LOSS-COMPENSATED REFERENCE FRAME OPTIMIZATION FOR MULTI-PATH VIDEO STREAMING

Gene Cheung

Hewlett-Packard Laboratories, Japan
Takaido Office Bldg.#3
3-8-13, Takaido-Higashi, Suginami-ku
Tokyo, 168-0072 Japan
Email: gene-cs.cheung@hp.com

Wai-tian Tan

Hewlett-Packard Laboratories, Palo Alto
1501 Page Mill Rd.
Palo Alto, CA
USA
Email: wai-tian.tan@hp.com

ABSTRACT

Recent video coding standards such as H.264 offer the flexibility to select reference frames during motion estimation for predicted frames. In this paper, by tracking loss compensation during distortion minimization, we improve upon an earlier proposal to jointly select reference frame, level of QoS and transmission path for each video frame in a multi-path streaming scenario. An algorithm that efficiently calculates the loss compensation value of an earlier correctly decodeable frame during error concealment is presented. Results show significant streaming quality improvement when loss compensation is used.

1. INTRODUCTION

The goal of this paper is to extend and improve upon the proposed reference frame selection scheme for multi-path video streaming in [1] by tracking loss compensation during optimization. New video coding standards such as H.264 [2] offer many coding flexibilities to facilitate better coding and streaming performance. An example is *reference picture selection* (RPS), where each predicted frame can choose among a number of frames for motion estimation. Often at the cost of lower coding efficiency, RPS can be used to improve error resilience of the video stream by controlling the effect of error propagation due to packet loss. Given the available RPS feature of video coding, we consider in this paper reference frame selection in the context of multi-path streaming over QoS networks.

There are many possible flavors of QoS network, including network-layer QoS such as DiffServ, and application-layer QoS as achieved by applying forward error correction (FEC) codes of different strengths. While our formulation in this paper can be applied for both, we only present results for the latter for lack of space.

By multi-paths, we mean two (or more) delivery paths are simultaneously available for packet delivery to end-hosts during a streaming session. Multi-paths can exist for wireless cellular networks if mobile terminals can simultaneously connect to two nearby base-stations. We additionally assume the application requires very low network delivery delay, to the extent that it cannot tolerate even one end-to-end packet retransmission. This can happen due to a small playback buffer at the client side, and a relatively large transmission delay, e.g., that of wireless links such as 3G cellular links. Both conditions mean that any retransmitted packet will miss its playback deadline and hence be rendered useless.

Given the described RPS feature of video coding and the network streaming scenario, we address the following problem: for each predicted video frame, how to select: i) an appropriate reference frame for motion estimation, and ii) a QoS level and transmission path for packet delivery, such that the overall streaming performance is optimized? In [1], an optimization algorithm based on integer rounding techniques and with algorithmic complexity and worst-case error bounds is presented. In this paper, we further improve upon [1] by modeling the loss compensation process during optimization so that the importance of a particular correctly decodeable frame can be more accurately assessed. The rest of the paper is organized as follows. After discussing related work in Section 2, we formulate the optimization problem and provide a solution in Sections 3 and 4, respectively. Results and conclusion are presented in Section 5 and 6, respectively.

2. PREVIOUS WORK

H.264 [2] is a new video coding standard that has demonstrated superior coding performance over previous standards such as MPEG-4 and H.263 over a range of bit rates. As part of the new standard definition is the flexibility of using any arbitrary frame to perform motion-estimation, originally introduced as Annex N in H.263+ and later as Annex U in H.263++. Early work on optimizing streaming quality us-

0-7803-9332-5/05/$20.00 ©2005 IEEE

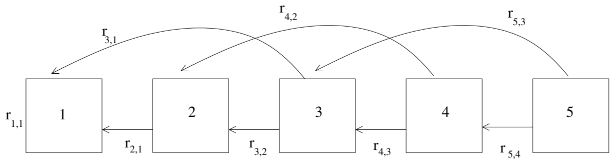

Fig. 1. Directed Acyclic Graph Source Model

ing reference frame selection includes [3, 4, 5]. Our work differs from previous works by employing a complexity-scalable optimization procedure and also applying optimization to jointly perform reference frame (RF) and transmission path (TP) selection.

Our previous work [1] used an integer rounding-based algorithm to jointly select reference frame / QoS selection for multi-path streaming. In this paper, we further track the loss compensation process to more accurately determine value of a correctly decodeable frame.

3. PROBLEM FORMULATION

We first outline the source and network models. We then discuss how loss compensation is tracked given our source and network models. We conclude the section by discussing the objective function and formalizing the optimization.

3.1. Directed Acyclic Graph Source Model

Consider an M-frame video sequence with one intra-coded frame (I-frame) followed by $M-1$ inter-coded frames (P-frames). Denote the frames by F_i, $i \in \{1, \ldots, M\}$. We model the decoding dependencies of the video using a directed acyclic graph (DAG) $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}, |\mathcal{V}| = M$, and edge set $\mathcal{E}$. Each frame F_i, represented by a node $i \in \mathcal{V}$, has a set of outgoing edges $e_{i,j} \in \mathcal{E}$ to nodes j's. F_i can use F_j as reference iff $\exists e_{i,j} \in \mathcal{E}$. We define $x_{i,j}$ to be the binary variable indicating whether F_i uses F_j as reference. Equivalently, we define $x_{i,j}$ given i:

$$x_{i,j} = \begin{cases} 1 & \text{if } F_i \text{ uses } F_j \text{ as RF} \qquad \forall j \in \{\mathcal{V} | e_{i,j} \in \mathcal{E}\} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Since a P-frame can have only one reference frame, we have the following *RF constraint*:

$$\sum_{\{j | e_{i,j} \in \mathcal{E}\}} x_{i,j} = 1 \qquad \forall i \in \mathcal{V}, i \neq 1 \tag{2}$$

We assume that only frames in the past are used for reference, i.e. $\forall e_{i,j} \in \mathcal{E}, i > j$. Further, since it is impractical to use a reference frame too far in the past, we limit the number of candidate reference frames for any given predicted frame F_i to $E_{\max} \ll |\mathcal{V}|$. We also assume only frame 1 is intra-coded, and hence $\nexists e_{1,j} \in \mathcal{E}$. An example DAG model for a 5-frame sequence is shown in Figure 1 with $E_{\max} = 2$.

Associated with edge $e_{i,j}$, we denote by $r_{i,j} \in \mathcal{I}$ the integer number of bytes in frame i when frame j is used as reference. The byte size of the starting I-frame is $r_{1,1}$. Since the size of F_i depends on chosen F_j, as well as RF for F_j and so on, $r_{i,j}$ is an approximate.

In the event of decoding failure of frame F_i due to irrecoverable packet loss, we assume the most recently correctly decoded F_j is used for error concealment. The loss compensation effort of using F_j to reconstruct F_i results in approximate reconstructed value $d_{i-j} < 1$, which we assumed to depend only on $i - j$ and hence is time-invariant to both i and j. We will assume *rate matrix* $\mathbf{r}$ of size $O(M^2)$ and *partial construction vector* $\mathbf{d}$ of length $O(M)$ are computed *a priori* as input to the optimization.

3.2. Network Model

We assume a set of QoS levels $\mathcal{Q} = \{0, 1, \ldots, Q\}$ is available for all transmission paths. We allow each frame F_i to select a QoS level $q_i \in \mathcal{Q}$ and a transmission path $t_i \in \{0, 1\}$. QoS service level $q_i = 0$ denotes the case when F_i is not transmitted at all. For given observable network condition, QoS level q_i, transmission path t_i and frame size $r_{i,j}$ will induce a frame delivery success probability $p_{t_i}(q_i, r_{i,j}) \in \mathcal{R}$, where $0 \leq p_{t_i}(q_i, r_{i,j}) \leq 1$. Generally, $p()$ depends on $r_{i,j}$ because a large frame size will likely negatively impact the delivery success probability of the entire frame as more data is pushed through the network.

3.2.1. Network Resource Constraint

We impose constraints on the amount of resource we can use, which in this case is the aggregate ability to protect the M-frame sequence per transmission path from network losses using QoS. Assuming a QoS assignment q_i results in a cost of $c(q_i) \in \mathcal{R}$ per byte, the constraints for path 0 and path 1 are respectively:

$$\begin{aligned} \sum_{i=1}^{M} \sum_{\{j | e_{i,j} \in \mathcal{E}\}} x_{i,j} \, c(q_i) \, (1 - t_i) \, r_{i,j} \quad &\leq \quad \bar{R}_0 \\ \sum_{i=1}^{M} \sum_{\{j | e_{i,j} \in \mathcal{E}\}} x_{i,j} \, c(q_i) \, t_i \, r_{i,j} \quad &\leq \quad \bar{R}_1 \end{aligned} \tag{3}$$

where there is a separate constraint per path, and $c(q_i)$ is the overhead in channel coding given QoS level q_i, and constraint parameters $\bar{R}_0$ and $\bar{R}_1$ are given, and can be obtained from congestion control algorithms so that the total output bytes for M-frame time for path 0 and 1 do not exceed $\bar{R}_0$ and $\bar{R}_1$ bytes, respectively.

3.3. Tracking Loss Compensation Value of An Earlier Correct Decodeable Frame

To estimate the likelihood of using an earlier F_b for loss compensation for F_j given a set of RF, QoS and path selection, we need to derive the probability that frame sequence

```
algorithm $L_b(i,j)$
1. $\forall k \le j$, color $F_k$ white.
2. $\forall k \preceq b$, color $F_k$ black.
3. $\mathcal{S}_0 := \{F_i, \ldots, F_j\}$.
4. $\mathcal{S} := \mathcal{S}_0$.
5. $w_k := 0, \ \forall F_k \in \mathcal{S}$.
6. $w_k := 1, \ \forall F_k \notin \mathcal{S}$.
7. $fin := 1$.
8. while ($\mathcal{S} \neq$ empty set)
9. {   let $l$ be the largest frame number of elements in $\mathcal{S}$.
10.       $\forall k \mid x_{l,k} = 1$,
11.       {  $w_k *= (1 - p_{t_l}(q_l, r_{l,k})) + p_{t_l}(q_l, r_{l,k}) * w_l$.
12.           remove $F_l$ from $\mathcal{S}$.
13.           if ($F_k$ is black),
14.               $fin := fin * w_k$.
15.               $w_k := 1$.
16.           if (($F_k$ is white) & ($F_k \notin \mathcal{S}$)),
17.               sort $F_k$ into $\mathcal{S}$.
18. }  }
19. return $fin$.
```

Fig. 2. Decoding Failure Probability $L_b(i,j)$

F_{b+1} to F_j cannot be correctly decoded, given correct decoding of frame F_b, denoted as $L_b(b+1,j)$. Each frame F_s is correctly decoded iff F_s and all frames F_t's it depends on for correct decoding are delivered drop-free. We write $t \preceq s$ if frame s depends on frame t.

For example, to compute $L_3(4,5)$ for frame dependence depicted in Figure 3a, we assume that frame 3 is correctly decoded, i.e., frames 1 and 3 are received. Given that, frames 4 and 5 cannot be decoded with respective probability of $(1-p_2)+p_2*(1-p_4)$ and $1-p_5$. Given conditional independence, we have $L_3(4,5) = [(1-p_2)+p_2*(1-p_4)](1-p_5)$.

Generally, calculating $L_b(i,j)$ is non-trivial. We compute $L_b(i,j)$ using algorithm outlined in Figure 2. We distinguish between three classes of frames. The *black* frames, which are known to have been received correctly (line 2), the *failure set* $\mathcal{S}_0$, which contains frames that must not be decodeable (line 3), and the remaining, about which we do not make assumptions. In summary, the algorithm calculates weights w_f's for non-black frames f's and propagates them upwards to the black nodes, where w_f represents the decoding failure probability for all frames in $\mathcal{S}_0$ that are dependent on f. Weights are initialized to 1 except for $\mathcal{S}_0$. During successive iteration, the weight of a parent node k (line 10) of the last node l in ordered $\mathcal{S}$ (line 9) is updated (line 11):

$$w_k := w_k * [(1 - p_{t_l}(q_l, r_{l,k})) + p_{t_l}(q_l, r_{l,k})\, w_l] \,. \qquad (4)$$

The terms in bracket represent the contribution of decoding failure probability of frames in $\mathcal{S}_0$ that stem from node l. For a non-black node k, such contributions from all its immediate dependent l's are eventually multiplied together, before node k's contribution itself is folded up to its parent. This is due to the fact that dependent l's of k must have relation $k < l$, and elements in $\mathcal{S}$ are selected in reverse order (line 9). With correct decoding probability of 1, black nodes indicate the termination of a particular tree branch of failure probability calculation, and hence the branch contribution is folded to the final answer fin instead (line 14). Figure 3 illustrates three iterations of the algorithm in Figure 2. With one more iteration, the algorithm terminates with the correct answer.

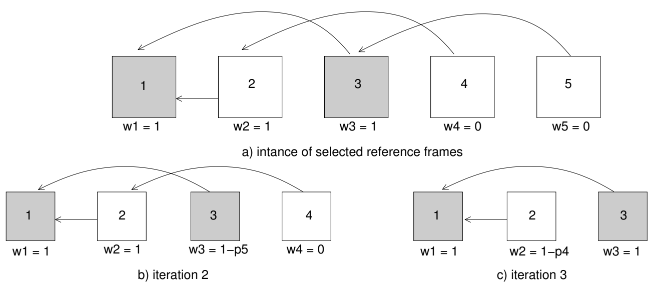

Fig. 3. Examples of $L_3(4,5)$, where p_k denotes probability that frame k is received.

We can bound the complexity of $L_b(i,j)$ as follows. The coloring and the initialization of weights (line 1-7) are bounded by the number of nodes under consideration, $O(j)$. We can bound the number of iterations in the loop (line 8-18) by looking at the evolution of the frame number of the last element in $\mathcal{S}$, which always decrements by at least 1 since $\forall e_{i,j} \in \mathcal{E}, j < i$. Hence the number of iterations is bounded by $O(j)$. Therefore we can conclude that $L_b(i,j)$ is $O(j) \le O(M)$.

3.4. Objective Function

Having defined $L_b(i,j)$, we now define our objective function, which we select to be the *expected construction* of the M-frame sequence at the decoder. Mathematically, it can be written as:

$$\max_{\{x_{i,j}\},\{q_i\},\{t_i\}} \sum_{i=1}^{M} P_i * [1 + D_i] \qquad (5)$$

where, P_i and D_i are defined as:

$$\begin{aligned} P_i &= \prod_{\{j \mid j \preceq i\}} \sum_{\{k \mid e_{j,k} \in \mathcal{E}\}} x_{j,k}\, p_{t_j}(q_j, r_{j,k}) \\ D_i &= \sum_{j=i+1}^{M} d_{j-i} L_i(i+1,j) \end{aligned} \qquad (6)$$

where P_i is the probability that F_i is decoded correctly, and D_i is the concealment benefit (loss compensation) of using F_i for subsequence frames $F_j, i+1 \le j \le M$ given F_i is decoded correctly.

Given pre-computed rate matrix $\mathbf{r}$, partial construction vector $\mathbf{d}$, delivery success probability function $p_{t_i}(q_i, r_{i,j})$ and cost function $c(q_i)$, our goal is to find variables $\{x_{i,j}\}$, $\{q_i\}$ and $\{t_i\}$ that maximize (6) while satisfying the integer constraint (1), the RF constraint (2) and the network resource constraints (3). We call this the *RF / QoS / Path selection problem with loss compensation* (RQP selection with loss compensation).

4. DYNAMIC PROGRAMMING ALGORITHM PLUS LOCAL REFINEMENTS

To solve RQP selection with loss compensation, we first use optimization algorithm developed in [1] to solve (5) without loss compensation (D_i's are all zeros). We then perform local refinements to further improve the obtained solution, this time using (5) with loss compensation as objective. Specifically, we define an *upgrade* of F_i as a single perturbation in value of reference frame selection variables $x_{i,j}$'s or QoS variable q_i. Correspondingly, the cost function in (5) increases by γ_i and the bit expenditure increases by Δ_i. The *optimum* upgrade of F_i is the single variable perturbation with the largest γ_i/Δ_i. If Δ_i causes the solution to outspend the bit budgets R_1 or R_0, we seek a series of F_j's with *optimum downgrades* (smallest decrease in objective value divided by decrease of bit expenditure) until the budget is met again. The sequence of one optimum upgrade of F_i followed by optimum downgrades of F_j's is called an *upgrade bundle* of F_i. The net-benefit will be the initial increase of upgrade of F_i minus the decreases of downgrades of F_j's. We find the most beneficial upgrade bundle of all F_i's, perform the refinements, and repeat until no beneficial upgrade bundle can be found.

5. RESULTS

To test the loss-compensated optimization algorithm, we built an experimental testbed using network simulator 2 (ns2) [6]. We first assume the following QoS set $\mathcal{Q} = \{0, 1, 2\}$. $q = 1$ implies an unprotected packet transmission through selected path t with packet loss rate α_t, and $q = 2$ implies Reed-Solomon $RS(3, 2)$ is used for packet protection. Correspondingly, the costs are $c(1) = 1$ and $c(2) = 1.5$ respectively. We performed two trials, with raw path loss rates at $(\alpha_0, \alpha_1) = (0.06, 0.10)$ and $(\alpha_0, \alpha_1) = (0.04, 0.08)$, respectively. The total bandwidth of both paths are kept constant while the bandwidth of path 0 is varied.

For video source, we use H.264 JM4.2 [2] to encode two 300-frame QCIF sequences, `news` and `stef`, at 15 fps. The quantization parameters for I-frame and P-frames are, respectively, 25 and 20 for `news`, and 37 and 32 for `stef`. This results in source rate of $180kbps$ (`news`) and $190kbps$ (`stef`) if each P-frame F_i is coded using F_{i-1}.

We fixed the combined bandwidth of the two paths, $\bar{R}_0 + \bar{R}_1$, at $200kbps$ and $210kbps$ for the two sequences respectively. By varying the share of total bandwidth to the first path bandwidth $\bar{R}_0$, we tracked the corresponding PSNR at the client. The sequences were replayed 400 times. We use values of $E_{\max} = 5$ and $M = 10$.

We compared our new optimization scheme with loss compensation `lc` to our previous scheme without loss compensation `nlc`. We see that our proposed `lc` outperformed `nlc` for both sequences for almost all ranges. In particular, for `news`, `lc` outperformed `nlc` up to $2.03dB$ for trial 1 and $1.55dB$ for trail 2. For `stef`, `lc` outperformed `nlc` up to $1.75dB$ for trial 1 and $1.45dB$ for trail 2.

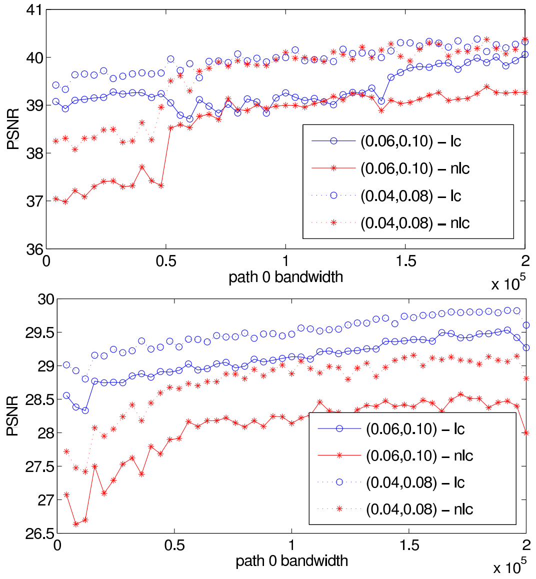

Fig. 4. PSNR comparison for `news` at 200 kbps (above) and `stef` at 210 kbps (below).

6. CONCLUSION

In this paper, we improve upon a previously proposed reference frame / QoS / path selection algorithm for multipath streaming by intelligently tracking loss compensation during optimization. An algorithm that efficiently tracks the partial reconstruction value of correctly decoded frames during loss compensation is presented. Results show noted improvement over a previous optimization scheme.

7. REFERENCES

[1] G. Cheung and W. t. Tan, "Reference frame optimization for multipath video streaming using complexity scaling," in *International Packet Video Workshop*, Irvine, CA, December 2004.

[2] "The TML project web-page and archive," http://kbc.cs.tu-berlin.de/ stewe/vceg/.

[3] T. Wiegand, N. Farber, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," in *IEEE J. Select. Areas. Comm.*, June 2000, vol. 18, no.6, pp. 1050–1062.

[4] Y.J. Liang, M. Flieri, and B. Girod, "Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection," in *IEEE International Conference on Image Processing*, Rochester, NY, September 2002.

[5] Y.J. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in *IEEE Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, December 2002.

[6] "The network simulator ns-2," August 2003, release 2.26, http://www.isi.edu/nsnam/ns/.