REFERENCE FRAME SELECTION FOR LOSS-RESILIENT TEXTURE & DEPTH MAP CODING IN MULTIVIEW VIDEO CONFERENCING

Bruno Macchiavello [*], *Camilo Dorea* [*], *Edson M. Hung* [*], *Gene Cheung* [#], *Wai-tian Tan* [o]

[*] Universidade de Brasilia, Brazil, [#] National Institute of Informatics, Japan,
[o] Hewlett-Packard Laboratories, USA.

ABSTRACT

In a free-viewpoint video conferencing system, the viewer can choose any desired viewpoint of the 3D scene for observation. Rendering of images for arbitrarily chosen viewpoint can be achieved through depth-image-based rendering (DIBR), which typically employs "texture-plus-depth" video format for 3D data exchange. Robust and timely transmission of multiple texture and depth maps over bandwidth-constrained and loss-prone networks is a challenging problem. In this paper, we optimize transmission of multiview video in texture-plus-depth format over a lossy channel for free viewpoint synthesis at decoder. In particular, we construct a recursive model to estimate the distortion in synthesized view due to errors in both texture and depth maps, and formulate a rate-distortion optimization problem to select reference pictures for macroblock encoding in H.264 in a computation-efficient way, in order to provide unequal protection to different macroblocks. Results show that the proposed scheme can outperform random insertion of intra refresh blocks by up to 0.73 dB at 5% loss.

Index Terms— Depth-image-based rendering, video streaming

1. INTRODUCTION

Depth-image-based rendering (DIBR) is an image synthesis technique that enables rendering of an image from an arbitrarily chosen virtual viewpoint in a multiview video system. It requires the per-pixel distances between the closely spaced cameras and locations of the captured objects (depth maps), in addition to RGB images (texture maps) and camera calibration parameters. Depth maps can be obtained by estimation algorithms like stereo-matching, or sensors like time-of-flight cameras. DIBR can be used in several scenarios. In some advanced 3D video systems, only texture and depth maps of a single view are transmitted, and the additional neighboring view required for stereo vision is rendered [1]. DIBR can also be used to implement *free-viewpoint television*, in which a viewer can choose any desired view for personalized video playback [2].

In an interactive multiview video conferencing system, free viewpoint is desirable and can improve human's visual perception of depth in the 3D scene through *motion parallax*. Most previous works focus on the compression performance of multiview video in "texture-plus-depth" format [3]. Some recent works focus specifically on depth maps compression [4, 5]. For example, [4] showed that during depth map encoding, H.264/AVC coding modes can be selected based on individual macroblock's (MB) impact on synthesized view distortion. In contrast, in this work we focus on the problem of reliable and timely transmission of "texture-plus-depth" format of multiview video over bandwidth-constrained and loss-prone networks, which is challenging for conferencing applications with stringent video playback deadlines.

Error-resilient streaming of video (texture) has been widely studied, and different techniques have been proposed including encoding, transport, as well as post-processing error concealment methods [6]. In this paper, we utilize reference picture selection (RPS) in H.264 to provide unequal protection to different MBs in texture and depth videos. The main goal is to minimize the expected synthesized view distortion due to packet losses during transmission. RPS is done at the MB level since not all MBs are equally important during view synthesis. In particular, we first construct a recursive distortion model to capture the effects of MB losses in texture and depth maps on synthesized view distortion. Then, we formulate a rate-distortion optimization problem, where MBs in texture and depth maps can select different coding modes and reference pictures, inducing different tradeoffs between rate and synthesized distortion. We propose a Lagrangian-based algorithm to solve the MB RPS problem in a computation-efficient manner, which is suitable for our intended real-time video conferencing application. Experimental results show that our proposed scheme can outperform random insertion of intra refresh blocks by up to 0.73dB at 5% loss.

2. RELATED WORK

Using the flexibility of RPS in H.264 to control error propagation in motion compensated frames have been studied for single-view video at the frame level [7]. In contrast, we optimize RPS at the MB level according to the importance of each MB in texture and depth maps, in order to minimize the expected synthesized view distortion.

There are well-known recursive distortion models in the literature. In [8], an *recursive optimal per-pixel estimate* (ROPE) is proposed to estimate per-*pixel* distortion by calculating the first and second order moments of its decoded value. A frame-level distortion model for multiview video transmission is proposed in [9]. Our work is inspired by [9], but extends the estimation from per-frame to per-block to support selective use of reference pictures. Compared to [8], our work supports loss in both texture and depth, and avoids the unnecessary computation in deriving per-pixel distortion when decisions are made at block level.

In [10], a similar scheme to minimize expected synthesized view distortion based on selection of reference frame at the block level was proposed for depth maps only. In this work, we first extend the idea in [10] to encoding of both texture *and* depth maps, where the relative importance of texture and depth MBs must be determined. Second, we expand the coding modes available to each MB to include intra block coding. Third, we derive an objective function and corresponding optimization algorithm for encoding of both texture and depth maps that nonetheless remains computationally efficient.

This work was partially supported DPP/Universidade de Braslia and by CNPq grant 310375/2011-8.

3. PROBLEM FORMULATION

For both texture and depth map, we intelligently select a predictor MB for motion compensation (MC) for each MB in a current frame at time t. Our goal is to minimize the expected distortion of an intermediate view at instant t, synthesized via DIBR using texture and depth maps of two adjacent coded views at decoder, and subject to a transmission rate constraint. We first discuss how synthesized distortion in an interpolated view is affected by reconstruction errors in anchor texture and depth maps. We then present the mathematical formulation of the optimization.

3.1. Overview of Distortion in DIBR

Signal distortion at a synthesized viewpoint can arise from two causes: i) *textural error* representing copying of erroneous texture pixels in the anchor texture map of an adjacent captured view, and ii) *disparity error* caused by error in the anchor depth map of an adjacent captured view, leading to geometric error of the captured scene, and subsequent pixel copy to the wrong spatial location in the synthesized image. We examine how these two kinds of errors contribute to the synthesized distortion more closely next.

A pixel (i, j) in texture map X_t at a captured viewpoint will map to a pixel in synthesized image S_t shifted horizontally by disparity $Y_t(i, j) * \eta$:

$$S_t(i, j - Y_t(i, j) * \eta) = X_t(i, j) \tag{1}$$

where η is the shift parameter that depends on the particular camera setup, and the location of the intermediate viewpoint between the two captured views. Hence, if texture value $X_t(i, j)$ is incorrectly reconstructed at decoder by an amount e, then reconstructed pixel $S_t(i, j - Y_t(i, j) * \eta)$ also inherits[1] distortion e.

On the other hand, if depth value $Y_t(i, j)$ is incorrectly reconstructed by ϵ, then the wrong geometric information will lead to synthesis of a wrong pixel $(i, j - (Y_t(i, j) + \epsilon) * \eta)$ in interpolated view S_t. Assuming the texture in synthesized view S_t is similar to the texture map X_t, then the disparity error ϵ will lead to synthesized distortion $X_t(i, j) - X_t(i, j + \epsilon * \eta)$. For example, if $X_t(i, j)$ is a texture pixel inside a relatively smooth spatial region, then disparity error ϵ will lead to little (if any) synthesized distortion. If $X_t(i, j)$ is a texture pixel close to an object boundary, however, then disparity error ϵ may lead to copying of texture pixel from foreground object to background (or vice versa), resulting in large distortion. We can thus make two observations regarding disparity error ϵ: i) unlike textural error, the resulting distortion in synthesized view S_t is not linear to ϵ, and ii) resulting synthesized distortion in S_t depends on textural patterns in local neighborhood around $X_t(i, j)$.

3.2. A Recursive Error Model

We now derive the expected textural error e in differentially coded texture maps due to channel losses. (Derivation for expected disparity error ϵ in differentially coded depth maps is the same and thus omitted.) Let $e_{t,i}(\tau_{t,i}, v_{t,i})$ be the error of MB i of frame X_t, given it is motion-compensated using a block identified by motion vector (MV) $v_{t,i}$ inside frame $X_{\tau_{t,i}}$, $\tau_{t,i} \leq t$. If MB i is coded as an intra block, then $\tau_{t,i} = t$ and $v_{t,i} = 0$. Let p be the probability that MB i of X_t is correctly *received*. We can now write $e_{t,i}(\tau_{t,i}, v_{t,i})$ in terms of $e^+_{t,i}(\tau_{t,i}, v_{t,i})$ and $e^-_{t,i}$, the error of MB i of X_t if coded MB i is correctly received and lost, respectively:

$$e_{t,i}(\tau_{t,i}, v_{t,i}) = p\, e^+_{t,i}(\tau_{t,i}, v_{t,i}) + (1 - p)\, e^-_{t,i} \tag{2}$$

[1]If pixel blending is used during DIBR where one pixel from each captured view is mixed for each synthesized pixel in S_t, then distortion in synthesized pixel is linear to e.

If MB i of X_t is correctly received, then the corresponding error $e^+_{t,i}(\tau_{t,i}, v_{t,i})$ depends first on whether MB i is coded as intra or inter block, and if it is the latter, on the quality of the reference block b identified by MV $v_{t,i}$ in frame $X_{\tau_{t,i}}$. In general, reference block b can be interpolated using several neighboring MBs $k \in v_{t,i}$ at integer pixel coordinates, due to sub-pixel accuracy in H.264's MC. We can hence write $e^+_{t,i}(\tau_{t,i}, v_{t,i})$ as a weighted sum of errors of these MBs if MB i is an inter block:

$$e^+_{t,i}(\tau_{t,i}, v_{t,i}) = \begin{cases} 0 & \text{if } \tau_{t,i} = t \\ \gamma \sum_{k \in v_{t,i}} \alpha_k\, e_{\tau_{t,i},k}(\tau_{\tau_{t,i},k}, v_{\tau_{t,i},k}) & \text{o.w.} \end{cases} \tag{3}$$

where α_k's are the weights for the summation, and $\gamma < 1$ is the attenuation factor that reflects the dissipating effect of error in an earlier frame over a sequence of motion-compensated frames.

If MB i of X_t is lost, then we assume a simple block copy procedure is used for loss concealment, where MB i of previous frame X_{t-1} is used in its place. In this case, the error $e^-_{t,i}$ will be the previous MB's error $e_{t-1,i}(\tau_{t-1,i}, v_{t-1,i})$ *plus* block difference between MB i of frame X_{t-1} and MB i of frame X_t, $\delta_{t-1,i}$:

$$e^-_{t,i} = e_{t-l,i}(\tau_{t-1,i}, v_{t-1,i}) + \delta_{t-1,i} \tag{4}$$

The recursive definitions above compute channel-induced errors given inter-frame dependencies established during MC of previous frames. To provide a base case for the recursion, we assume there exists either an intra block or an *acknowledged MB* (ACKed MB) in every dependency chain, one where the receiver has indicated it has been decoded without error, so that its channel-induced error is 0.

3.3. Optimization Formulation

Having derived expected textural error $e_{t,i}$ and disparity error $\epsilon_{t,i}$ in differentially coded texture and depth maps X_t and Y_t, we perform optimization by minimizing synthesized distortion subject to a transmission rate constraint, as follows. Textural error $e_{t,i}$, as discussed in Section 3.1, contributes directly to the synthesized distortion. For disparity error $\epsilon_{t,i}$, we first determine the curvature $a_{t,i}$ of a quadratic penalty function $g_{t,i}()$ [5] that models the local synthesized view distortion sensitivity to disparity value for MB (t, i). The resulting synthesized distortion due to disparity error is then the quadratic function $g_{t,i}(\epsilon_{t,i})$ evaluated with argument $\epsilon_{t,i}$. For example, if MB (t, i) is inside a flat spatial region with little texture, then $g_{t,i}()$ is very flat with small resulting distortion, for reasonably small disparity error $\epsilon_{t,i}$. In summary, our objective function is the following:

$$\min_{\{\tau_{t,i}, v_{t,i}, \rho_{t,i}, u_{t,i}\}} \sum_i e_{t,i}(\tau_{t,i}, v_{t,i}) + g_{t,i}\left(\epsilon_{t,i}(\rho_{t,i}, u_{t,i})\right) \tag{5}$$

where the penalty function $g_{t,i}()$ is:

$$g_{t,i}(\epsilon_{t,i}(\rho_{t,i}, u_{t,i})) = \frac{1}{2} a_{t,i} \left[\epsilon_{t,i}(\rho_{t,i}, u_{t,i})\right]^2 \tag{6}$$

Note that (5) is an approximation of the actual synthesized distortion at intermediate view S_t, since in general, textural error $e_{t,i}$ and disparity error $\epsilon_{t,i}$ affect synthesized distortion in a complicated, non-linear way, especially when both $e_{t,i}$ and $\epsilon_{t,i}$ are large. Nonetheless, (5) is a good approximation when only one of the two errors is non-zero, and it leads to a simple optimization procedure as discussed in Section 4.

The optimization is subject to the rate constraint R_t at instant t:

$$\sum_i r_{t,i}(\tau_{t,i}, v_{t,i}) + \zeta_{t,i}(\rho_{t,i}, u_{t,i}) \leq R_t \tag{7}$$

where $r_{t,i}$ and $\zeta_{t,i}$ are the resulting bit overhead required to code texture and depth MB (t,i), given selection of reference frame / MV pair, $(\tau_{t,i}, v_{t,i})$ and $(\rho_{t,i}, u_{t,i})$, respectively.

4. ALGORITHM

Instead of solving the constrained optimization problem (5) and (7), we can solve the corresponding Lagrangian problem instead for given multiplier $\lambda > 0$:

$$\min_{\{\tau_{t,i}, v_{t,i}, \rho_{t,i}, u_{t,i}\}} \quad \sum_i e_{t,i}(\tau_{t,i}, v_{t,i}) + g_{t,i}\left(\epsilon_{t,i}(\rho_{t,i}, u_{t,i})\right) + \\ + \lambda \sum_i r_{t,i}(\tau_{t,i}, v_{t,i}) + \zeta_{t,i}(\rho_{t,i}, u_{t,i}) \quad (8)$$

To solve (8) optimally, it is clear that we can separately optimize each texture or depth MB (t,i), each containing its corresponding textural or disparity error and rate term:

$$\min_{\tau_{t,i}, v_{t,i}} \quad e_{t,i}(\tau_{t,i}, v_{t,i}) + \lambda\, r_{t,i}(\tau_{t,i}, v_{t,i}) \quad \forall i \quad (9)$$

$$\min_{\rho_{t,i}, u_{t,i}} \quad g_{t,i}\left(\epsilon_{t,i}(\rho_{t,i}, u_{t,i})\right) + \lambda\, \zeta_{t,i}(\rho_{t,i}, u_{t,i}) \quad \forall i \quad (10)$$

(9) and (10) are minimized by searching through all feasible MVs in all valid reference frames. This can be done efficiently, for example, in a parallel implementation.

5. EXPERIMENTATION

We compared our methods outlined in Sections 3-4 to a method using insertion of random intra-coded blocks for resilience. Both schemes exploit feedback to allow an encoder to avoid using as prediction reference earlier frames that are known to be loss-impaired. We call our scheme *"Modified H.264"*, and the other scheme *"Conventional H.264 + Feedback and Intra Refresh"*. The introduced intra-coded blocks are constrained to prohibit intra-prediction using inter-coded blocks to enhance error resilience. Currently, our schemes are implemented only for $P16 \times 16$ mode in H.264/AVC JM reference software v18.0 [11]. Therefore, the only modes available in all simulations are $P16 \times 16$ or Intra blocks. More extensive comparison using larger number of available modes is a subject of future study. The number of Intra Refresh blocks (for *"Conventional H.264 + Feedback and Intra Refresh"*) inserted is varied to match the bit-rate in both schemes.

Each depth map frame is divided into three packets, while each texture frame is divided into twelve packets due to higher associated bit rates. Simulations include losses of 2%, 3% and 5% of the packets for both texture and depth maps. In order to provide meaningful comparisons, the same packets are lost in all schemes. Both depth maps and texture are encoded using 64 pixel search window, CABAC entropy encoder and $IPPP...$ encoding mode. When a MB is lost during transmission, the co-located block from the previous frame is used in its place. For view interpolation we used the MPEG View Synthesis Reference Software (VSRS v3.5) [12]. The simulation is for round-trip-delay of 4 frames or 133 ms.

The results are shown in Fig 1. We also included the error-free results for reference. For "Kendo" sequence [13] view 1 was synthesized from views 0 and 2, and for "Champagne" sequence [13] view 40 was synthesized from views 39 and 41. The original views were used as ground truth for PSNR calculations. The total bit-rate for each curve (for both textures and depth maps) are indicated in the caption. Due to the use of feedback, we see that both schemes are generally able to recover from losses within one round-trip time. Nevertheless, we see that our optimized scheme can better withstand the transient effect of packet losses by providing stronger protection to more important regions, especially when loss rates are higher. Specifically, for Kendo at 5% loss, we see that our scheme provides over 1 dB of PSNR improvement in 4 out of the 7 episodes of loss followed by recovery. Corresponding number for Champagne is 4 out of 10 episodes. The average PSNR improvements for Kendo are 0.204 dB, 0.722 dB and 0.734 dB for 2%, 3% and 5% losses, respectively. Similarly, the PSNR improvement for Champagne at 2%, 3% and 5% losses are 0.161 dB, 0.214 dB and 0.355 dB, respectively. The results are generated for only a portion of the sequences with interesting motion, since losses in static frames can be readily concealed.

Detail crops are shown in Fig.2 and Fig. 3. The errors around the swordsman of the right in Fig. 2-(a) and the errors in the arm in Fig. 3-(a) attest to the effectiveness of our scheme.

6. CONCLUSION

In this paper, we have presented a recursive distortion model to estimate the effects of packet losses on view interpolation using "texture-plus-depth" video in DIBR. We extended earlier model that consider losses in depth map only to cover both depth and texture maps, and developed an algorithm for the solution. Our experiments using H.264/AVC, though without support of block partitions and sub-partitions, show a significant improvement in PSNR and subjective quality compare to random insertion of intra blocks with a feedback channel available.

7. REFERENCES

[1] C. Fehn, K. Hopf, and Q. Quante, "Key technologies for an advanced 3D-TV system," in *Proc. of SPIE Three-Dim. TV*, Philadelphia, Oct. 2004.

[2] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint TV," in *IEEE Signal Processing Magazine*, January 2011, vol. 28, no.1.

[3] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE ICIP*, San Antonio, TX, October 2007.

[4] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *IEEE ICIP*, Cairo, Egypt, November 2009.

[5] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform domain sparsification of depth maps using iterative quadratic programming," in *IEEE ICIP*, Brussels, Belgium, September 2011.

[6] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *IEEE Proceedings*, vol. 86, pp. 974–997, 1998.

[7] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multiple-path video streaming with complexity scaling," in *IEEE Trans. on CSVT*, June 2007, vol. 17, no.6, pp. 649–662.

[8] R. Zhang, S.L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," in *IEEE J. Select. Areas. Comm.*, June 2000, vol. 18, no.6, pp. 966–976.

[9] Y. Zhou, C. Hou, W. Xiang, and F. Wu, "Channel distortion modeling for multi-view video transmission over packet-switched networks," in *IEEE Trans. on CSVT*, November 2011, vol. 21, no.11, pp. 1679–1692.

[10] B. Macchiavello, C. Dorea, E. M. Hung, G. Cheung, and W. t. Tan, "Reference frame selection for loss-resilient depth map coding in multi-view video conferencing," in *IS&T/SPIE Visual Info. Proc. and Comm. Conf.*, Burlingame, CA, January 2012.

[11] "JM H.264 reference software v18.0," in *http://iphome.hhi.de/suehring/tml/*.

[12] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," in *ISO/IEC JTC1/SC29/WG11 MPEG2008/M15377*, Archamps, April 2008.

[13] "Nagoya university ftv test sequences," in *http://www.tanimoto.nuee.nagoya-u.ac.jp/*.

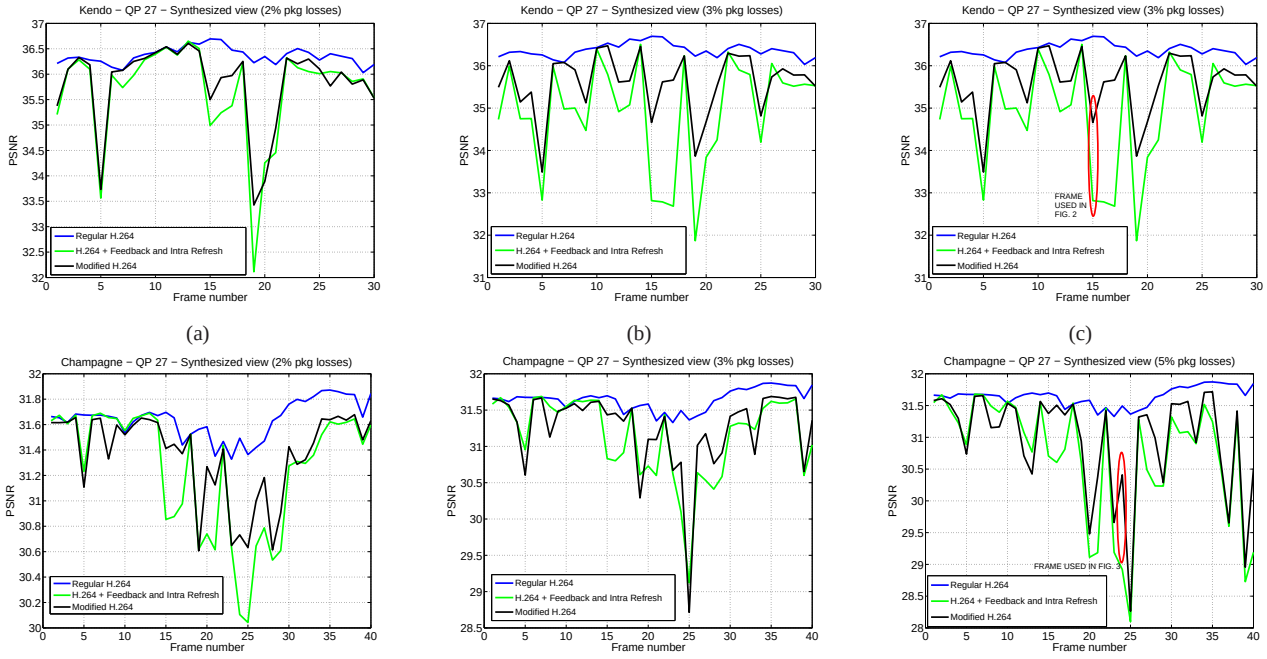

Fig. 1. PSNR vs Frame plots for Kendo at 1024×768 pixels (top row) and Champagne at 1280×960 (bottom row). The bitrates for Kendo are 6.6 Mbps for *Conventional H.264*, and 9.0 Mbps for both *"Conventional H.264 + Feedback and Intra Refresh"* and *Modified H.264*. For Champagne, the bitrates are 5.0 Mbps and 8.0 Mbps, respectively.

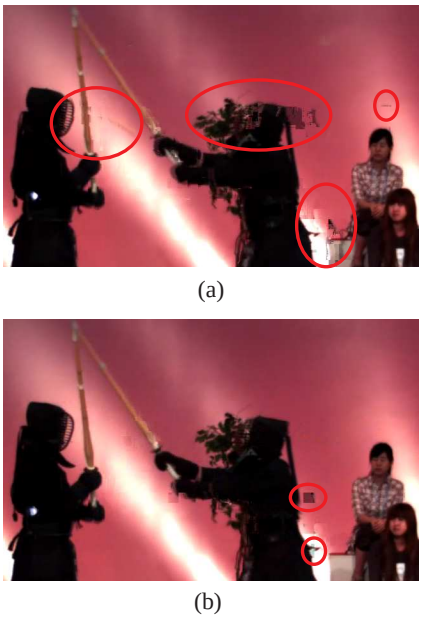

Fig. 2. Cropped frame of Kendo with 5% packet losses for (a) *"Conventional H.264 + Feedback and Intra Refresh"*, and (b) *Modified H.264*.

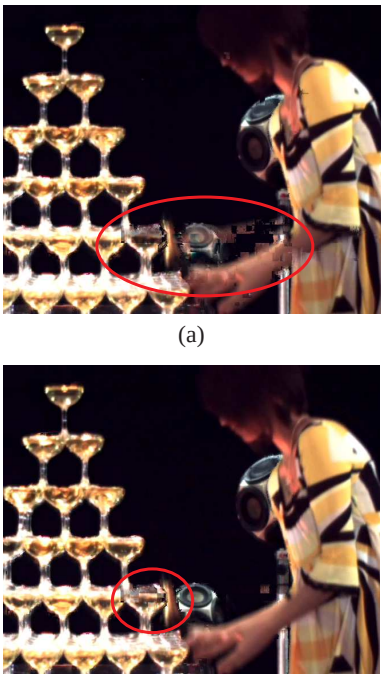

Fig. 3. Cropped frame of Champagne with 5% packet losses for, (a) *"Conventional H.264 + Feedback and Intra Refresh"*, and (b) *Modified H.264*.