_u^k \), 2) the MV packet error state, which depends on instantaneous MV channel condition, and 3) the MV error concealment algorithm if the received MV packet is erroneous. The propagated error \( f_{u+\tilde{mv}_u^k}^{k-1} \) includes the error propagated from the reference frames, and therefore depends on all samples in the previous frames indexed by \( i \), where \( 1 \leq i < k \) and their reception error states as well as error concealment algorithms. In this paper, we consider the temporal error concealment [20, 21] in deriving the transmission distortion formulae.

The non-linear clipping function within the pixel trajectory makes the distortion estimation more challenging. However, it is interesting to observe that clipping actually reduces transmission distortion. In Section III, we will quantify the effect of clipping on transmission distortion.

### C. Definition of Transmission Distortion

In a video sequence, all pixel positions in the \( k \)-th frame form a two-dimensional vector set \( V^k \), and we denote the number of elements in set \( V^k \) by \( |V^k| \). So, for any pixel at position \( u \) in the \( k \)-th frame, i.e., \( u \in V^k \), its reference pixel position is chosen from set \( V^{k-1} \) for single-reference motion compensation.

Given the joint probability mass function (PMF) of \( f_u^k \) and \( \hat{f}_u^k \), we define the pixel-level transmission distortion (PTD) for pixel \( u^k \) by

\[
D_u^k \triangleq E[(\hat{f}_u^k - f_u^k)^2], \tag{4}
\]

where \( E[\cdot] \) represents expectation and the randomness comes from both random video input and random channel error state. Then, we define the frame-level transmission distortion (FTD) for the \( k \)-th frame by

\[
D^k \triangleq E\left[ \frac{1}{|V^k|} \sum_{u \in V^k} (\hat{f}_u^k - f_u^k)^2 \right]. \tag{5}
\]

It is easy to prove that the relationship between FTD and PTD is characterized by

\[
D^k = \frac{1}{|V^k|} \sum_{u \in V^k} D_u^k. \tag{6}
\]

In fact, (6) is a general form for distortions of all levels. If \(|V^k| = 1\), (6) reduces to (4). For slice/packet-level distortion, \( V^k \) is the set of the pixels contained in a slice/packet. For GOP-level distortion, \( V^k \) could be replaced by the set of the pixels contained in a GOP. In this paper, we only show how to derive formulae for PTD and FTD. Our methodology is also applicable to deriving formulae for slice/packet/GOP-level distortion by using appropriate \( V^k \).

---

\(^3\)We will also discuss intra-predicted pixels in Section III.
D. Limitations of the Existing Transmission Distortion Models

We define the clipping noise for pixel \( u^k \) at the encoder as
\[
\hat{\Delta}_u \triangleq (\hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k) - \Gamma(\hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k),
\]
and the clipping noise for pixel \( u^k \) at the decoder as
\[
\tilde{\Delta}_u \triangleq (\hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k) - \Gamma(\hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k).
\]
Using (1), Eq. (7) becomes
\[
\hat{f}_u^k = \hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k - \hat{\Delta}_u,
\]
and using (3), Eq. (8) becomes
\[
\tilde{f}_u^k = \tilde{f}^{k-1}_{u+mv_u^k} + \tilde{c}_u^k - \tilde{\Delta}_u,
\]
where \( \hat{\Delta}_u \) only depends on the video content and encoder structure, e.g., motion estimation, quantization, mode decision and clipping function; and \( \tilde{\Delta}_u \) depends on not only the video content and encoder structure, but also channel conditions and decoder structure, e.g., error concealment and clipping function. In most existing works [3], [7], [9], [10], [15], both \( \hat{\Delta}_u \) and \( \tilde{\Delta}_u \) are neglected, i.e., these works assume \( \hat{f}_u^k = \hat{f}^{k-1}_{u+mv_u^k} + \hat{c}_u^k \) and \( \tilde{f}_u^k = \tilde{f}^{k-1}_{u+mv_u^k} + \tilde{c}_u^k \). However, this assumption is only valid for stored video or error-free communication, where \( \hat{\Delta}_u = \tilde{\Delta}_u \) since \( \hat{\Delta}_u = 0 \) with very high probability. For error-prone communication, decoder clipping noise \( \tilde{\Delta}_u \) has a significant impact on transmission distortion and hence should not be neglected. Without taking into consideration \( \hat{\Delta}_u \), the estimated distortion can be much larger than true distortion [22].

III. Transmission Distortion Formulae

In this section, we derive formulae for PTD and FTD. The section is organized as follows: Section III-A presents an overview of our approach to analyzing PTD and FTD. Then, we elaborate on the derivation details in Section III-B through Section III-E. Specifically, Section III-B quantifies the effect of residual concealment error (RCE) on transmission distortion; Section III-C quantifies the effect of motion vector concealment error (MVCE) on transmission distortion; Section III-D quantifies the effect of propagated error and clipping noise on transmission distortion; Section III-E quantifies the effect of correlations (between any two of the error sources) on transmission distortion. Finally, Section III-F summarizes the key results of this paper, i.e., the formulae for PTD and FTD.

A. Overview of the Approach to Analyzing PTD and FTD

To analyze PTD and FTD, we take a divide-and-conquer approach. We first divide transmission reconstructed error into four components: three random errors (RCE, MVCE and propagated error) due to their different physical causes, and clipping noise, which is a non-linear function of these three random errors. This error decomposition allows us to further decompose transmission distortion into four terms, i.e., distortion caused by 1) RCE, 2) MVCE, 3) propagated error plus clipping noise, and 4) correlations between any two of the error sources, respectively. This distortion decomposition facilitates the derivation of a simple and accurate closed-form formula for each of the four distortion terms. Next, we elaborate on error decomposition and distortion decomposition.

Define transmission reconstructed error for pixel \( u^k \) by \( \hat{\zeta}_u^k \triangleq f_u^k - \hat{f}_u^k \). From (9) and (10), we obtain
\[
\hat{\zeta}_u^k = (\hat{f}_u^k + \hat{c}_u^k) - \hat{\Delta}_u - (\hat{f}_u^k + \hat{c}_u^k) + \tilde{f}_u^k = (\hat{\Delta}_u - \tilde{\Delta}_u) = (\hat{\Delta}_u - \tilde{\Delta}_u).
\]
Define RCE \( \hat{\zeta}_u^k \) by \( \hat{\zeta}_u^k \triangleq \hat{\Delta}_u - \tilde{\Delta}_u \) and define MVCE \( \hat{\xi}_u^k \) by \( \hat{\xi}_u^k \triangleq \hat{f}_u^k - \hat{c}_u^k \). Denote that \( \hat{f}_u^k + \hat{c}_u^k - f_u^k - c_u^k = \hat{\zeta}_u^k + \hat{\xi}_u^k = f_u^k + c_u^k \), which is the transmission reconstructed error of the concealed reference pixel in the reference frame; we call \( \hat{\zeta}_u^k \) propagated error. As mentioned in Section II-D, we assume \( \hat{\Delta}_u = 0 \). Therefore, (11) becomes
\[
\tilde{\zeta}_u^k = \hat{\zeta}_u^k + \tilde{\xi}_u^k \triangleq \hat{\Delta}_u + \tilde{\Delta}_u.
\]
(12) is our proposed error decomposition. In Table II, we list the abbreviations that will be used frequently in the following sections.

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DEFINITIONS</strong></td>
</tr>
<tr>
<td>RCE : residual concealment error</td>
</tr>
<tr>
<td>MVCE: motion vector concealment error</td>
</tr>
<tr>
<td>PTD : pixel-level transmission distortion</td>
</tr>
<tr>
<td>FTD : frame-level transmission distortion</td>
</tr>
<tr>
<td>XEP : pixel error probability</td>
</tr>
<tr>
<td>PEP : packet error probability</td>
</tr>
<tr>
<td>FMO : flexible macroblock ordering</td>
</tr>
<tr>
<td>UEP : unequal error protection</td>
</tr>
<tr>
<td>SDP : slice data partitioning</td>
</tr>
<tr>
<td>PMF : probability mass function</td>
</tr>
</tbody>
</table>

Combining (4) and (12), we have
\[
D_u^k = E[|\hat{\zeta}_u^k|^2] + E[|\hat{\xi}_u^k|^2] + E[|\hat{\Delta}_u^k|^2] - 2E[\hat{\zeta}_u^k \cdot \hat{\xi}_u^k] \triangleq D_u^k(0) + D_u^k(m) + D_u^k(P) + D_u^k(c).
\]
Denote \( D_u^k(r) \triangleq E[|\hat{\zeta}_u^k|^2] \), \( D_u^k(m) \triangleq E[|\hat{\zeta}_u^k|^2] + E[|\hat{\xi}_u^k|^2] \) and \( D_u^k(c) \triangleq 2E[\hat{\zeta}_u^k \cdot \hat{\xi}_u^k] \). Then, (13) becomes
\[
D_u^k = D_u^k(r) + D_u^k(m) + D_u^k(P) + D_u^k(c).
\]
are 8 possible error events due to three individual random errors, there are a total of $8 \times (5 + 10) = 120$ terms for PTD, making the analysis highly complicated. In contrast, our decompositions in (12) and (14) significantly simplify the analysis. Second, each term in (12) and (14) has a clear physical meaning, which lessens the requirement for joint PMF of $f_k^u$ and $f_k^m$ and leads to accurate estimation algorithms with low complexity. Third, such decompositions allow our formulation to be easily extended for supporting advanced video codec with more performance-enhancing parts, e.g., multi-reference prediction [22] and interpolation filtering in fractional-pel motion estimation [23].

To derive the formula for FTD, from (6) and (14), we obtain

$$D^k = D^k(r) + D^k(m) + D^k(P) + D^k(c),$$

where

$$D^k(r) = \frac{1}{|V|} \cdot \sum_{u \in V^k_k} D_{u}^k(r),$$

$$D^k(m) = \frac{1}{|V|} \cdot \sum_{u \in V^k_k} D_{u}^k(m),$$

$$D^k(P) = \frac{1}{|V|} \cdot \sum_{u \in V^k_k} D_{u}^k(P),$$

$$D^k(c) = \frac{1}{|V|} \cdot \sum_{u \in V^k_k} D_{u}^k(c).$$

(15) is our proposed distortion decomposition for FTD. Usually, the cardinality, i.e. the number of elements, of set $V^k_k$ in a video sequence is the same for all frames. That is, $|V^k_k| = \cdots = |V^k_k| = |V|$ for all $k \geq 1$. Hence, we remove the frame index $k$ and denote $|V^k_k|$ for all $k \geq 1$ by $|V|$. Note that in a video codec, e.g. H.264 [16], a reference pixel may be in a position out of picture boundary; however, the cardinality of set consisting of reference pixels, although larger than the cardinality of the input pixel set $|V|$, is still the same for all frames.

### B. Analysis of Distortion Caused by RCE

In this subsection, we first derive the pixel-level residual caused distortion $D_{u}^k(r)$. Then, we derive the frame-level residual caused distortion $D^k(r)$.  

1) **Pixel-level Distortion Caused by RCE:** We denote $S_{u}^k$ as the state indicator of whether there is transmission error for pixel $u^k_k$ after channel decoding. Note that as mentioned in Section II-A, both the residual channel and the MV channel contain channel decoding; hence in this paper, the transmission error in the residual channel or the MV channel is meant to be an uncorrectable error after channel decoding. To distinguish the residual error state and the MV error state, here we use $S_{u}^k(r)$ to denote the residual error state for pixel $u^k_k$. That is, $S_{u}^k(r) = 1$ if $\hat{e}_u^k$ is received with error, and $S_{u}^k(r) = 0$ if $\hat{e}_u^k$ is received without error. At the receiver, if there is no residual transmission error for pixel $u^k_k$, $\tilde{c}_u^k$ is equal to $\hat{e}_u^k$.

However, if the residual packets are received with error, we need to conceal the residual error at the receiver. Denote $\hat{e}_u^k$ the concealed residual when $S_{u}^k(r) = 1$, and we have,

$$\tilde{c}_u^k = \begin{cases} \hat{e}_u^k, & S_{u}^k(r) = 1 \\ \hat{e}_u^k, & S_{u}^k(r) = 0. \end{cases}$$

(20)

Note that $\hat{e}_u^k$ depends on $\hat{e}_u^k$ and the residual concealment method, but does not depend on the channel condition. From the definition of $\hat{e}_u^k$ and (20), we have

$$\hat{e}_u^k = (\hat{e}_u^k - S_{u}^k(r) \cdot S_{u}^k(r)) = (\hat{e}_u^k - S_{u}^k(r)) \cdot (1 - S_{u}^k(r)).$$

(21)

$\hat{e}_u^k$ depends on the input video sequence and the encoder structure, while $S_{u}^k(r)$ depends on the random multiplicative and additive noises in the wireless channel. Under our framework shown in Fig. 1, the input video sequence and the encoder structure are independent of communication system parameters. Therefore, we assume $\hat{e}_u^k$ and $S_{u}^k(r)$ are independent as the following assumption.

**Assumption 1:** $S_{u}^k(r)$ is independent of $\hat{e}_u^k$.

Denote $\varepsilon_u^k \equiv \hat{e}_u^k - \hat{e}_u^k$; we have $\hat{e}_u^k = \hat{e}_u^k \cdot S_{u}^k(r)$. Denote $P_{u}^k(r)$ as the residual pixel error probability (XEP) for pixel $u^k_k$, that is, $P_{u}^k(r) = 1 - P\{S_{u}^k(r) = 1\}$. Then, given $P_{u}^k(r)$, from (21) and Assumption 1, we have

$$D_{u}^k(r) = E[(\varepsilon_u^k)^2] = E[(\hat{e}_u^k)^2] \cdot E[(S_{u}^k(r))^2] = E[(\hat{e}_u^k)^2] \cdot (1 - P_{u}^k(r)).$$

(22)

Hence, our formula for the pixel-level residual caused distortion is

$$D_{u}^k(r) = E[(\varepsilon_u^k)^2] \cdot P_{u}^k(r).$$

(23)

Note that we may also generalize (23) for I-MB. For pixels in I-MB, if the packet containing those pixels has error, $\hat{e}_u^k$ is still available since all the erroneous pixels will be concealed in the same way. However, since there is no $\hat{e}_u^k$ available, in order to use (23) to predict the transmission distortion, we may need to find the best reference, in terms of R-D cost, for the reconstructed I-MB by doing a virtual motion estimation and then calculate $\varepsilon_u^k$ for (23). The estimated $\hat{m}_{u}^k$ can be used to predict $D_{u}^k(m)$ for I-MB in the next subsection. An alternative method to calculate $\varepsilon_u^k$ for I-MB is to use the same position of previous frame as reference, i.e. assuming $\hat{m}_{u}^k = 0$. Note that if the packet containing those pixels in I-MB is correctly received, $D_{u}^k(r) = 0$.

2) **Frame-level Distortion Caused by RCE:** To derive the frame-level residual caused distortion, the encoder needs to know the second moment of RCE for each pixel in that frame. In most, if not all, existing distortion models [3], [7], [9], [10], [15], the residual error concealment method is to let $\hat{e}_u^k = 0$ for all erroneous pixels. However, as long as $\hat{e}_u^k$ and $\hat{e}_u^k$ satisfy some properties, we can derive a formula for more general residual error concealment methods instead of assuming $\hat{e}_u^k = 0$. We make the following assumption for $\hat{e}_u^k$ and $\hat{e}_u^k$.

Note that although they have the same cardinality, different sets are very different, i.e. $\forall k \neq k$. 

$^3$P_{u}^k(r) depends on the communication system parameters such as delay bound, channel coding rate, transmission power, channel gain of the wireless channel.
Assumption 2: The residual $\tilde{e}_u^k$ is stationary with respect to 2D variable $u$ in the same frame. In addition, $\tilde{e}_u^k$ only depends on $\{\tilde{e}_v^k : v \in N_u\}$ where $N_u$ is a fixed neighborhood of $u$.

In other words, Assumption 2 assumes that 1) $\tilde{e}_u^k$ is a 2D stationary stochastic process and the distribution of $\tilde{e}_u^k$ is the same for all $u \in V^k$, and 2) $\tilde{e}_u^k$ is also a 2D stationary stochastic process since it only depends on the neighboring $\tilde{e}_u^k$. Hence, $\tilde{e}_u^k - \tilde{e}_u^k$ is also a 2D stationary stochastic process, and its second moment $E[(\tilde{e}_u^k - \tilde{e}_u^k)^2] = E[(\tilde{e}_u^k)^2]$ is the same for all $u \in V^k$. Therefore, we can drop $u$ from the notation, and let $E[(\epsilon_{\tilde{u}}^k)^2] = E[(\epsilon_{\tilde{u}}^k)^2]$ for all $u \in V^k$.

Denote $N^k_r(r)$ as the number of pixels contained in the $i$-th residual packet of the $k$-th frame; denote $P^k_i(r)$ as PEP of the $i$-th residual packet of the $k$-th frame; denote $N^k_i$ as the total number of residual packets of the $k$-th frame. Since for all pixels in the same packet, the residual XEP is equal to its PEP, from (16) and (23), we have

$$D^k(r) = \frac{1}{|V|} \sum_{u \in V^k} E[(\epsilon_{\tilde{u}}^k)^2] \cdot P_u^k(r)$$

$$= \frac{1}{|V|} \sum_{u \in V^k} E[(\epsilon_{\tilde{u}}^k)^2] \cdot P^k_i(r)$$

$$= \frac{E[(\epsilon_{\tilde{u}}^k)^2]}{|V|} \sum_{i=1}^{N^k_r(r)} (P^k_i(r) \cdot N^k_i(r))$$

$$= E[(\epsilon_{\tilde{u}}^k)^2] \cdot \bar{P}^k_i(r).$$

where (a) is due to $P^k_i(r) = P^k_i(r)$ for pixel $u$ in the $i$-th residual packet; (b) is due to

$$\bar{P}^k_i(r) \triangleq \frac{1}{|V|} \sum_{i=1}^{N^k_r(r)} (P^k_i(r) \cdot N^k_i(r)) = \frac{1}{|V|} \sum_{i=1}^{N^k_r(r)} (P^k_i(r) \cdot N^k_i(r)).$$

$\bar{P}^k_i(r)$ is a weighted average over PEP of all residual packets in the $k$-th frame, in which different packets may contain different numbers of pixels. Hence, given PEP of all residual packets in the $k$-th frame, our formula for the frame-level residual caused distortion is

$$D^k(r) = E[(\epsilon_{\tilde{u}}^k)^2] \cdot \bar{P}^k_i(r).$$

Note that with FMO mechanism, many neighboring pixels may be encoded into different slices and transmitted in different packets. Since each packet may experience different PEP especially in a fast fading channel, even neighboring pixels may have very different XEP. Therefore, (29) works perfectly under the FMO consideration. This situation is taken into consideration throughout this paper.

C. Analysis of Distortion Caused by MVCE

Similar to the derivations in Section III-B1, in this subsection, we derive the formula for the pixel-level MV caused distortion $D^k_u(m)$, and the frame-level MV caused distortion $D^k(m)$.

1) Pixel-level Distortion Caused by MVCE: Denote the MV error state for pixel $u^k$ by $S^k_{u^k}(m)$, and denote the concealed MV by $\hat{m}^k_{u^k}$ for general temporal error concealment methods when $S^k_{u^k}(m) = 1$. Therefore, we have

$$m^k_{u^k} = \begin{cases} m^k_{u^k}, & S^k_{u^k}(m) = 1 \\ \hat{m}^k_{u^k}, & S^k_{u^k}(m) = 0. \end{cases}$$

Denote $\xi^k_u \triangleq f^{k-\hat{m}^k_{u^k}} - f^{k-1}$, where $\xi^k_u$ depends on the accuracy of MV concealment, and the spatial correlation between reference pixel and concealed reference pixel at the encoder. A more comprehensive analysis of effect of inaccurate MV estimation on $\xi^k_u$ can be found in Ref. [24], which is then extended to support multi-hypothesis motion-compensated prediction [25] and to derive a rate-distortion model taking into account the temporal prediction distance [26].

We also make the following assumption.

Assumption 3: $S^k_{u^k}(m)$ is independent of $\xi^k_u$.

Denote $P^k_u(m)$ as the MV XEP for pixel $u^k$, that is, $P^k_u(m) \triangleq P(S^k_{u^k}(m) = 1)$. Note that it is possible that $P^k_u(m) \neq P^k_u(r)$ if SDP and UEP are applied. Given $P^k_u(m)$, following the same derivation process in Section III-B1, we can obtain

$$D^k_u(m) = E[(\epsilon_{\tilde{u}}^k)^2] \cdot P^k_u(m).$$

Also note that in H.264 specification [16], there is no SDP for an instantaneous decoding refresh (IDR) frame; so $S^k_u = S^k_{u^k}(m)$ in an IDR-frame and hence $P^k_u(m) = P^k_u(r)$. This is also true for MB without SDP. For P-MB with SDP in H.264, $S^k_{u^k}(m)$ and $S^k_{u^k}(m)$ are dependent. In other words, if the MV packet is lost, the corresponding residual packet cannot be decoded even if it is correctly received, since there is no slice header in the residual packet. Therefore, the residual channel and the MV channel in Fig. 1 are actually dependent if the encoder follows H.264 specification. In this paper, we study transmission distortion in a more general case where $S^k_{u^k}(m)$ and $S^k_{u^k}(m)$ can be either independent or dependent.\(^6\)

2) Frame-level Distortion Caused by MVCE: To derive the frame-level MV caused distortion, we also make the following assumption.

Assumption 4: The second moment of $\xi^k_u$ is the same for all $u \in V^k$.

Under Assumption 4, we can drop $u$ from the notation, and let $E[(\epsilon_{\tilde{u}}^k)^2] = E[(\epsilon_{\tilde{u}}^k)^2]$ for all $u \in V^k$. Denote $N^k_i$ as the number of pixels contained in the $i$-th MV packet of the $k$-th frame; denote $P^k_i(m)$ as PEP of the $i$-th MV packet of the $k$-th frame; denote $N^k(m)$ as the total number of MV packets of the $k$-th frame. Then, given PEP of all MV packets in the $k$-th frame, following the same derivation process in Section III-B2, we obtain the frame-level MV caused distortion for the $k$-th frame as

$$D^k(m) = E[(\epsilon_{\tilde{u}}^k)^2] \cdot \bar{P}^k(m),$$

\(^6\)To achieve this, we change the H.264 reference code JM14.0 by allowing residual packets to be used for decoder without the corresponding MV packets being correctly received, that is, $\tilde{e}_u^k$ can be used to reconstruct $\bar{P}^k(m)$ even if $m^k_{u^k}$ is not correctly received.
where $P^k(m) \triangleq \frac{1}{|\mathbb{M}|} \sum_{i=1}^{N_u^k} (D^k_i(m) \cdot N^k(m))$, a weighted average over PEP of all MV packets in the $k$-th frame, in which different packets may contain different numbers of pixels.

D. Analysis of Distortion Caused by Propagated Error Plus Clipping Noise

In this subsection, we derive the distortion caused by error propagation in a non-linear decoder with clipping. We first derive the pixel-level propagation and clipping caused distortion $D^k_u(P)$. Then, we derive the frame-level propagation and clipping caused distortion $D^k(P)$.

1) Pixel-level Distortion Caused by Propagated Error Plus Clipping Noise: First, we analyze the pixel-level propagation and clipping caused distortion $D^k_u(P)$ in P-MBs. From the definition, we know $D^k_u(P)$ depends on propagated error and clipping noise; and clipping noise is a function of RCE, MVCE and propagated error. Hence, $D^k_u(P)$ depends on RCE, MVCE and propagated error. Let $r, m, p$ denote the event of occurrence of RCE, MVCE and propagated error respectively, and let $\bar{r}, \bar{m}, \bar{p}$ denote logical NOT of $r, m, p$ respectively (indicating no error). We use a triple to denote the joint event of three types of error; e.g., $\{r, m, p\}$ denotes the event that all the three types of errors occur, and $u^k(\bar{r}, \bar{m}, \bar{p})$ denotes the pixel $u^k$ experiencing none of the three types of errors.

When we analyze the condition that several error events may occur, the notation could be simplified by the principle of formal logic. For example, $\Delta^k_u(\bar{r}, \bar{m})$ denotes the clipping noise under the condition that there is neither RCE nor MVCE for pixel $u^k$, while it is not certain whether the reference pixel has error. Correspondingly, denote $P^k_u(\bar{r}, \bar{m})$ as the probability of event $\{\bar{r}, \bar{m}\}$, that is, $P^k_u(\bar{r}, \bar{m}) = P(S^k_u(r) = 0$ and $\Delta^k_u(m) = 0)$. From the definition of $P^k_u(r)$, the marginal probability $P^k_u(\bar{r}) = 1 - P^k_u(r)$ and the marginal probability $P^k_u(\bar{m}) = 1 - P^k_u(m)$. Similarly, $P^k_u(m) = P^k_u(\bar{m})$ and $P^k_u(\bar{m}) = 1 - P^k_u(m)$.

Define $D^k_u(p) \triangleq E(\tilde{e}^k_{u+mv} + \Delta^k_u(\bar{r}, \bar{m}))^2$; and define $\alpha^k_u \triangleq \frac{D^k_u(p)}{P^k_u(\bar{r}, \bar{m})}$, which is called propagation factor for pixel $u^k$. The propagation factor $\alpha^k_u$ defined in this paper is different from the propagation factor [10], leakage [7], or attenuation factor [15], which are modeled as the effect of spatial filtering or intra update; our propagation factor $\alpha^k_u$ is also different from the fading factor [8], which is modeled as the effect of using fraction of referenced pixels in the reference frame for motion prediction. Note that $D^k_u(p)$ is only a special case of $D^k_u(P)$ under the error event of $\{\bar{r}, \bar{m}\}$ for pixel $u^k$. However, most existing models inappropriately use their propagation factor, obtained under the error event of $\{\bar{r}, \bar{m}\}$, to replace $D^k_u(P)$ directly.

To calculate $E(\tilde{e}^k_{u+mv} + \Delta^k_u)^2$ in (13), we need to analyze $\Delta^k_u$ in four different error events for pixel $u^k$: 1) both residual and MV are erroneous, denoted by $u^k(\bar{r}, \bar{m})$; 2) residual is erroneous but MV is correct, denoted by $u^k(\bar{r}, m)$; 3) residual is correct but MV is erroneous, denoted by $u^k(\bar{r}, \bar{m})$; and 4) both residual and MV are correct, denoted by $u^k(r, \bar{m})$. So, $D^k_u(P) = P^k_u(r, m) \cdot E(\tilde{e}^k_{u+mv} + \Delta^k_u)^2 + P^k_u(\bar{r}, m) \cdot E(\tilde{e}^k_{u+mv} + \Delta^k_u)^2 + P^k_u(\bar{r}, \bar{m}) \cdot E(\tilde{e}^k_{u+mv} + \Delta^k_u)^2 + P^k_u(r, \bar{m}) \cdot E(\tilde{e}^k_{u+mv} + \Delta^k_u)^2$.

Note that the concealed pixel value should be in the clipping function range, that is, $\Delta^k_u(\bar{r}) = 0$. Also note that if the MV channel is independent of the residual channel, we have $P^k_u(r, m) = P^k_u(r) \cdot P^k_u(m)$. However, as mentioned in Section III-C1, in H.264 specification, these two channels are dependent. In other words, $P^k_u(\bar{r}, m) = 0$ and $P^k_u(\bar{r}, \bar{m}) = P^k_u(\bar{r})$ for P-MBs with SDP in H.264. In such a case, (33) is simplified to $D^k_u(P) = P^k_u(r, m) \cdot D^k_{u+mv} + P^k_u(\bar{r}, \bar{m}) \cdot D^k_{u+mv} + P^k_u(\bar{r}, \bar{m}) \cdot D^k_u(\bar{r})$.

Note that for P-MB without SDP, we have $P^k_u(\bar{r}, \bar{m}) = P^k_u(\bar{r}) = \tilde{e}^k_u + \tilde{e}^k_{u+mv}$, so $\Delta^k_u(\bar{r}) = 0$. Also note that if the MV channel is independent of the residual channel, we have $P^k_u(r, m) = P^k_u(r) \cdot P^k_u(m)$. However, as mentioned in Section III-C1, in H.264 specification, these two channels are dependent. In other words, $P^k_u(\bar{r}, m) = 0$ and $P^k_u(\bar{r}, \bar{m}) = P^k_u(\bar{r})$ for P-MBs with SDP in H.264. In such a case, (33) is simplified to $D^k_u(P) = P^k_u(r, m) \cdot D^k_{u+mv} + (1 - P^k_u) \cdot D^k_u(\bar{r})$.

Also note that for I-MB, there will be no transmission distortion if it is correctly received, that is, $D^k_u(p) = 0$. So (35) can be further simplified to $D^k_u(P) = P^k_u \cdot D^k_{u+mv}$.

Comparing (36) with (35), we see that I-MB is a special case of P-MB with $D^k_u(p) = 0$, that is, the propagation factor $\alpha^k_u = 0$ according to the definition. It is important to note that $D^k_u(P) > 0$ for I-MB since $P^k_u \neq 0$. In other words, I-MB also contains the distortion caused by propagation error and it can be predicted by (36). However, existing linear time-invariant (LTI) models [7], [8] assume that there is no distortion caused by propagation error for I-MB, which underestimates the transmission distortion.

In the following part of this subsection, we derive the propagation factor $\alpha^k_u$ for P-MB and prove some important properties of clipping noise. To derive $\alpha^k_u$, we first give Lemma 1 as below.

Lemma 1: Given the PMF of the random variable $\tilde{e}^k_{u+mv}$ and the value of $f^k_u D^k_u(p)$ can be calculated at the encoder by $D^k_u(p) = E(\Phi^2(\tilde{e}^k_{u+mv}, \tilde{f}^k_u))$, where $\Phi(x, y)$ is called error reduction function and defined by $\Phi(x, y) \triangleq y - \Gamma(y - x)$.
Lemma 1 is proved in Appendix A. In fact, we have found in our experiments that in any error event, \( \tilde{\zeta}_{u+\text{mv}_1}^{k-1} \) approximately follows Laplacian distribution with zero mean. If we assume \( \zeta_{u+\text{mv}_1}^{k-1} \) follows Laplacian distribution with zero mean, the calculation for \( D_k^u(p) \) becomes simpler since the only unknown parameter for PMF of \( \zeta_{u+\text{mv}_1}^{k-1} \) is its variance. Under this assumption, we have the following proposition.

**Proposition 1:** The propagation factor \( \alpha \) for propagated error with Laplacian distribution of zero-mean and variance \( \sigma^2 \) is given by

\[
\alpha = 1 - \frac{1}{2} e^{-\frac{y-y_0}{b}} \left( \frac{y-y_0}{b} + 1 \right) - \frac{1}{2} e^{-\frac{y_h-y}{b}} \left( \frac{y_h-y}{b} + 1 \right),
\]

where \( y \) is the reconstructed pixel value, and \( b = \frac{\sqrt{2}}{\sigma} \).

Proposition 1 is proved in Appendix B. In the zero-mean Laplacian case, \( \alpha^u_k \) will only be a function of \( \tilde{f}_k^u \) and the variance of \( \zeta_{u+\text{mv}_1}^{k-1} \), which is equal to \( D_{u+\text{mv}_1}^{k-1} \) in this case. Since \( D_{u+\text{mv}_1}^{k-1} \) has already been calculated during the phase of predicting the \((k-1)\)-th frame transmission distortion, \( \hat{D}_u^k(p) \) can be calculated by \( \hat{D}_u^k(p) = \alpha^u_k \cdot D_{u+\text{mv}_1}^{k-1} \) via the definition of \( \alpha^u_k \). Then, we can recursively calculate \( \hat{D}_u^k(P) \) in (34) since both \( D_{u+\text{mv}_1}^{k-2} \) and \( D_{u+\text{mv}_1}^{k-1} \) have been calculated previously for the \((k-1)\)-th frame.

Next, we prove an important property of the non-linear clipping function in Proposition 2. To prove Proposition 2, we need to use the following lemma.

**Lemma 2:** The error reduction function \( \Phi(x, y) \) satisfies \( \Phi^2(x, y) \leq x^2 \) for any \( \gamma_i \leq y \leq \gamma_h \).

Lemma 2 is proved in Appendix C. From Lemma 2, we know that the function \( \Phi(x, y) \) reduces the energy of propagated error. This is the reason why we call it error reduction function. With Lemma 1, it is straightforward to prove that whatever the PMF of \( \zeta_{u+\text{mv}_1}^{k-1} \) is,

\[
D_u^k(p) = E[\Phi^2(\zeta^{k-1}_{u+\text{mv}_1}, \tilde{f}_k^u)] \leq E[(\zeta^{k-1}_{u+\text{mv}_1})^2] = D_{u+\text{mv}_1}^{k-1},
\]

i.e., \( \alpha^u_k \leq 1 \). In other words, we have the following proposition.

**Proposition 2:** Clipping reduces propagated error, that is, \( D_u^k(p) \leq D_{u+\text{mv}_1}^{k-1} \) or \( \alpha^u_k \leq 1 \).

Proposition 2 tells us that if there is no newly induced errors in the \( k \)-th frame, transmission distortion decreases from the \((k-1)\)-th frame to the \( k \)-th frame. Fig. 2 shows the experimental result of transmission distortion propagation for ‘bus’ sequence in cif format, where the third frame is lost at the decoder and all other frames are correctly received\(^8\). The experiment setup for Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6 is: JM14.0 [27] encoder and decoder are used; the first frame is an I-frame, and the subsequent frames are all P-frames without including I-MB; For temporal error concealment, MV error concealment is the default frame copy in JM14.0 decoder due to its simplicity; residual packets can be used for decoder without the corresponding MV packets being correctly received as aforementioned; interpolation filter and deblocking filter are disabled. That is, the error reduction is caused only by the clipping noise.

In fact, if we consider the more general cases where there may be new error induced in the \( k \)-th frame, we can still prove that \( E[(\zeta^{k-1}_{u+\text{mv}_1} + \Delta^k_u)^2] \leq E[(\zeta^{k-1}_{u+\text{mv}_1})^2] \) as shown in (60) during the proof for the following corollary.

**Corollary 1:** The correlation coefficient between \( \zeta^{k-1}_{u+\text{mv}_1} \) and \( \Delta^k_u \) is non-positive. Specifically, they are negatively correlated under the condition \( \{\bar{r}, p\} \), and uncorrelated under other conditions.

Corollary 1 is proved in Appendix D. This property is very important for designing a low complexity algorithm to estimate propagation and clipping caused distortion in PTD, which is presented in the sequel paper [19].

2) Frame-Level Distortion Caused by Propagated Error Plus Clipping Noise: Define \( D^k_u(p) \) as the mean of \( \hat{D}_u^k(p) \) over all \( u \in \mathcal{V}_k \), i.e., \( D^k_u(p) \triangleq \frac{1}{|\mathcal{V}_k|} \sum_{u \in \mathcal{V}_k} D_u^k(p) \); the formula for frame-level propagation and clipping caused distortion is given in Lemma 3.

**Lemma 3:** The frame-level propagation and clipping caused distortion in the \( k \)-th frame is

\[
D^k(P) = D^{k-1} \cdot \hat{P}^k(r) + D^k(p) \cdot (1 - \hat{P}^k(r))(1 - \beta^k),
\]

where \( \hat{P}^k(r) \) is defined in (28); \( \beta^k \) is the percentage of I-MBs in the \( k \)-th frame; \( D^{k-1} \) is the transmission distortion in the \((k-1)\)-th frame.

Lemma 3 is proved in Appendix F. Define the propagation factor for the \( k \)-th frame \( \alpha^k \triangleq \frac{D^k(P)}{D^k(p)} \), then, we have \( \alpha^k = \frac{\sum_{u \in \mathcal{V}_k} \alpha^u_k D^{k-1}_{u+\text{mv}_1}}{D^{k-1} + \sum_{u \in \mathcal{V}_k} D^{k-1}_{u+\text{mv}_1}} \). As explained in Appendix F, when the number of pixels in the \((k-1)\)-th frame is sufficiently large, the sum of \( D^{k-1}_{u+\text{mv}_1} \) over all the pixels in the \((k-1)\)-th frame will converge to \( D^{k-1} \) due to the randomness of \( \text{mv}_1^u \). Therefore, we have \( \alpha^k \approx \frac{\sum_{u \in \mathcal{V}_k} \alpha^u_k D^{k-1}_{u+\text{mv}_1}}{\sum_{u \in \mathcal{V}_k} D^{k-1}_{u+\text{mv}_1}} \), which is a weighted average of \( \alpha^u_k \) with the weight being \( D^{k-1}_{u+\text{mv}_1} \). As
a result, \( D^h(P) \leq D^h(P) \) with high probability\(^9\). However, most existing works directly use \( D^h(P) = D^h(p) \) in predicting transmission distortion. This is another reason why LTI models [7], [8] underestimate transmission distortion when there is no MV error.

E. Analysis of Correlation Caused Distortion

In this subsection, we first derive the pixel-level correlation caused distortion \( D_u^b(c) \). Then, we derive the frame-level correlation caused distortion \( D_u^b(c) \).

1) Pixel-level Correlation Caused Distortion: We analyze the correlation caused distortion \( D_u^b(c) \) at the decoder in four different cases: i) for \( \mathbf{u}^k \{ r, \hat{m} \} \), both \( \hat{c}_u = 0 \) and \( \hat{c}_u^v = 0 \), so \( D_u^b(c) = 0 \); ii) for \( \mathbf{u}^k \{ r, \hat{m} \} \), \( \hat{c}_u = 0 \) and \( D_u^b(c) = 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, \hat{m} \})] \); iii) for \( \mathbf{u}^k \{ r, \hat{m} \}, \hat{c}_u = 0 \) and \( D_u^b(c) = 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, \hat{m} \})] \); iv) for \( \mathbf{u}^k \{ r, m \} \), \( D_u^b(c) = 2E[e_{u}^k \cdot \hat{c}_u + 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})] + 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})] \). From Section III-D1, we know \( \Delta_u^k \{ r, m \} = 0 \). So, we obtain

\[
D_u^b(c) = P_{u}^k \{ r, m \} \cdot (2E[e_{u}^k \cdot \hat{c}_u + 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})] + 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})]
\]

(41)

In our experiments, we find that in the trajectory of pixel \( \mathbf{u}^k \), 1) the residual \( \hat{c}_u^v \) is almost uncorrelated with the residual in all other frames \( \hat{c}_u^v \) where \( i \neq k \), i.e. their correlation coefficient is almost zero, as shown in Fig. 3; and 2) also the residual \( \hat{c}_u^k \) is almost uncorrelated with the MVCE of the corresponding pixel, i.e. \( \Delta_u^k \), and the MVCE in all previous frames, i.e. \( \Delta_u^v \), where \( i < k \), as shown in Fig. 4. Based on the above observations, we further assume that for any \( i < k \), \( \hat{c}_u^k \) is uncorrelated with \( \hat{c}_u^v \) and \( \Delta_u^v \) if \( v^v \) is not in the trajectory of pixel \( \mathbf{u}^k \), and make the following assumption.

Assumption 5: \( \hat{c}_u^k \) is uncorrelated with \( \Delta_u^k \), and is uncorrelated with both \( \hat{c}_u^v \) and \( \Delta_u^v \) for any \( i < k \).

Since \( c_k^{i-1} u + m v \) and \( c_k^{i-1} u + m v \), are the transmission reconstructed errors accumulated from all the frames before the \( k \)-th frame, \( c_k^i u + m v \) is uncorrelated with \( u + m v \) and \( u + m v \), due to Assumption 5. Thus, (41) becomes

\[
D_u^b(c) = P_{u}^k \{ m \} \cdot E[\hat{c}_u^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})] + P_{u}^k \{ r, m \} \cdot E[\hat{c}_u^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})] + 2E[\hat{c}_u^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})]
\]

(42)

However, we observe that in the trajectory of pixel \( \mathbf{u}^k \), 1) \( \hat{c}_u^k \) is correlated with \( c_u^v \) when \( i > k \) and especially when \( i = k + 1 \) there are peaks, as seen in Fig. 4; and 2) \( \Delta_u^k \) is highly correlated with \( \Delta_u^v \) as shown in Fig. 5. These interesting statistical relationships could be exploited by an error concealment algorithm, e.g. finding a concealed MV for pixel \( v^v \) with proper \( \Delta_u^v \) given \( \hat{c}_u^v \) or \( \hat{c}_u^v \), and is subject to our future study.

\(^9\)When the number of reference pixels in the \((k-1)\)-th frame is small, \( \sum_u^v \hat{c}_u^k \cdot D_{u + m v}^k \) may be larger than \( D_{u + m v}^k \) in case the reference pixels with high distortion are used more often than the reference pixels with low distortion.

\(^{10}\)Fig. 3, Fig. 4, Fig. 5 and Fig. 6 are plotted for low motion sequence, e.g. ‘foreman’, and high motion sequence, e.g. ‘stefan’, in cif format. All other sequences show the similar statistics.

As mentioned in Section III-D1, for P-MBs with SDP in H.264, \( P_{u}^k \{ r, m \} = 0 \). So, (42) becomes

\[
D_u^b(c) = 2P_{u}^k \{ m \} \cdot E[\hat{c}_u^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})]
\]

(43)

Note that in the more general case that \( P_{u}^k \{ r, m \} \neq 0 \), Eq. (43) can be used as an approximation since in (42), \( E[\hat{c}_u^k \cdot \Delta_u^k \{ r, m \}] \) is much smaller than \( E[\hat{c}_u^k \cdot \hat{c}_u^v + \Delta_u^k \{ r, m \}] \). For MBs without SDP, since \( P_{u}^k \{ r, m \} = P_{u}^k \{ r, m \} = 0 \) and \( P_{u}^k \{ r, m \} = P_{u}^k \{ m \} = P_{u}^k \{ m \} \) as mentioned in Section III-D1, (41) can be simplified to

\[
D_u^b(c) = 2P_{u}^k \{ r \} \cdot (2E[e_{u}^k \cdot \Delta_u^k \{ r, m \}] + 2E[e_{u}^k \cdot (\hat{c}_u^v + \Delta_u^k \{ r, m \})]
\]

(44)

Under Assumption 5, (44) reduces to (43).

Define \( \lambda_u^k = \frac{E[\hat{c}_u^k \cdot \hat{c}_u^v + \Delta_u^k \{ r, m \}]}{E[\hat{c}_u^k \cdot \Delta_u^k \{ r, m \}]} \), \( \lambda_u^k \) is a corratio, that is, the ratio of the correation between MVCE and concealed reference pixel value at the decoder, to the correation between MVCE and concealed reference pixel value at the encoder. \( \lambda_u^k \) quantifies the effect of the correlation between the MVCE and propagated error on transmission distortion.

Note that although we do not know the exact value of \( \lambda_u^k \) at the encoder, its range is characterized by XEP of all pixels
in the trajectory \( T \), which passes through the pixel \( u^k \), as
\[
\prod_{i=1}^{k-1} P_{T(i)} \{ \tilde{r}, \tilde{m} \} \leq \lambda_u^k \leq 1,
\]
where \( T(i) \) is the reference pixel position in the \( i \)-th frame for the trajectory \( T \). For example, \( T(k-1) = u^k + mv_u^k \) and \( T(k-2) = T(k-1) + mv_u^{k-1} \). The left inequality in (45) holds in the extreme case that any error in the trajectory will cause \( \xi_u^k \) and \( \tilde{f}_u^{k-1} \) to be uncorrelated, which is usually true for high motion video. The right inequality in (45) holds in another extreme case that all errors in the trajectory do not affect the correlation between \( \xi_u^k \) and \( \tilde{f}_u^{k-1} \), that is \( E[\xi_u^k \cdot \tilde{f}_u^{k-1}] \approx E[\xi_u^k] \cdot E[\tilde{f}_u^{k-1}] \), which is usually true for low motion video. The details on how to estimate \( \lambda_u^k \) is presented in the sequel paper [19].

Using the definition of \( \lambda_u^k \), we have the following lemma.

**Lemma 4:**
\[
D_u^k(c) = (\lambda_u^k - 1) \cdot D_u^k(m).
\]

Lemma 4 is proved in Appendix G.

If we assume \( E[\xi_u^k] = 0 \), we may further derive the correlation coefficient between \( \xi_u^k \) and \( \tilde{f}_u^{k-1} \). Denote \( \rho \) as their correlation coefficient, from (70), we have
\[
\rho = \frac{E[\xi_u^k \cdot \tilde{f}_u^{k-1}] - E[\xi_u^k] \cdot E[\tilde{f}_u^{k-1}]}{\sigma_{\xi_u^k} \cdot \sigma_{\tilde{f}_u^{k-1}}} = \frac{E[(\xi_u^k)^2]}{2 \cdot \sigma_{\xi_u^k} \cdot \sigma_{\tilde{f}_u^{k-1}}} = \frac{\sigma_{\xi_u^k}}{2 \cdot \sigma_{\xi_u^k}}.
\]

Similarly, it is easy to prove that the correlation coefficient between \( \xi_u^k \) and \( \tilde{f}_u^{k-1} \) is \( \frac{\sigma_{\xi_u^k}}{2 \cdot \sigma_{\xi_u^k}} \). This agrees well with the experimental results shown in Fig. 6. Via the same derivation process, one can obtain the correlation coefficient between \( \tilde{e}_u^{k-1} \) and \( \tilde{f}_u^{k-1} \), and between \( \tilde{e}_u^k \) and \( \tilde{f}_u^k \). One possible application of these correlation properties is error concealment with partial information available.

2) **Frame-Level Correlation Caused Distortion:** Denote \( V_k^i \{ m \} \) the set of pixels in the \( i \)-th MV packet of the \( k \)-th frame. From (19), (71) and Assumption 4, we obtain
\[
D_u^k(c) = \frac{E[(\xi_u^k)^2]}{|V|} \sum_{u \in V_k^i} (\lambda_u^k - 1) \cdot P_u^k(m) = \frac{E[(\xi_u^k)^2]}{|V|} \sum_{i=1}^{N_u^i} \{ P_i^k(m) \sum_{u \in V_k^i} (\lambda_u^k - 1) \}.
\]

Define \( \lambda_u^k \triangleq \frac{1}{|V|} \sum_{u \in V_k^i} \lambda_u^k \cdot \frac{1}{N_u^i} \sum_{u \in V_k^i} (\lambda_u^k - 1) \lambda_u^k \) will converge to \( \lambda_u^k \) for any packet that contains a sufficiently large
number of pixels. By rearranging (48), we obtain
\[ D^k(c) = \frac{E[(\xi^k)^2]}{|\mathcal{V}|} \sum_{i=1}^{N^k(m)} \{ P^k_i(m) \cdot N^k_i(m) \cdot (\lambda^k - 1) \} \]
\[ = (\lambda^k - 1) \cdot E[(\xi^k)^2] \cdot \tilde{P}^k(m). \] (49)

From (32), we know that \( E[(\xi^k)^2] \cdot \tilde{P}^k(m) \) is exactly equal to \( D^k(m) \). Therefore, (49) is further simplified to
\[ D^k(c) = (\lambda^k - 1) \cdot D^k(m). \] (50)

F. Summary

In Section III-A, we decomposed transmission distortion into four terms; we derived a formula for each term in Sections III-B through III-E. In this section, we combine the formulae for the four terms into a single formula.

1) Pixel-Level Transmission Distortion:

Theorem 1: Under single-reference motion compensation, the PTD of pixel \( u^k \) is
\[ D^k_u = D^k_u(r) + \lambda^k \cdot D^k_u(m) + P^k_u(r, m) \cdot D^{k-1}_{u+\text{mv}_u} + P^k_u(r, \bar{m}), \] (51)

Proof: (51) can be obtained by plugging (23), (31), (34), and (71) into (14).

Corollary 2: Under single-reference motion compensation and no SDP, (51) is simplified to
\[ D^k_u = \tilde{P}^k_u \cdot (E[(\xi^k)^2] + \lambda^k \cdot E[(\xi^k)^2] + D^{k-1}_{u+\text{mv}_u}) + (1 - \tilde{P}^k_u) \cdot \alpha^k \cdot D^{k-1}_{u+\text{mv}_u}. \] (52)

2) Frame-Level Transmission Distortion:

Theorem 2: Under single-reference motion compensation, the FTD of the \( k \)-th frame is
\[ D^k = D^k(r) + \lambda^k \cdot D^k(m) + \tilde{P}^k(r) \cdot D^{k-1} + (1 - \tilde{P}^k(r)) \cdot D^k(p) \cdot (1 - \beta^k) \cdot D^{k-1}. \] (53)

Proof: (53) can be obtained by plugging (29), (32), (40) and (50) into (15).

Corollary 3: Under single-reference motion compensation and no SDP, the FTD of the \( k \)-th frame is simplified to
\[ D^k = \tilde{P}^k \cdot (E[(\xi^k)^2] + \lambda^k \cdot E[(\xi^k)^2] + D^{k-1}) + (1 - \tilde{P}^k) \cdot \alpha^k \cdot D^{k-1}. \] (54)

Following the same deriving process, it is not difficult to obtain the distortion prediction formulæ under multi-reference case. Due to the space limit, in this paper we just present the formulæ for distortion estimation under single-reference case. Interested reader may refer to Ref. [22] for the analysis of multi-reference case. In Ref. [22], we also identify the relationship between our result and existing models, and specify the conditions, under which those models are accurate.

IV. CONCLUSION

In this paper, we derived the transmission distortion formulæ for wireless video communication systems. With consideration of spatio-temporal correlation, nonlinear codec and time-varying channel, our distortion prediction formulæ improve the accuracy of distortion estimation from existing works. Besides that, our formulæ support, for the first time, the following capabilities: 1) prediction at different levels (e.g., pixel/frame/GOP level), 2) prediction for multi-reference motion compensation, 3) prediction under SDP, 4) prediction under arbitrary slice-level packetization with FMO mechanism, 5) prediction under time-varying channels, 6) one unified formula for both I-MB and P-MB, and 7) prediction for both low motion and high motion video sequences. In addition, this paper also identified two important properties of transmission distortion for the first time: 1) clipping noise, produced by non-linear clipping, causes decay of propagated error; 2) the correlation between motion vector concealment error and propagated error is negative, and has dominant impact on transmission distortion, among all the correlations between any two of the four components in transmission error.

In the sequel paper [19], we use the formulæ derived in this paper to design algorithms for estimating pixel-level and frame-level transmission distortion and apply the algorithms to video codec design; we also verify the accuracy of the formulæ derived in this paper through experiments; the application of these formulæ shows superior performance over existing models.
ACKNOWLEDGMENTS

This work was supported in part by an Intel gift, the US National Science Foundation under grant CNS-0643731 and ECCS-1002214. The authors would like to thank Jun Xu and Qian Chen for many fruitful discussions related to this work and suggestions that helped to improve the presentation of this paper. The authors would also like to thank the anonymous reviewers for their valuable comments to improve the quality of this paper.

APPENDIX

A. Proof of Lemma 1

Proof: From (10) and (12), we obtain
\[ \hat{j}_u \geq \tilde{c}_u - \xi, \quad \tilde{c}_u - \xi - \tilde{c}_u \]
\[ \Delta_u = (f_u - \tilde{c}_u) - (\hat{j}_u - \xi - \tilde{c}_u). \]

Since \( y \geq 1 \), we obtain \( (y - \gamma_i)^2 < x^2 \) when \( x > y - \gamma_i \). Similarly, since \( y \leq \gamma_h \), we obtain \( (y - \gamma_h)^2 < x^2 \) when \( x < y - \gamma_h \). Therefore, Fig. 7 shows a pictorial example of the case that \( \gamma_h = 255 \).

D. Proof of Corollary 1

Proof: From (55), we obtain \( \Delta_u \{ \hat{p} \} = (f_u - \tilde{c}_u - \tilde{c}_u) - \Gamma(f_u - \tilde{c}_u - \tilde{c}_u) \). Together with Lemma 5, which is presented and proved in Appendix E, we have \( \gamma \leq \hat{j}_u - \tilde{c}_u - \tilde{c}_u \leq \gamma_h \).

From Lemma 2, we have \( \Phi^2(x, y) \leq x^2 \) for any \( \gamma \leq y \leq \gamma_h \); therefore, \( E[\Phi^2(f_u - \tilde{c}_u - \tilde{c}_u)] \leq E[(\tilde{c}_u - \tilde{c}_u)^2] \).

By expanding \( E[\Delta_u \{ \hat{p} \}] \), we obtain
\[ E[\Delta_u \{ \hat{p} \}] \leq \frac{1}{2} E[(\Delta_u)^2] \leq 0. \]

The physical meaning of (61) is that \( \tilde{c}_u - \tilde{c}_u \) and \( \tilde{c}_u \) are negatively correlated if \( \Delta_u \neq 0 \). Since \( \Delta_u \{ \hat{p} \} = 0 \) as noted in Section III-D, and \( \Delta_u \{ \hat{p} \} \neq 0 \) as proved in Lemma 5, we know that \( \Delta_u \neq 0 \) is valid only for the error events \( \{ \hat{r}, m, p \} \) and \( \{ \hat{r}, m, p \} \), and \( \Delta_u = 0 \) for any other error event. In other words, \( \tilde{c}_u - \tilde{c}_u \) and \( \tilde{c}_u \) are negatively correlated under the condition \( \{ \hat{r}, m, p \} \), and they are uncorrelated under other conditions.

E. Lemma 5 and Its Proof

Before presenting the proof, we first give the definition of Ideal Codec.

Definition 1: Ideal Codec: both the true MV and concealed MV are within the search range, and the position pointed by the true MV, i.e., \( u + \tilde{m}_v \), is the best reference pixel, under the MMSE criteria, for pixel \( u_v \) within the whole search range \( y_{SR} \), that is, \( v = \arg \min \{ \{(f_u - \tilde{f}_v)^2\} \} \).
To prove Corollary 1, we need to use the following lemma.

Lemma 5: In an ideal codec, \( \Delta^k_u(\bar{p}) = 0 \). In other words, if there is no propagated error, the clipping noise for the pixel \( u^k \) at the decoder is always zero no matter what kind of error event occurs in the \( k \)-th frame.

Proof: In an ideal codec, we have \( (\hat{e}^k_u)^2 = (\hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k)^2 = (\hat{f}^k_u - \hat{f}^{k-1}_u)^2 \). Due to the spatial and temporal continuity of the natural video, we can prove by contradiction that in an ideal codec \( \hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k \) and \( \hat{f}^k_u - \hat{f}^{k-1}_u \) have the same sign, that is either

\[
\hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k \geq \hat{e}^k_u \geq 0, \quad \text{or} \quad \hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k \leq \hat{e}^k_u \leq 0.
\]

(62)

If the sign of \( \hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k \) and \( \hat{f}^k_u - \hat{f}^{k-1}_u \) is not the same, then due to the spatial and temporal continuity of the input video, there exists a better position \( v \in \gamma^k \) between \( mv^k_u \) and \( mv^{k+1}_u \), and therefore in the search range, so that \( (\hat{e}^k_v)^2 \geq (\hat{f}^k_v - \hat{f}^{k-1}_v)^2 \). In this case, encoder will choose \( v \) as the best reference pixel within the search range. This contradicts the assumption that the best reference pixel is \( u + mv^k_u \) within the search range.

Therefore, from (62), we obtain

\[
\hat{f}^k_u \geq \hat{f}^{k-1}_u + \hat{e}^k_u \geq \hat{f}^{k-1}_u + u_{mv}^k, \quad \text{or} \quad \hat{f}^k_u \leq \hat{f}^{k-1}_u + \hat{e}^k_u \leq \hat{f}^{k-1}_u + u_{mv}^k.
\]

(63)

Since both \( \hat{f}^k_u \) and \( \hat{f}^{k-1}_u + u_{mv}^k \) are reconstructed pixel value, they are within the range \( \gamma^k \). From (63), we have \( \gamma^k \geq \hat{f}^k_u + \hat{e}^k_u = \hat{f}^{k-1}_u + \hat{e}^k_u \), thus \( \Gamma(\hat{f}^{k-1}_u + \hat{e}^k_u) = \hat{f}^{k-1}_u + \hat{e}^k_u \). As a result, we obtain \( \Delta^k_u(\bar{r}, m, \bar{p}) = (\hat{f}^{k-1}_u + \hat{e}^k_u - \hat{f}^{k-1}_u + \hat{e}^k_u) \).

Since \( \Delta^k_u(\bar{r}, m, \bar{p}) = \Delta^k_u = 0 \), and from Section III-D1, we know that \( \Delta^k_u(\bar{r}, \bar{p}) = 0 \), hence we obtain \( \Delta^k_u(\bar{p}) = 0 \).

Remark 1: Note that Lemma 5 is proved under the assumption of pixel-level motion estimation. In a practical encoder, block-level motion estimation is adopted with the criterion of minimizing the MSE of the whole block, e.g., in H.263, or minimizing the cost of residual bits and MV bits, e.g., in H.264. Therefore, some reference pixels in the block may not be the best reference pixel within the search range. On the other hand, Rate Distortion Optimization (RDO) as used in H.264 may also cause some reference pixels not to be the best reference pixels. However, the experiment results for all the test video sequences show that the probability of \( \Delta^k_u(\bar{r}, m, \bar{p}) \neq 0 \) is negligible.

F. Proof of Lemma 3

Proof: For P-MBS with SDP, from (18) and (34) we obtain

\[
D^k(P) = \frac{1}{|V|} \sum_{u \in V^k} (P^k_u(r, m) \cdot D^{k-1}_{u+mv}u^k) + \frac{1}{|V|} \sum_{u \in V^k} (P^k_u(r, m) \cdot D^{k-1}_{u+mv}u^k)
\]

(64)

Denote \( V^k(r, m) \) the set of pixels in the \( k \)-th frame with the same XEP \( P^k(r, m) \); denote \( N^k(r, m) \) the number of pixels in \( V^k(r, m) \); denote \( N^k(r, m) \) the number of sets with different XEP \( P^k(r, m) \) in the \( k \)-th frame. Although \( D^{k-1}_{u+mv}u^k \) may be very different for different pixels \( u + mv^k \) in the \( (k-1) \)-th frame, e.g. under a fast fading channel with FMO mechanism, for large \( N^k(r, m) \), we have \( \frac{1}{N^k(r, m)} \sum_{u \in V^k(r, m)} D^{k-1}_{u+mv}u^k \) converges to \( D^{k-1} \). Therefore,

\[
\frac{1}{|V|} \sum_{u \in V^k} (P^k_u(r, m) \cdot D^{k-1}_{u+mv}u^k) = \frac{1}{|V|} \sum_{i=1}^{N^k(r, m)} \sum_{u \in V^k(r, m)} D^{k-1}_{u+mv}u^k = D^{k-1} \cdot \bar{P}^k(r, m),
\]

(65)

where \( \bar{P}^k(r, m) = \frac{1}{|V|} \sum_{i=1}^{N^k(r, m)} P^k_i(r, m) \cdot N^k_i(r, m) \).

Following the same process, we obtain the first term in the right-hand side in (64) as \( D^{k-1} \cdot \bar{P}^k(r, m) \), where \( P^k(r, m) = \frac{1}{|V|} \sum_{i=1}^{N^k(r, m)} P^k_i(r, m) \cdot N^k_i(r, m) \); and

\[
\frac{1}{|V|} \sum_{u \in V^k} (P^k_u(r, m) \cdot D^k(p)) = \frac{1}{|V|} \sum_{i=1}^{N^k(r, m)} \sum_{u \in V^k(r, m)} D^k(p).
\]

(66)

For large \( N^k(r, m) \), we have \( \frac{1}{N^k(r, m)} \sum_{u \in V^k(r, m)} D^k(p) \) converges to \( D^k(p) \), so the third term in the right-hand side in (64) is \( D^k(p) = (1 - \bar{P}^k(r)) \).

Note that \( P^k(r, m) + P^k_i(r, m) = P^k(r) \) and \( N^k_i(r, m) = N^k_i(r, m) \). So, we obtain

\[
D^k(P) = D^{k-1} \cdot \bar{P}^k(r) + D^k(p) \cdot (1 - \bar{P}^k(r))
\]

(67)

For P-MBS without SDP, it is straightforward to acquire (67) from (35). For I-MBS, from (36), it is also easy to obtain \( D^k(P) = D^{k-1} \cdot \bar{P}^k(r) \). So, together with (67), we obtain (40).

G. Proof of Lemma 4

Proof: Using the definition of \( \lambda_u^k \), (43) becomes

\[
D^k_u(c) = 2P^k_u(m) \cdot (1 - \lambda_u^k) \cdot E[\hat{f}^k_u - \hat{f}^{k-1}_u + u_{mv}^k].
\]

(68)

Under the condition that the distance between \( mv_u^k \) and \( mv_u^{k+1} \) is small, for example, inside the same MB, the statistics

\[11\text{According to the definition, for any given } u \in \gamma^k, D^{k-1} \text{ is an expected value, that is, it is not a random variable. However, due to the randomness of } mv_u^k, \text{ each pixel in the } k \text{ - } 1 \text{ - th frame can be used as a reference for multi pixels in the } k \text{-th frame. In other words, }\]

\[N^k(r, m) \sum_{u \in V^k(r, m)} D^{k-1}_{u+mv}u^k \text{ can be described as simple random sampling with replacement (SRSWR) and take their average. On the other hand, according to (64), } D^{k-1}_{u+mv}u^k \text{ the mean of } D^{k-1}_{u+mv}u^k \text{ over all } u \in \gamma^k \text{. Therefore, using the theorem in Ref. [28], it is easy to prove that expectation of } D^{k-1}_{u+mv}u^k \text{ is exactly equal to } D^{k-1} \text{. And using the Theorem 5.5.2 in Ref. [28], it is also easy to prove that } \frac{1}{N^k(r, m)} \sum_{u \in V^k(r, m)} D^{k-1}_{u+mv}u^k \text{ converges in probability to } D^{k-1} \text{. Note again that the randomness of } D^{k-1}_{u+mv}u^k \text{ is caused by } mv_u^k.\]
of $f_{uk+1}^{k-1}$ and $f_{uk+1}^{k-1}$ loss patterns are almost the same. Therefore, we may assume $E[(f_{uk+1}^{k-1})^2] = E[(f_{uk+1}^{k-1})^2]$. Since $\xi_u = f_{uk+1}^{k-1} - f_{uk+1}^{k-1}$, we have

$$E[(f_{uk+1}^{k-1})^2] = E[(f_{uk+1}^{k-1})^2] = E[(\xi_u + f_{uk+1}^{k-1})^2],$$

and therefore

$$E[\hat{\xi}_u f_{uk+1}^{k-1} + f_{uk+1}^{k-1}] = -\frac{E[(\hat{\xi}_u)^2]}{2}.$$

Note that following the same derivation process, we can prove $E[\xi_u f_{uk+1}^{k-1}] = \frac{E[(\xi_u)^2]}{2}$. Therefore, (68) can be simplified as

$$D_u^k(c) = (\lambda^k - 1) \cdot E[(\xi_u)^2] \cdot P^k_u(m).$$

From (31), we know that $E[(\hat{\xi}_u)^2] \cdot P^k_u(m)$ is exactly equal to $D_u^k(m)$. Therefore, (71) is further simplified to (46).

REFERENCES


