# Cross Segment Decoding of HEVC for Network Video Applications

Jiangtao Wen, Shunyao Li, Yao Lu, Pin Tao

***Abstract*—In this paper, we present an improved algorithm for decoding video bitstreams with time-varying visual quality. The algorithm extracts information available to the decoder from a high visual quality segment of the clip that has already been received and decoded, but was encoded independently from the current lower quality segment. The proposed decoder is capable of significantly improving the Quality of Experience of the user without incurring significant delays and overhead to the storage and computational complexities of both the encoder and the decoder, or loss of coding efficiency. We present simulation results using the HEVC reference encoder and standard test clips, and discuss areas of improvements to the algorithm.**

## I. Introduction

Video encoding and communications systems have traditionally been designed under the assumption that the encoder has a much higher computational power and much larger storage than the decoder. With the ever widening popularity of mobile multimedia applications, especially with user generated content, this assumption is no longer valid. Many widely watched clips on YouTube were captured on mobile phones, but are played back not only on mobile phones or tablets but also on smart TVs, smart set-top-boxes, as well as laptop and desktop computers, all of which may possess much more computational and storage resources than the mobile phone on which the video clip was originally captured and encoded.

Due to the high complexity associated with implementing a state-of-the-art video encoding system using the H.264/AVC [1] and especially the up-coming HEVC standard [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], encoders found in mobile devices, although compliant to the standard syntaxes, were often designed with sub-optimal implementations of only small subsets of the numerous encoding tools supported by the standard. Typical video encoding tools that are not fully implemented include sub-pel motion estimation (ME), many Inter partition sizes and/or Intra prediction directions, and etc. The number of reference frames used in the ME is often limited. In addition, the algorithms for selecting the tools that *are* supported, for conducting rate-distortion optimized ME, quantization, mode decision, frame prediction type decision and group of picture (GOP) reference structures are also drastically simplified in many encoding systems. Various implementation constrains (e.g. cache sizes) as well as application requirements (e.g. video conferencing or live streaming) make it very difficult to optimize bit rate allocation globally for the entire clip.

Furthermore, when the bitstreams generated by such encoders are streamed over the network, in response to network bandwidth variations, the streaming servers will often adjust the rate at which the clip is encoded downward so as to prevent the playback on the receiving device from stalling.

All of the above factors contribute to losses in coding efficiency as well as visual quality variations in the bitstream that the playback device receives. It is desirable, therefore, to come up with a system that can improve the quality of experience (QoE) of the end user given all the limitations, constraints and quality variations present in the content encoding, transcoding, streaming and playback processes.

In this paper, we present an improved algorithm for decoding video bitstreams with time-varying qualities. The algorithm utilizes information received by the decoder in a segment of the clip that 1) has already been received and decoded, but 2) was encoded independently from the current segment, and 3) has a higher visual quality than the current segment. By extracting information contained in such a segment that is available to the decoder but was not taken advantaged by the encoder, the proposed decoder is capable of significantly improve the QoE of the user without incurring significant overhead to the storage and computational complexities of both the encoder and the decoder, or introducing significant delays or losses to coding efficiency.

The rest of the paper is organized as the following:

Jiangtao Wen, Shunyao Li, and Pin Tao are with Tsinghua University, Beijing, China (email: jtwen@tsinghua.edu.cn)

Yao Lu was with Tsinghua University, Beijing, China, and is now with the University of California San Diego, La Jolla, CA 92093 (email: luyao@ucsd.edu)

An early version of the paper with a very rudimentary algorithm for enhanced decoding of low motion clips was submitted and accepted by the IEEE 2013 Data Compression Conference.

Section II reviews some related ideas for improving user QoE and decoder quality. A detailed description of the proposed algorithm is given in Section III with experimental results using the HEVC standard and standard test clips presented in Section IV. Finally in Section V, we discuss various areas for improving the algorithm.

## II. Related Work

Scalable, error resilient and high quality streaming of video content over networks has been studied extensively for well over a decade. In additional to taking the tradeoff between scalability, error resilience and coding efficiency (as measured in rate-distortion performances) into the consideration when designing the video encoding algorithms (e.g. for ME, rate control, and mode decision), many additional tools were introduced to various video coding standards so as to facilitate video streaming with low start-up latency, as well as easier and drift-free bitstream switching. These include scalable video coding support in the MPEG family of video coding standards, S-frames [12], as well as SI and SP frames in the AVC/H.264 standard [1], [13], [14].

When encoding video content in a rate-disortion-scalability-error-resiliency optimal manner, to prevent error propagation and to facilitate drift-free bitstream switching, encoders usually do not rely on video encoding tools that aggressively eliminate temporal redundancies, e.g. long term motion prediction with a large number of reference frames. Quite the contrary, the encoders will usually introduce Intra coded frames that will serve as re-synchronization points or drift-free switching pointers to switch between bitstreams of different bitrates. Due to the low coding efficiency of Intra frames, the more efficient SP and SI frames were proposed, which preserved the drift-free characteristic, but with coding efficiencies significantly improved over Intra frames, albeit still significantly lower than P or B frames. Using Intra, SI and SP frames as switching points, an encoder may encode a given video clip to multiple possible bitstreams with multiple configurations of the reference structure, and/or bitrates. When streaming, depending on network packet losses and network bandwidth variations, the server/streamer may compose a bitstream to be streamed to the client on the fly, based on information about frame losses and bandwidth.

In contrast to encoding a given video clip at multiple bitrates and references structures, scalable video coding (SVC) [15] encodes the video clip into one scalable bitstream that can be parsed by the server to produce bitstreams of many target bitrates. Although more efficient than storing multiple bitstreams for the same content, SVC still significantly under-performs conventional, non-scalable coding systems with similar computational and storage resources and video encoding tools. Special, SVC-compliant decoders are also usually required at the receiving end.

On the other hand, many algorithms and systems have been introduced to improve the QoE on the receiving end when packet losses and bandwidth variations occur. The algorithms include various error concealment and error resilient decoding techniques, many of which were reviewed in [16] and [17]. In [18], a technique (termed “Adaptive Media Playout”) was introduced to dynamically adjust the playback speed of video and audio based on the playback buffer of the receiver, thereby preventing stalls. In [19], when decoding the prediction residual information in a received frame using an algorithm called “delayed decoding”, an optimized estimate of the transform coefficients to be decoded is obtained by the decoder using information contained in frames both before and after the current frame in *decoding* order. Deblocking filters have also been used to improve the subjective and objective qualities of the video at the decoder. Since H.264/AVC, a standardized deblocking filter has been incorporated into the encoding loop of the encoder [20], [21], [22].

In this paper, we introduce an improved decoding algorithm that also improves the decoded quality, but without extensive modifications to the encoding process, as were required in the case of scalable coding and encoding with S, SI and SP frames. Similar to delayed decoding, the proposed algorithm also achieves improvement to the decoded quality by using information the decoder already processes but is not traditionally utilized by conventional decoders. However, unlike delayed-decoding, the extra delay and storage and computational complexity introduced are much lower.

## III. Algorithm Description

The proposed algorithm is more easily understood in the context of bitrate adaptive streaming of video over the Internet, where to facilitate fine granularity bitrate adaptation in reaction to changes in network conditions, a video clip is divided into relatively short segments, each of which is encoded independently of each other, as illustrated in Figure 1. In the figure, a video clip is divided into 3 segments, each encoded at 3 different bitrates, $bitrate1 < bitrate2 < bitrate3$.

When a clip encoded using the system in Figure 1 is streamed over a network where the bandwidth varies, the server may "stitch" together bitstreams for neighboring segments that have been encoded at different bitrates, as shown in Figure 2, resulting in variations of video quality over time. In many applications, such variations in visual quality is noticeable, annoying and significantly impair the user QoE.

Similar variations in visual quality may also occur to an encoder with a rate allocation algorithm that is not able to allocate the target bitrate in a globally optimized manner over the entire clip. This may be due to the lack of multiple pass encoding (e.g. for encoding live events) or sufficient look ahead (due to memory or delay requirements), and/when the complexity of the input video varies significantly over time. Figure 3 shows an example, where a uniform 1Mbps was allocated to encode the "Chroma Key" test clip. Figure 3(a) and Figure 3(b) show two frames of roughly the same size after 1-pass encoding using the x264 [23] encoder with the default settings. The visual qualities of the two frames are noticeably different.

For the remainder of the paper, when the visual quality of an input bitstream to a video decoder incorporating the proposed algorithm varies over time, at the transition from a segment with higher video quality to a temporally neighboring segment that is encoded completely independent of the good quality segment and whose video quality is poorer (Figure 2), we term the last frame (in display order) in the higher quality segment a "good frame" (GF), the first IDR frame of the poor quality segment the "start frame" (SF), and the output from the current algorithm the "fresh start" (FS). Note that the SF as an IDR frame was encoded without referencing the GF or any other frames in the higher quality segment.

As described previously, the goal of the enhancement algorithm is to use information contained in the GF to improve the quality of the decoded SF to get an improved reference frame FS for subsequent frames in the low quality segment. Depending on the level of motion for different spatial regions of the SF, two enhancement algorithms might be used by the decoder, one for relatively low motion areas, the other for the higher motion areas. For both algorithms, the decoder will look for matches between areas in the decoded GF and the SF, as determined by a distortion metric and a threshold calculated by the decoder.

### A. Automatic Segmentation of the SF

To segment the SF into high motion and low motion areas, ME was conducted between the SF and the GF at the *decoder*. After the ME, the SF was divided into non-overlapping 32x32 patches with the motion vectors (MVs) for each patch averaged and compared to a threshold $Th_{MV}$. Note that each patch may overlap with multiple Prediction Units (PUs). In our experiments, $Th_{MV}$ was set to

$$\frac{width \times QP}{30000}, \tag{1}$$

where $w$ was the width of the video, and $QP$ was the (average) quantization parameter of the frame. The patches whose average MVs were below the threshold were designated as the low motion areas, denoted as $SF_{low}$, while the rest were designated as the high motion areas, denoted by $SF_{hi}$.

### B. Low Motion Area Enhancement

We then partitioned the low motion areas $SF_{low}$ into non-overlapping 16x16 patches. For each 16x16 patch, we calculated the Sum of Squared Differences (SSD) between its pixels and the corresponding pixels in the GF. If the SSD was smaller than a threshold $Th_{SSD}$, the patch in $SF_{low}$ was replaced with the patch in the GF.

Obviously, the performance of the proposed algorithm depends on the value of $Th_{SSD}$. In our study, we first exhaustively experimented all integer values of $Th_{SSD}$ between 10 and 600, and found the threshold $Th_{Opt}$ that provided the largest average PSNR gain over all frames after (and including) the SF in display order.

In Figure 4, we plotted the relationship between the values of $Th_{Opt}$ and a) the PSNR of the SF after Intra encoding, as well as b) the average (with regard to the number of MVs in the bitstream) rate-distortion (RD) cost for the MVs [24] [25] ($MECost$) between the decoded GF and SF as calculated by the decoder, i.e.

$$MECost = \frac{\sum_{\forall mv}\{SAD(mv) + \lambda_{ME} Bits(mv)\}}{\sum_{\forall mv} 1}, \tag{2}$$

were $SAD(mv)$ is the Sum of Absolute Differences for $mv$.

We then data-fitted the relationship between $Th_{Opt}$ and the $PSNR$ and $MECost$ (Figure 4) using a Laplacian and a power function respectively. The best fittings were found to be:

$$Th_1 = 1.112 \times e^{(-0.2963 \times PSNR + 15.14)} - 10.21 \tag{3}$$

for the Laplacian function, and

$$Th_2 = 6.213 \times MECost^{1.348} \tag{4}$$

for the power function. We define

$$Th_{SSD} = max(Th_1, Th_2), \tag{5}$$

and used $Th_{SSD}$ in all of our experiments. The intuition behind equations (3) to (5) is that

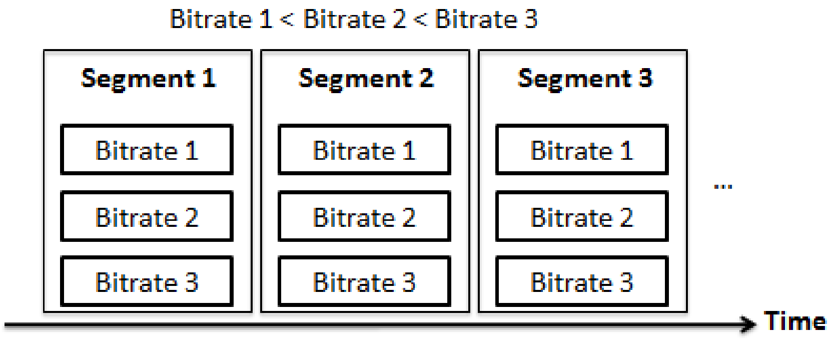


Fig. 1: Segment Based Bitstream Switching for Adaptive Video Streaming

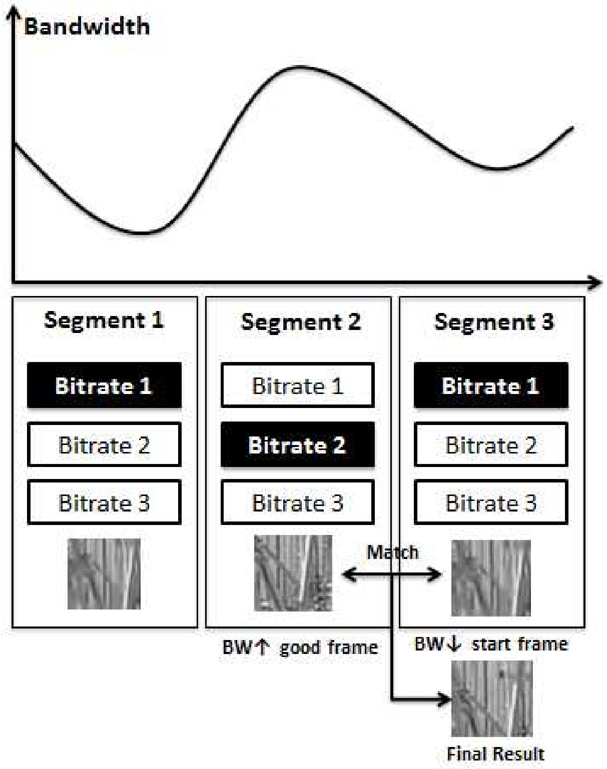


Fig. 2: Stitching Bitstreams for Segments for Streaming over Bandwidth Varying Channel

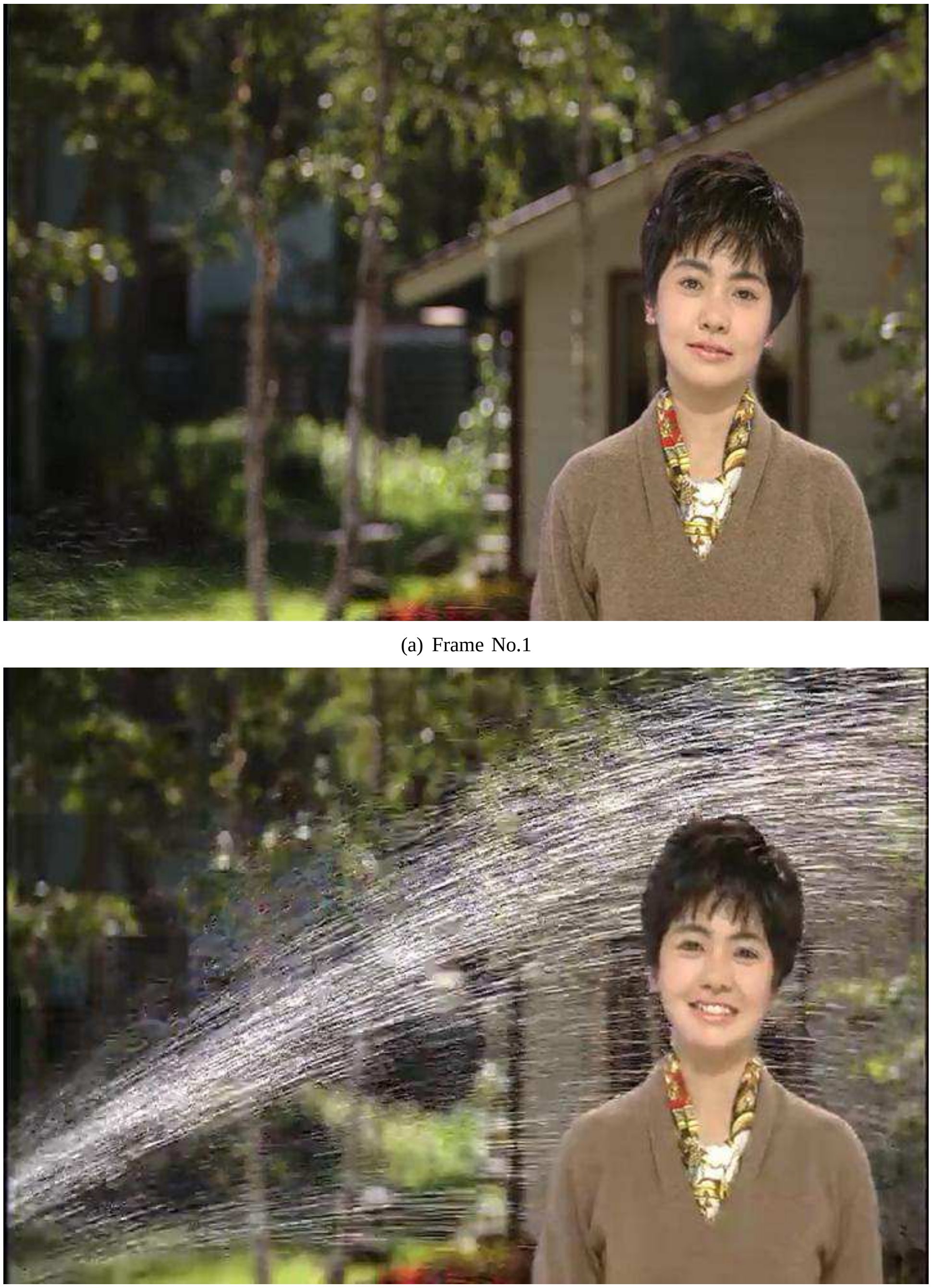
- The one of the two thresholds of (3) and (4) that leads to a larger number of patches designated as “matched” should be used to maximize the benefit of the presence of the GF,
- The value of the thresholds should be determined by the temporal similarity between GF and SF before encoding (hence the $MECost$ in (4)), as well as the loss of fidelity after encoding (therefore the PSNR in (3)).

The $PSNR$ value for the SF after IDR encoding can be embedded into the HEVC bitstream (e.g. as SEI information or user data) by the encoder using 16 bits. The PSNR could also be estimated by using techniques such as that in [26] without data embedding.

The pseudo code for the enhancement algorithm for low motion areas is given in Algorithm 1.

### C. High Motion Area Enhancement

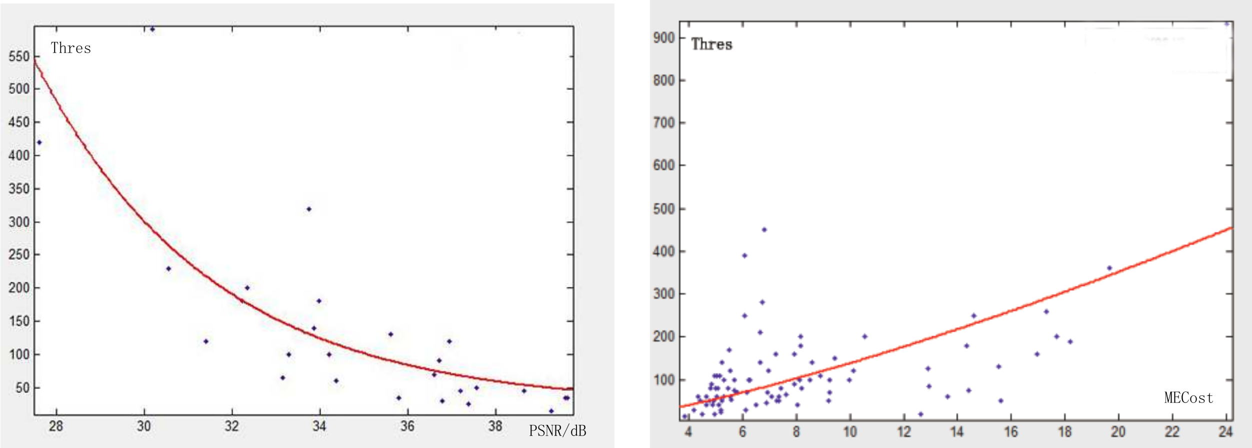
Motion information was required in the enhancement of the high motion areas $SF_{hi}$ with reference to the GF. In our experiments, we simply re-used the MVs obtained in the decoder ME process between the



(a) Frame No.1

(b) Frame No.250

Fig. 3: Two Frames of Almost Identical Size after Compression but Different Quality. Variations in Video Complexity and Uniformly Allocated Bitrate Result in Significant Variation of Quality over Time.



(a) $Th_{SSD}$-$PSNR$

(b) $Th_{SSD}$-$MECost$

Fig. 4: SSD Threshold Parameters Fitting

**Algorithm 1** LowMotionEnhancement ($SF_{low}$, GF)

```
for Each pixel 16x16 patch $P \in SF_{low}$ do
    Calculate $SSD(P, P')$ between P and co-located patch P' in GF.
    if $SSD(P, P') < Th_{SSD}$ then
        Copy $P'$ to $P$
    end if
end for
```

**Algorithm 2** HighMotionEnhancement ($SF_{hi}$, GF)

```
for Each 4x4 patch $P \in SF_{hi}$ do
    Find the 8 MVs from 8 immediate spatially neighboring 4x4 blocks of $P$
    if $MV(P)$ matches more than $Th_{mv}$ out of 8 neighbor MVs then
        for Each pixel $p \in P$ do
            find pixel $p'$ in the GF referenced by $MV(P)$
            if $|p - p'| < Th_Y$ then
                Copy $p'$ to $p$
            end if
        end for
    end if
end for
```

GF and the SF for the motion area segmentation and the calculations of the $MECost$ and $Th_{SSD}$. After the ME, we compared the motion vector $MV(P)$ for each 4x4 patch $P \in SF_{hi}$ and its eight immediate spatially neighboring 4x4 patches. If $MV(P)$ matched more than $Th_{mv}$ out of the 8 MVs from the eight 4x4 neighbors, then for each pixel $p \in P$, the difference between $p$ and the pixel $p'$ in the GF referenced by $MV(p)$ was calculated. The difference was then compared with a threshold $Th_Y$, with $p$ replaced by $p'$ if the difference is lower than $Th_Y$. In our experiments, we set $Th_{mv}$ to 6, and exhaustively tested possible values of $Th_Y$ between 5 and 53 using a step size of 2.

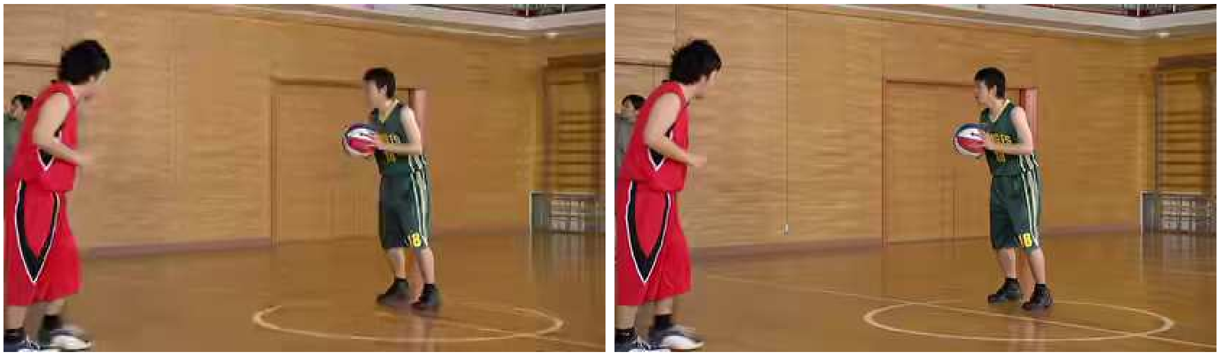
The pseudo code for the enhancement algorithm for high motion content is given in Algorithm 2.

## IV. Experimental Results

To evaluate the proposed algorithm, we used the HEVC HM 8.2 encoder and the low delay configuration to encode the test bitstreams. For each test clip, we ran the HEVC encoder for the first 32 frames of the clip to create the high quality segment, followed by HEVC encoding (with the same HEVC low delay configuration) of the remaining frames as the low quality segment with frame No. 33 encoded as an IDR frame and the SF. The QP used for encoding the first frame was set to be 5 levels lower than for the SF. The test clips included screen captures such as SlideEditing, video conferencing clips such as the Vidyo clips, as well as relatively higher motion clips such as the BaseketballPass and PartyScene.

The PSNR improvements for the SF, and averaged over 30 and 60 frames after (and including) the SF are given in Table I. In the table, the values listed under the QP column are the values used for encoding the first frame of the high quality segment.

As we can see, the PSNR improvements were significant for most of the test clips, with an average gain (with regard to all clips and bitrates) of 0.91 dB for the SF, and in most cases, a significant gain was achieved



(a) Standard Decoder

(b) Enhanced Decoder

Fig. 5: Subjective Quality Comparison for Motion Clip - BasetballPass

for at least 30 to 60 frames after the SF, even though the SF was the only frame to which the enhanced processing was performed. For some clips, the initial gain for the SF was lost after some frames, showing a net loss of average PSNR after 30-60 frames. This loss of the improvement to the SF over time occurred because after enhancing the SF, the decoder still used the same MV and residual information in the low quality bitstream for the decoding of the remaining frames in the low quality segment, even though the SF had already been modified to produce the actual reference frame of the FS. This led to mismatches between the residual information needed now that the FS was used as the reference, and the residual information in the bitstream, created by the encoder using the un-enhanced SF as the reference frame.

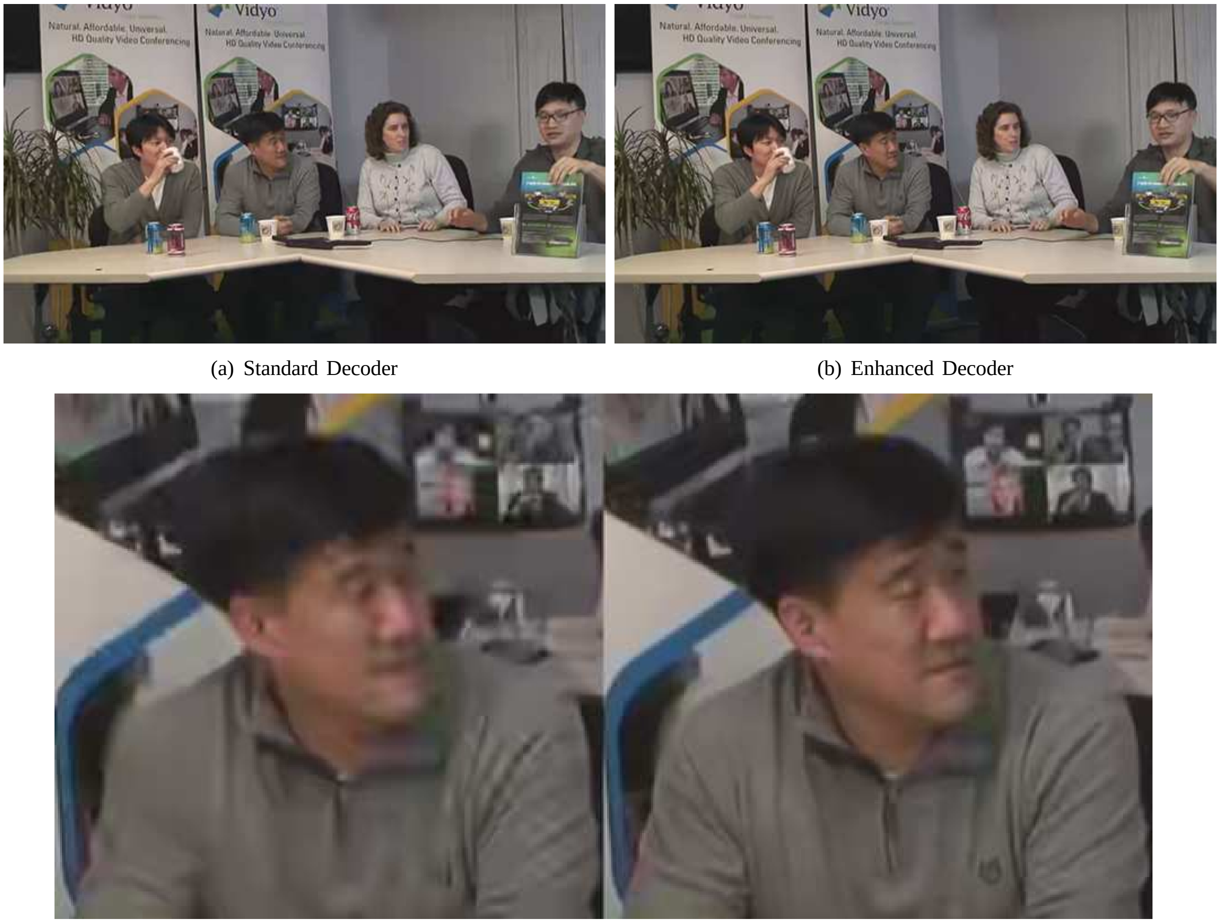
However, even with such mismatches, for many sequences, especially for video conferencing, screen capture and video surveillance applications and some clips with higher motion, a net gain was still achieved for many frames after the SF. As a matter of fact, for clips such as SlideEditing and the Vidyo clips, we observed an average PSNR gain of well over 1dB for the entire clip after the SF, containing hundreds of frames.

Some comparisons of the SF and the FS (aka the enhanced SF) for different types of test clips are given in Figures 5 - 7. As can be seen from the figures, the proposed algorithm was able to introduce improvements in the subjective quality of key areas of the decoded frames (e.g. the face of the man in the green shirt, the texture on the wall, the lines on the court in Figure 5, the faces in Figure 6 and Figure 7, and etc.) that included both static backgrounds and moving objects (e.g. the basketball player).

As mentioned previously, the side information required from the encoder in our implementation was the PSNR for the SF after encoding as the first IDR frame of the low quality segment. This corresponds to a total of 16 bits using natural binary representation without entropy coding, and was negligible. Therefore, the PSNR gains reported reflect the “net” gains considering both the PSNR and the bitrate.

In terms of complexity, because the proposed processing was carried out for only one frame of the low quality segment, even though the decoding process involves ME and calculations of SAD/SSD, the increase to the complexity of the decoding of SF is still reasonable, and lower than that for HEVC encoding of a similar frame. This is because processing required for the HEVC encoding for transform, quantization, the bulk of the processing for mode decision, and the deblocking filter are not necessary for enhanced decoding. Averaged for all frames in the low quality segment, the increase is modest considering the potential gain in PSNR and subjective quality achieved.

Finally, we analyzed the clips for which a PSNR gain was not achieved in Table I. Figures 8 and 9 show two typical examples. In Figure 8, subjective quality improvements *were* achieved (e.g. on the table cloth and in the background), even though the subjective quality improvements were not reflected in the PSNR. This might have been due to small mis-alignments of some pixels that might not be visible, but still have caused the PSNR to degrade. On the other hand, Figure 9 shows a case where although visible subjective improvements were achieved for both static (e.g. the background, the Christmas tree, the teddy bear) as well as moving (e.g. the face of the sitting girl, the skirt and the right leg of the running girl) areas, some relatively large mis-aligned/matched patches (e.g. in the areas near the hair of the running girl) led to an overall PSNR loss. We notice that such mis-alignments are visually similar to artifacts created by erroneously


Vidyo
Natural. Affordable. Universal.
HD Quality Video Conferencing

(a) Standard Decoder

(b) Enhanced Decoder

(c) Details Comparison

Fig. 6: Subjective Quality Comparison for Video Conferencing Clip - FourPeople

received motion vectors when video bitstreams are sent over error prone networks. Therefore, techniques developed for error concealment of such artifacts may be helpful in remedying such PSNR losses while preserving the gain in other areas under the proposed enhanced decoding framework [17].

In the current implementation, the value for $Th_Y$ for higher motion areas was selected from the range between 5 and 53 based on the clip and bitrate . The values were listed in Table I. We noticed that the value for most clips were 5, while for other clips, one might be able to determine the value by estimating the decoded PSNR (e.g. with a technique similar to that in [26]).

## V. Conclusions and Future Work

In this paper, we describe a cross-segment decoder for improving the quality of HEVC decoding. The algorithm utilizes information that is 1) available to the decoder and 2) that might not have been utilized by the encoder for improving the quality of the decoded video sequence, especially for applications where the bitstream has been produced with limited encoding resources and/or the visual quality of the bitstream varies due to reasons such as network based adaptation. The same algorithm could also be used for AVC decoding.

Experimental results using the HEVC test clips showed significant PSNR and/or visual quality improvements for a relatively wide variety of test clips. Because only one frame needs to be processed by the enhanced decoder, after which the bitstream is decoded with a standard decoder, the complexity of the proposed system is very low without incurring loss of compression efficiency and significant delay.

Further areas of improvement including more precise and adaptive segmentation of the SF, intelligent determination of the various thresholds, especially $Th_Y$, and techniques for correcting mis-

| | QP | $Th_Y$ | Gain-Start Frame (dB) | Gain-30 Frames (dB) | Gain-60 Frames (dB) | Avg PSNR (dB)<br>1st/30/60 |
|---|---|---|---|---|---|---|
| BasketballPass | 34 | 7 | 0.68 | 0.24 | -0.51 | 34.66/33.47/33.05 |
| | 35 | 5 | 0.56 | 0.17 | 0.02 | 34.08/32.92/32.48 |
| | 36 | 5 | 0.34 | 0.06 | 0.01 | 33.43/32.33/31.91 |
| | 38 | 13 | 0.86 | 0.29 | 0.11 | 32.16/31.22/30.81 |
| | 39 | 9 | 0.63 | 0.19 | 0.07 | 31.61/30.64/30.27 |
| | 40 | 9 | 0.38 | 0.16 | 0.06 | 31.07/30.22/29.80 |
| ChromaKey | 34 | 5 | 0.35 | -0.03 | -0.08 | 36.98/35.57/34.85 |
| | 35 | 5 | 0.23 | -0.13 | -0.16 | 36.46/35.12/34.37 |
| | 36 | 5 | 0.46 | 0.03 | -0.05 | 35.95/34.59/33.84 |
| | 38 | 5 | 0.63 | 0.05 | -0.01 | 34.97/33.60/32.81 |
| | 39 | 5 | 0.90 | 0.20 | 0.09 | 34.41/33.07/32.30 |
| | 40 | 5 | 0.78 | 0.08 | 0.01 | 34.02/32.60/31.81 |
| FourPeople | 34 | 15 | 0.96 | 0.77 | 0.59 | 37.44/36.66/36.62 |
| | 35 | 5 | 1.19 | 0.88 | 0.71 | 36.82/36.11/36.06 |
| | 36 | 5 | 1.49 | 1.16 | 0.96 | 36.23/35.55/35.48 |
| | 38 | 5 | 1.72 | 1.26 | 1.09 | 34.93/34.36/34.29 |
| | 39 | 5 | 1.84 | 1.36 | 0.78 | 34.27/33.74/33.66 |
| | 40 | 7 | 2.05 | 1.52 | 1.34 | 33.59/33.09/33.01 |
| Johnny | 34 | 5 | 0.63 | 0.36 | 0.25 | 38.90/38.17/38.13 |
| | 35 | 5 | 1.09 | 0.61 | 0.4 | 38.37/37.68/37.63 |
| | 36 | 5 | 1.08 | 0.65 | 0.51 | 37.87/37.21/37.15 |
| | 38 | 5 | 1.47 | 0.84 | 0.69 | 36.70/36.16/36.06 |
| | 39 | 5 | 1.53 | 0.89 | 0.71 | 36.19/35.66/35.58 |
| | 40 | 5 | 1.50 | 0.81 | 0.65 | 35.58/35.10/35.01 |
| SlideEditing | 34 | 27 | 2.50 | 1.93 | 1.55 | 35.96/36.26/36.24 |
| | 35 | 45 | 2.66 | 2.13 | 1.78 | 35.04/35.24/35.17 |
| | 36 | 47 | 2.67 | 2.11 | 1.75 | 34.18/34.42/34.38 |
| | 38 | 19 | 2.81 | 2.40 | 2.00 | 32.18/32.37/32.31 |
| | 39 | 23 | 2.79 | 2.38 | 1.99 | 31.23/31.44/31.40 |
| | 40 | 41 | 2.67 | 2.26 | 1.90 | 30.37/30.52/30.44 |
| KristenAndSara | 34 | 5 | 0.57 | 0.37 | 0.31 | 38.47/37.77/37.69 |
| | 35 | 5 | 0.81 | 0.54 | 0.46 | 37.90/37.25/37.16 |
| | 36 | 5 | 1.18 | 0.71 | 0.62 | 37.32/36.71/36.61 |
| | 38 | 5 | 1.40 | 0.92 | 0.8 | 36.09/35.57/35.48 |
| | 39 | 7 | 1.38 | 0.87 | 0.75 | 35.54/35.03/34.45 |
| | 40 | 7 | 1.38 | 0.92 | 0.8 | 34.95/34.45/34.35 |
| Vidyo1 | 34 | 5 | 1.11 | 0.77 | 0.62 | 38.71/38.02/38.00 |
| | 35 | 5 | 1.23 | 0.81 | 0.68 | 38.13/37.48/37.46 |
| | 36 | 5 | 1.48 | 0.95 | 0.78 | 37.59/36.94/36.91 |
| | 38 | 9 | 1.66 | 1.07 | 0.89 | 36.33/35.79/35.74 |
| | 39 | 5 | 1.80 | 1.17 | 0.98 | 35.77/35.22/35.18 |
| | 40 | 5 | 1.67 | 1.08 | 0.91 | 35.15/34.65/34.62 |
| Vidyo3 | 34 | 7 | 0.19 | 0.23 | 0.24 | 38.42/37.32/37.33 |
| | 35 | 7 | 0.42 | 0.35 | 0.38 | 37.79/36.72/36.73 |
| | 36 | 7 | 0.62 | 0.49 | 0.51 | 37.15/36.10/36.11 |
| | 38 | 7 | 0.96 | 0.67 | 0.64 | 35.87/34.89/34.89 |
| | 39 | 5 | 1.00 | 0.75 | 0.71 | 35.18/34.24/34.23 |
| | 40 | 5 | 1.04 | 0.76 | 0.71 | 34.54/33.65/33.63 |
| FlowerVase | 34 | 5 | -0.10 | -0.44 | -0.53 | 39.16/37.36/36.70 |
| | 35 | 5 | -0.05 | -0.39 | -0.49 | 38.52/36.79/36.11 |
| | 36 | 5 | 0.28 | -0.26 | -0.36 | 37.89/36.19/35.50 |
| | 38 | 5 | 0.46 | -0.07 | -0.18 | 36.52/34.99/34.30 |
| | 39 | 5 | 0.53 | -0.04 | -0.17 | 35.94/34.41/33.71 |
| | 40 | 5 | 0.56 | 0.04 | -0.10 | 35.31/33.86/33.16 |
| ChinaSpeed | 34 | 13 | -2.12 | -0.65 | -0.38 | 36.45/34.16/33.96 |
| | 35 | 29 | -1.66 | -0.63 | -0.41 | 35.70/33.50/33.31 |
| | 36 | 19 | -1.31 | -0.28 | -0.15 | 35.02/32.83/32.64 |
| | 38 | 9 | -0.71 | -0.13 | -0.01 | 33.58/31.44/31.28 |
| | 39 | 21 | -0.32 | 0.03 | 0.11 | 32.66/30.73/30.60 |
| | 40 | 11 | -0.33 | -0.20 | -0.01 | 32.10/30.07/29.96 |
| **Avg Gain** | | | **0.91 (dB)** | **0.60 (dB)** | **0.47 (dB)** | |

TABLE I: PSNR Improvement. The value under the QP column is the QP value for the first frame of the low quality segment (aka the SF) was set to 5 levels higher.

aligned/matched patches.

Finally, the reason why the gain in PSNR for the SF may be lost after decoding a large number of frames is due to mismatches between the residual when the enhanced SF (aka the FS) is used as the reference frame and the residual presented in the bitstream. This mismatch might be corrected with information proactively sent by the encoder with the knowledge that enhanced decoding is carried out by the decoder. Such information may be useful in scenarios where complexity and quality scalabilities are desired.

## VI. Acknowledgement

The authors wish to thank Xuan Dong, Meiyuan Fang, Huiwen Chang and Tong Shen of Tsinghua University for their contributions to the work. The work was partially supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61125102), the State Key Program of National Natural Science of China (Grant No. 61133008) and the National Significant Science and Technology Projects of China (Grant No. 2012ZX01039-001-003-2).

## References

[1] T.Wiegand, G.J.Sullivan, G.Bjontegaard, and A.Luthra, "Overview of the h.264/avc video coding standard," in *IEEE Trans Circuits and Systems for Video Technology*, 2003, pp. 560–576.

[2] G.Sullivan, J.Ohm, W.Han, and T.Wiegand, "Overview of the high efficiency videocoding (hevc) standard," in *IEEE Trans Circuit and System for Video Technology*, 2012.

[3] J.Ohm, G.Sullivan, H.Schwarz, T.Tan, and T.Wiegand, "Comparison of the coding efficiency of video coding standards including high efficiency video coding (hevc)," in *IEEE Trans Circuit and System for Video Technology*, 2012.

[4] Y. Yuan, X. Zheng, L. Liu, X. Cao, and Y. He, "Non-square quadtree transform structure for hevc," in *Picture Coding Symposium (PCS)*, 2012, pp. 505–508.

[5] F. Bossen, V. Drugeon, E. Francois, J. Jung, S. Kanumuri, M. Narroschke, H. Sasai, J. Sole, Y. Suzuki, T. Tan *et al.*, "Video coding using a simplified block structure and advanced coding techniques," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 12, pp. 1667–1675, 2010.

[6] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "Hevc complexity and implementation analysis," 2012.

[7] G. Correa, P. Assuncao, L. Agostini, and L. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 4, pp. 1866–1874, 2011.

[8] J. Ohm, G. Sullivan, H. Schwarz, T. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards–including high efficiency video coding (hevc)," *see pp.[INSERT PAGE NUMBER] of this issue*.

[9] M. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "Hevc: The new gold standard for video compression: How does hevc compare with h. 264/avc?" *Consumer Electronics Magazine, IEEE*, vol. 1, no. 3, pp. 36–46, 2012.

[10] M. Cassa, M. Naccari, and F. Pereira, "Fast rate distortion optimization for the emerging hevc standard," in *Picture Coding Symposium (PCS), 2012*. IEEE, 2012, pp. 493–496.

[11] J. Vanne, M. Viitanen, T. Hamalainen, and A. Hallpuro, "Comparative rate-distortion-complexity analysis of hevc and avc video codecs," 2012.

[12] N. Farber and B. Girod, "Robust h.263 compatible video transmission for mobile access to video servers," in *in Proceedings of the Picture Coding Symposium*, 1996, pp. 575–578.

[13] U. Jennehag and S. Pettersson, "On synchronization frames for channel switching in a gop-based iptv environment," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*. IEEE, 2008, pp. 638–642.

[14] P. Ni, D. Isovic, and G. Fohler, "User-friendly h. 264/avc for remote browsing," in *Proceedings of the 14th annual ACM international conference on Multimedia*. ACM, 2006, pp. 643–646.

[15] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," in *IEEE Trans Circuits and Systems for Video Technology*, 2007, pp. 1103–1120.

[16] Y.Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," in *IEEE Signal Processing Magazine*, 2000, pp. 61–82.

[17] Y.Wang and Q. Zhu, "Error control and concealment for video communication: A review," in *Proc. of the IEEE*, 1998, pp. 974–997.

[18] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media playout for low-delay video streaming over error-prone channels," in *IEEE Trans. Circuits and Systems for Video Technology*, 2010, pp. 841–851.

[19] J. Han, V. Melkote, and K. Rose, "Estimation-theoretic delayed decoding of predictively encoded video sequences," in *Proc. of the IEEE Data Compression Conference (DCC)*, 2010, pp. 119–128.

[20] M. Naccari, C. Brites, J. Ascenso, and F. Pereira, "Low complexity deblocking filter perceptual optimization for the hevc codec," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 737–740.

[21] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 614–619, 2003.

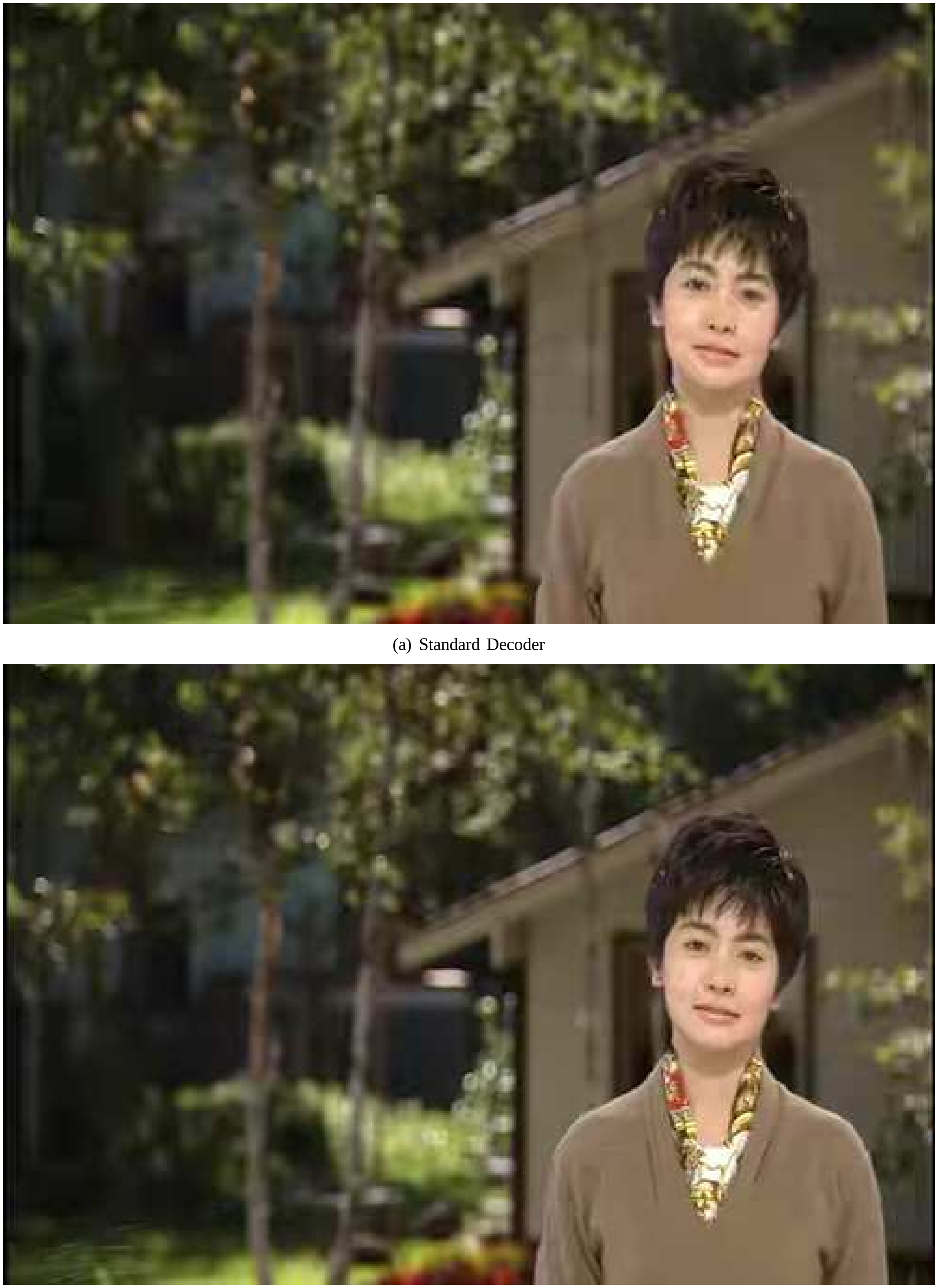
[22] S. Shih, C. Chang, and Y. Lin, "A near optimal deblocking filter for h. 264 advanced video coding," in *Design Automation, 2006. Asia and South Pacific Conference on*. IEEE, 2006, pp. 6–pp.

[23] L. M. E. Al, "X264: A high performance h.264/avc encoder," 2006.

[24] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," in *IEEE Signal Processing Magazine*, 1998, pp. 74–90.

[25] X. Li, M. Wien, and J. Ohm, "Rate-complexity-distortion optimization for hybrid video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 7, pp. 957–970, 2011.

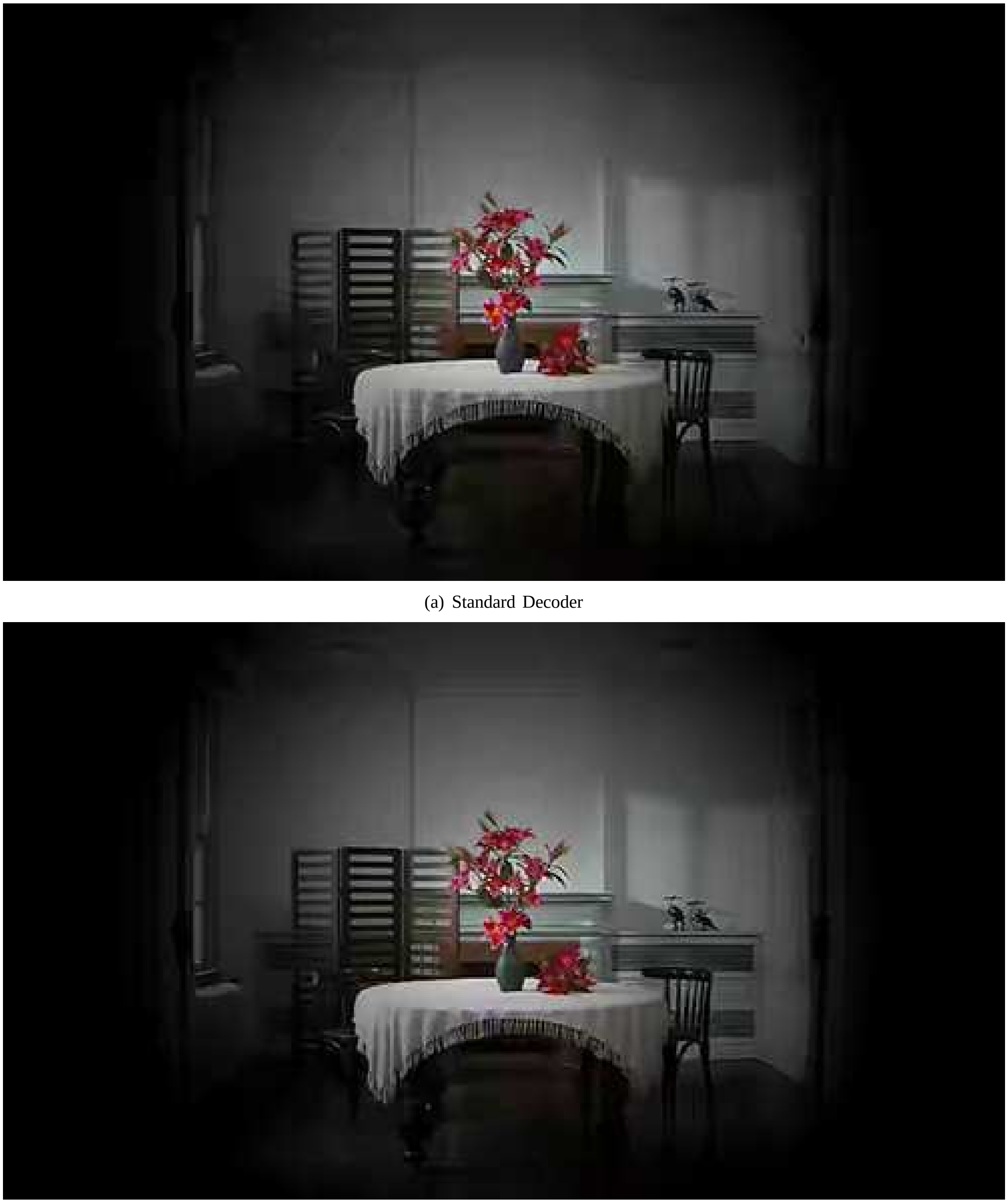
[26] T. Brandao and M. P. Queluz, "No-reference psnr estimation algorithm for h.264 encoded video sequences," in *Proc. of the 16th EUSIPCO*, 2008.



(a) Standard Decoder

(b) Enhanced Decoder

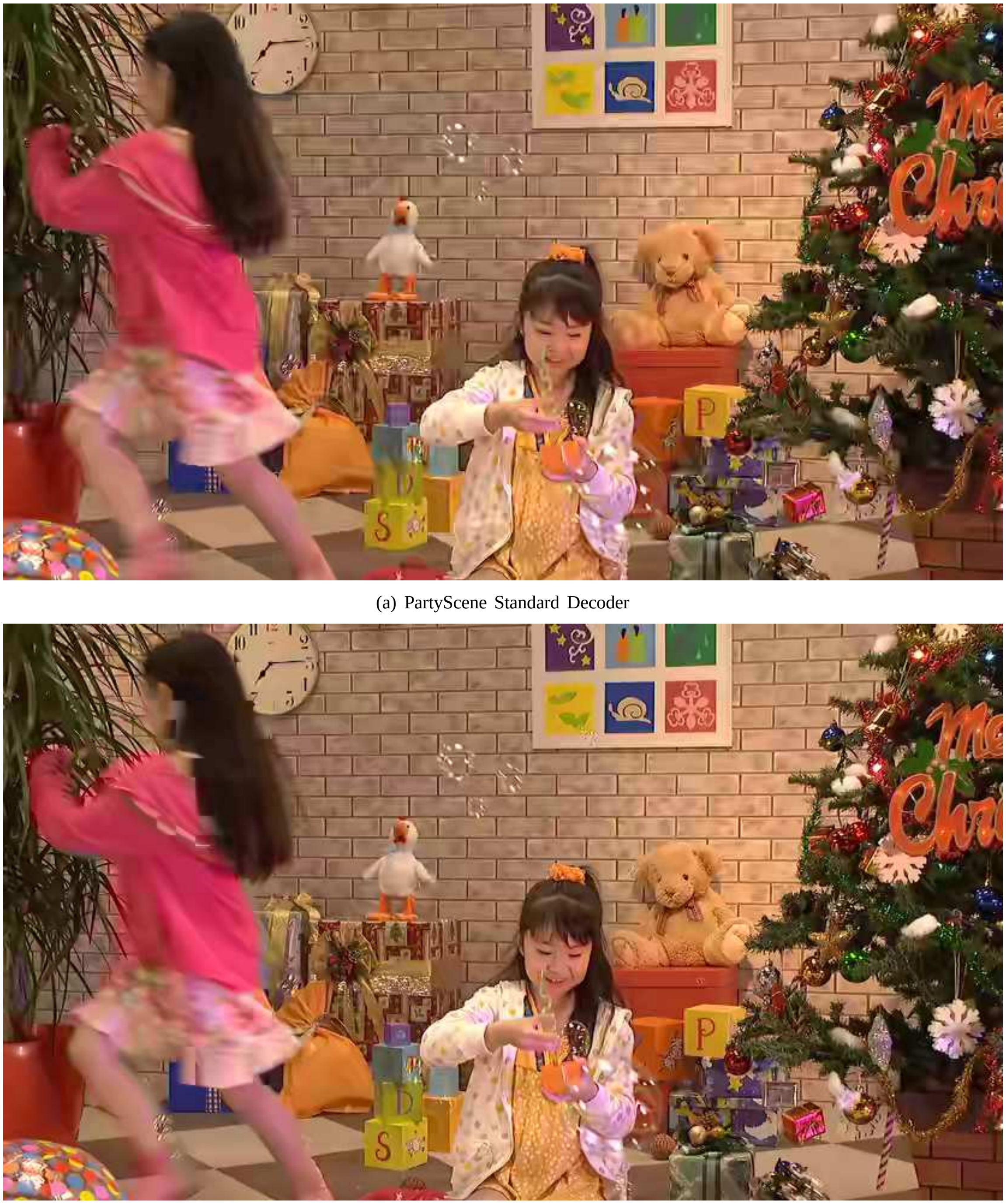
Fig. 7: Subjective Quality Comparison for Motion Clip - Chromakey



(a) Standard Decoder

(b) Enhanced Decoder

Fig. 8: An Example with Subjective Quality Gain but PSNR Loss after Enhancement - FlowerVase



(a) PartyScene Standard Decoder

(b) PartyScene Enhanced Decoder

Fig. 9: An Example of Mis-aligned Patches Resulting in PSNR Losses - PartyScene