

“© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Error Concealment for Cloud-based and Scalable Video Coding of HD Videos

Muhammad Usman, *Student Member, IEEE*, Xiangjian He, *Senior Member, IEEE*,  
Kin-Man Lam, *Senior Member, IEEE* Min Xu, *Member, IEEE*, Syed Mohsin Matloob Bokhari, *Member, IEEE*  
Jinjun Chen, *Senior Member, IEEE* and Mian Ahmad Jan, *Member, IEEE*

**Abstract**—The encoding of HD videos faces two challenges: requirements for a strong processing power and a large storage space. One time-efficient solution addressing these challenges is to use a cloud platform and to use a scalable video coding technique to generate multiple video streams with varying bit-rates. Packet-loss is very common during the transmission of these video streams over the Internet and becomes another challenge. One solution to address this challenge is to retransmit lost video packets, but this will create end-to-end delay. Therefore, it would be good if the problem of packet-loss can be dealt with at the user’s side. In this paper, we present a novel system that encodes and stores the videos using the Amazon cloud computing platform, and recover lost video frames on user side using a new Error Concealment (EC) technique. To efficiently utilize the computation power of a user’s mobile device, the EC is performed based on a multiple-thread and parallel process. The simulation results clearly show that, on average, our proposed EC technique outperforms the traditional Block Matching Algorithm (BMA) and the Frame Copy (FC) techniques.

**Index Terms**—HD, End-to-End Delay, Error Concealment, Block Matching, Test-Zone Search, Frame Copy

## I. INTRODUCTION

THE growth of High Definition (HD) video streaming over the Internet has increased tremendously due to the adoption of high speed fixed and mobile transmission media, easy availability of smartphone devices at affordable prices and high demand for access to the Internet [1]. Designing a multimedia system over heterogeneous transmission networks faces many challenges, related to fluctuating bandwidths, Packet Loss Rates (PLRs) in the transmission channels and capabilities of user’s devices. Scalable Video Coding (SVC) is considered as one of the efficient approaches to address these challenges [2]. Due to its popularity, Joint Video Team of Video Coding Experts Group (JVT-VCEG) has standardized a scalable extension of H.265/HEVC, formally known as SHVC

Manuscript is submitted on June 16, 2016. (Corresponding author: Xiangjian He.)

Muhammad Usman, Prof. Xiangjian He, Dr. Min Xu and Associate Prof. Jinjun Chen are with Global Big Data Technologies Center (GB-DTC), School of Electrical and Data Engineering, University of Technology Sydney, Australia. (E-mail: Muhammad.Usman-2@student.uts.edu.au, Xiangjian.He@uts.edu.au, Min.Xu@uts.edu.au, Jinjun.Chen@uts.edu.au).

Dr. Syed Mohsin Matloob Bokhari is with Department of Electronic Engineering, University of Engineering and Technology Peshawar, Pakistan (E-mail: Mohsin.Bokhari@uetpeshawar.edu.pk)

Prof. Kin-Man Lam is with Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (E-mail: enkmlam@polyu.edu.hk)

Dr. Mian Ahmad Jan is with Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan (E-mail: mianjan@awkum.edu.pk)

[3], [4]. SHVC enables the Video Service Providers (VSPs) to encode videos in parts, based on spatial, temporal and signal-to-noise-ratio encoding settings. Due to its flexibility, SHVC is considered to be an ideal approach for VSPs to provide streaming services in real-time to mobile users who have limited processing capabilities and fluctuating bandwidths.

Emerged cloud computing platforms facilitate the designing of next-generation video streaming systems. A cloud platform provides cost-effective, reliable and scalable services. Media cloud platforms are becoming popular to provide video streaming services. Recently, there have been little research exploring how a cloud platform can be utilized efficiently for video streaming services in [5]–[8]. To meet end-user’s demands for real-time services, a VSP can rent cloud resources, including storages and virtual CPUs, in order to avoid the investment on hardware and its maintenance. Fig. 1 shows an old-fashioned scenario, in which video streams are transmitted from cloud servers to end-users. In this scenario, the bit-rates (or resolution) of video streams can only be adjusted on cloud servers for smooth decoding of the videos on user sides.

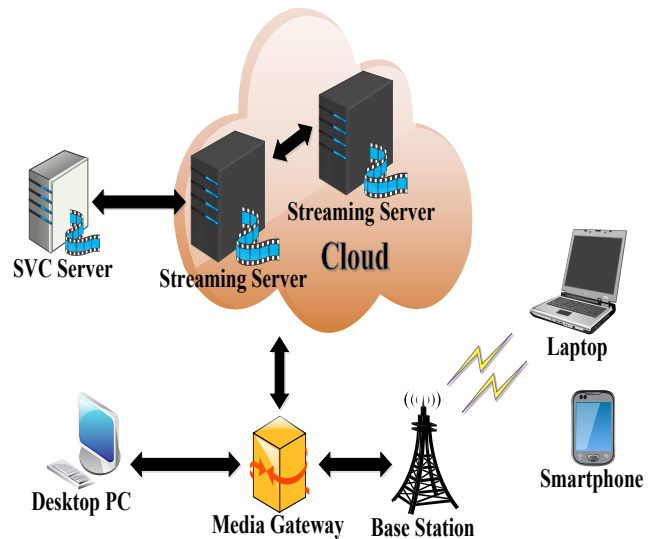


Fig. 1. The Old Fashioned Scenario

Cloud computing schemes provide a VSP an easy way to utilize computing and storage resources, in order to tackle dynamic needs of mobile users. Therefore, cloud data centers have been able to provide real-time video services in recent years. Nowadays, many VSPs are switching their infrastructure

to cloud platforms, such as YouTube. There are many studies utilizing mobile cloud computing and cloud storage resources to deal with mobile user's needs across the globe [9], [10]. This has happened due to the availability of public clouds, which help VSPs to improve Quality of Service (QoS) at a lower cost, by creating multiple agents to deal with a user's requests in real-time. However, in order to effectively and efficiently use a cloud platform, we need to take into account the factors of fluctuating wireless channels and mobility of users.

When burst errors occur in heterogeneous transmission networks either at bit or packet levels, lost/corrupted video frames need to be concealed. There are many solutions to deal with and minimize the effects of transmission errors such as error correction codes, retransmission of data and Error Concealment (EC) techniques.

The error correction codes can be used only if the data packet is successfully arrived with errors. These codes cannot fix the issue of packet-loss.

Retransmission can be considered as a suitable solution when the latency between the server and the client is small enough, and this is possible only if both the server and the client belong to a small local network. In the case of the Internet with remote cloud servers, retransmission of multiple data packets increases the end-to-end delay (cloud server delay, transmission delay and user device delay). Therefore, in real-time scenarios, retransmission is never considered as an optimal solution.

Error Concealment (EC) techniques attempt to recover the lost information using current and previously processed data. It is true that many VSPs are using cloud platforms for video transcoding. A cloud platform can only offer storage and computing resources but it cannot control transmission errors. There are three main reasons to justify the importance of an EC scheme for a VSP to maintain/improve its Quality of Experience (QoE). First, transmission error in the form of packet-drop can easily bring down the reputation of any VSP. Second, live video streaming is mostly simplex transmission i.e., it is of one way transmission, and it does not involve any feedback from a user device, e.g. a television. Third, an extra server will be required to deal with user complaints and requests for retransmissions of specific video frames. Based on these three reasons, the importance of EC in any video streaming/transmission scheme over unreliable networks, e.g. the Internet, cannot be ignored.

Until now, there are very few studies for EC on scalable video streams. Frame Copy (FC) technique is commonly used to conceal the lost/damaged video frames [11]. FC does not use any searching algorithm to find out motion vectors. It replaces a missing video frame with a previously processed video frame. However, FC technique produces high concealment errors in videos containing moving objects. Another popular EC technique is classic Block Matching Algorithm (BMA) [12]. BMA technique uses previously processed video frames to conceal the missing one. Many searching algorithms, such as Three Step Search (TSS), Four Step Search (FSS), Adaptive Root Pattern Search (ARPS), Logarithmic Search (LS) and Exhaustive or Full Search (FS) [13], have been proposed to work with a BMA and find the best match in

the previously processed video frames. The most accurate algorithm is FS. However, it is a computationally intensive algorithm. Another recent searching algorithm is Test Zone Search (TZS) [4]. It is a hybrid technique and is a combination of diamond and raster searches. TZS produces the same level of accuracies but has less computational complexity compared to the FS. The existing EC techniques are always applied on user's sides, but they ignore the processing capability of user's devices.

In this paper, we propose a novel EC technique for SHVC-generated video streams, encoded on a cloud platform. The proposed EC technique conceals both Intra and Inter video frames of HD and Ultra HD (UHD) videos lost during a transmission. The contributions of this papers are summarized as follows.

- The proposed EC approach does not require any alteration in the decoder architecture. It works as a standalone application, and is called whenever required. It does not demand a support from the encoder, but it utilizes the processing capability of user's devices.
- The proposed approach involves a parallel processing strategy. Unlike traditional parallel processing that completely process all parallel tasks, our approach terminates the tasks based on conditional thresholds. Therefore, it produces minimum processing load on user's devices and is capable to support real-time video processing.
- The proposed approach is energy efficient. Due to the use of conditional thresholds, it does not produce a heavy processing load, so it saves the battery backup time.

The work in this paper shows its significance because currently there is a dearth of literature that enhances the performance of cloud based video distribution with the aid of client side (error concealment) techniques. The proposed scheme targets on better quality videos at the client side, and thus helps in providing overall a better cloud service. The experiments show that the proposed approach produces better visual quality of videos in terms of average Peak-Signal-to-Noise-Ratio (PSNR) and average Structural SIMilarity (SSIM) index.

The rest of this article is organized as follows. Section II demonstrate the related work on multimedia management on a cloud platform, SVC schemes and EC techniques. In Section III, the proposed algorithm is introduced. Section IV contains experimental set-up and results. Finally, the paper is concluded in Section V.

## II. RELATED WORK

It is widely acknowledged that efficient video distribution is a major challenge for contemporary Cloud Service Providers (CSPs) and VSPs. Most of the literature in this area addresses methods and techniques in the Cloud Server side and very few have investigate the use of the Client Side techniques. In the following subsections, we discuss multimedia management on cloud platform, SVC to make the video streaming smooth between a CSP and end users, and EC techniques to improve the QoE of the cloud multimedia services.

### A. Multimedia Management

The current era is of big data, especially big video data. Management of such data is always complex and requires massive processing and huge storage space. A user friendly approach to manage big video data was presented using the Hadoop platform along with the MapReduce paradigm in [5]. Another similar approach was presented in [6] and focused on reducing network overhead. To deal with the storage and processing of large media databases, MapReduce was applied along with a Java-based framework for Hadoop streaming in [8]. Cloud platform was used to efficiently deal with video processing in [14], [15]. All of the approaches presented in [5], [6], [8], [14], [15] focused on the cloud side, i.e., the server side and ignored the packet loss problems.

### B. Scalable Video Coding

To deal with transmission errors, SVC has been considered as a simple and straight-forward approach. By considering the fluctuations in the wireless channels, an adaptive SVC-based streaming was presented for mobile users in [16]. This technique reduces the effect of video freezing, but does not bring noticeable improvement in terms of visual quality. To support the UHD video transmission using SHVC, a study was presented in [17], which considers the need for coding efficiency before transmission. A similar analysis on the impact of packet loss in SVC streams during the transmission was presented in [18]. The analysis presented in [17], [18] helped to adjust the encoding parameters before transmission. A new framework based on digital and analogue coding for SVC over wireless channels was proposed in [19] to reduce the transmission error effects. Although the performance was good in terms of visual quality, the presented technique was applied on low-resolution videos only.

### C. Error Concealment

The EC techniques are broadly classified into three major domains, i.e., Intra/Spatial, Inter/Temporal and Hybrid [20]. In the inter/temporal EC techniques, Motion Vectors (MVs) play a pivotal role in the ME process. They represent a distance between a matching pixel or a block of matching pixels in the current and the reference video frames [3], [4]. There are many approaches to compute this distance, such as Sum of Absolute Difference (SAD), Zero-mean Sum of Absolute Difference (ZSAD), Minimum Absolute Difference (MAD) and Normalized Cross Correlation (NCC). These distance measures involve multiple mathematical operations, such as addition, subtraction, multiplication, division and square root [21]. In [22], kernel-based signal reconstruction was performed in the spatial EC domain. Although the technique was developed for Intra-coded video frames with an improved PSNR, it was tested on simple images rather than on videos. Boundary matching score in combination with object detection was used to perform temporal EC in [23]. This technique produced better results only in the presence of multiple objects. Iterative dynamic programming was used in [24] to recover the MVs of lost video frames in the temporal domain. This technique

worked efficiently at the cost of heavy computations. Both error resilience and concealment were used together to support the robust transmission of scalable encoded videos over heterogeneous networks in [25]. This technique produced better quality, but it demanded a high bandwidth.

## III. SCALABLE ERROR CONCEALMENT

In this section, we propose a scalable EC (SEC) scheme. The EC scheme is based on the idea of Frame Interpolation (FI) proposed in [26], and is motivated by the traditional BMA and FC algorithms. FI is a pixel-level technique and is used to insert intermediate video frames between the existing video frames. Its purpose is to either slow down the motion of objects or support fast frame rates. This feature is quite common in modern displays, such as Phillips, Samsung, Sony and LG [27]. The videos are stored on a cloud platform and encoded in a scalable style. In an SHVC, multiple streams are generated. One stream is known as the Base Layer (BL), while the remaining streams are called Enhanced Layers (ELs). A BL is always encoded in an Intra mode, while ELs are encoded in an Inter mode [4]. A generic diagram for cloud-based scalable video streaming is shown in Fig. 2. The encoded videos are stored in the video storage center. Videos are encoded on various transcoding servers, under the supervision of a pipeline server. The streaming server is responsible to deal with user's requests for videos.

The videos are transmitted over the Internet and face transmission errors. The transmission errors cause frame loss. The proposed SEC technique recovers the lost video frames in HD videos. In the lost video frames, MVs are derived first. The process of estimating the lost MVs is known as ME. Usually the ME process is performed on the encoder side and consumes 40% to 80% of the encoding time [28]. In SEC, the main targets are to achieve less computational time and improved visual quality. The less computational time supports real-time video processing and communication and the improved visual quality helps in maintaining the QoE. To recover the MVs of a lost video frame in either stream, the ME process considers two video frames at a time, i.e., the previous and the next video frames. The blocks in the previous and the next video frames represent heads and tails of the estimated MVs. A tail represents the starting point of an MV, and the head represents the ending point as shown in Fig. 3. The estimated MVs help in recovering the lost video frames. In the end, the estimated video frames are further processed by an adaptive filter to remove blurring and hair-line effects produced during the video frame recovery process. All these tasks are performed by our proposed SEC technique, whenever called by an SHVC decoder as shown in Fig. 4. The step-by-step procedure of our SEC scheme is summarized in Subsections III-A to III-G.

### A. Reference Frame Selection For Enhancement Layer

In the case of a BL video frame concealment, the steps shown in Subsections III-B to III-G are followed. However, the case of an EL video frame concealment is slightly different. EL is an Inter-frame based. To conceal a missing video frame in an

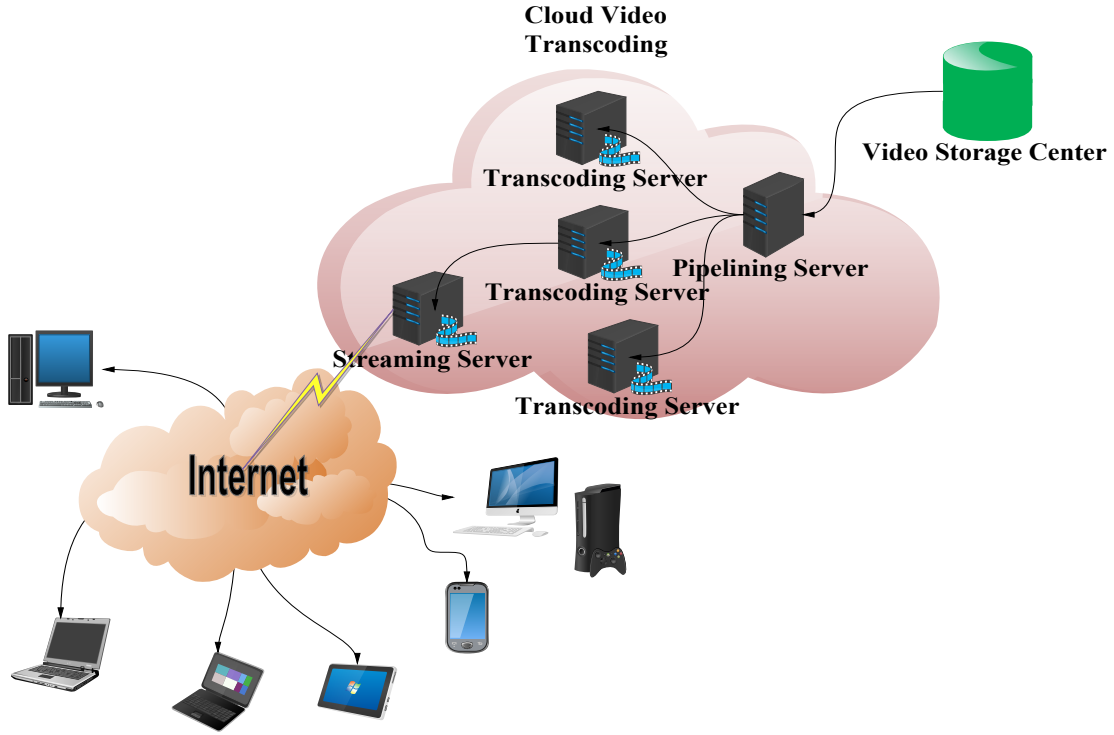


Fig. 2. Cloud-Hosted and Assisted Scalable Video Streaming

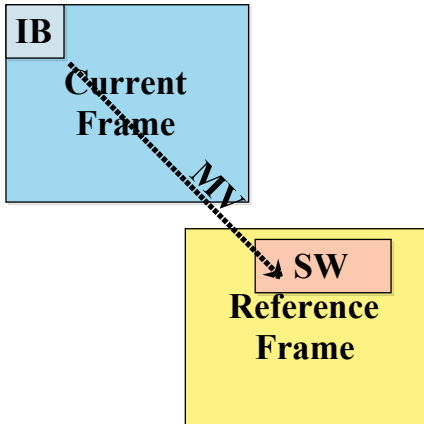


Fig. 3. The Motion Vector

EL stream, we first need to find the master BL or the EL video frames, on which the missing video frame is dependent on. Care must be taken at this stage, as the input video sequences are of HD or UHD, and are compressed significantly in the EL streams [4]. A flag in the header of an EL video frame indicates whether it is predicted from a BL or an EL frame. Therefore, the first step in this phase is to find out the master video frames for a missing video frame. The block diagram of an SHVC decoder is shown in Fig. 5. The *DEMUX* extracts the EL video frames from an incoming video stream. The *Inter Layer Processing* gives us a knowledge about the dependency of an EL video frame on the BL or the EL video frames.

In our proposed approach, we modify the decoder to extract the information about the dependencies of EL video frames and forward it to our SEC scheme. In the SEC scheme, the first step is to determine whether the missing video frame is from the BL or the EL stream. If the missing video frame belongs to the BL stream, then steps from Subsections III-B to III-G are followed. If the missing video frame belongs to the EL stream, the information received from the decoder is utilized to find the master or reference video frames for the missing video frame. If the master video frame is from the BL stream, then only one video frame will be involved. In that case, the SEC scheme follows the steps from Subsections III-B to III-G. If the master video frame is from the EL stream, then it means that more than one master frames are involved. Usually, the SEC scheme keeps copies of at least three previously processed EL video frames. The SEC scheme first finds the master video frames of the three previously processed EL video frames. Once the master video frames are found, The SEC scheme computes an average of intensities in those master video frames. This computed average is used to produce a master video frame for the missing EL video frame, as shown in Fig. 6.

### B. Block Size Selection

During this phase, the SEC scheme first divides every video frame into  $n$  blocks of equal size. In our experiments, a block size of  $16 \times 16$  is selected. The main reason to choose this block size is to make the SEC scheme compatible with prior standards where the maximum supported block size is  $16 \times 16$  [29].

Applying threads is very common in the video coding

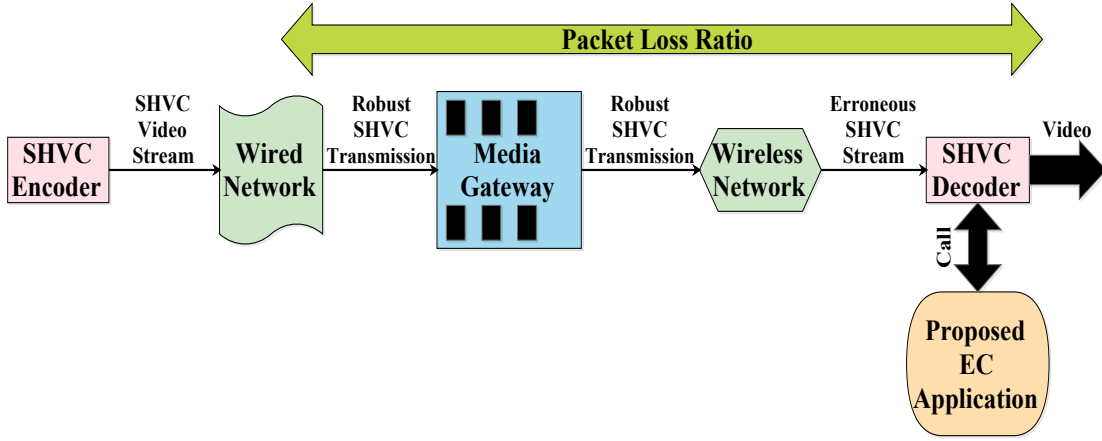


Fig. 4. Cloud-Hosted and Assisted Scalable Video Streaming

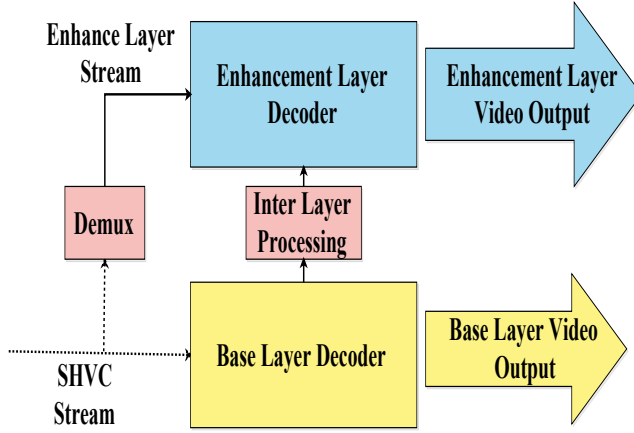


Fig. 5. SVC Decoder

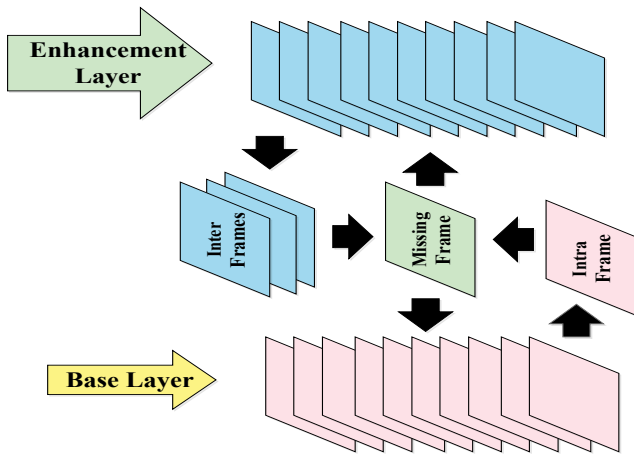


Fig. 6. Master Frame Referencing

and decoding, as it is always done to speed up the computations drastically. When the Motion Estimation (ME) is applied on the blocks of a video frame, it can be done in parallel by generating multiple threads. These threads can

be generated using any appropriate underlying thread library, such as *pthread* (C/C++), *java.lang.Thread* (JAVA/Andorid) or *Windows Threads*, and assigned to different blocks in the video frame to process them simultaneously.

In the SEC scheme, the Input Video (IV) stream is processed on frame by frame basis rather than as a group of video frames. The SEC scheme always keeps copies of the recently processed video frames. If frame  $N$  of a video is lost during a transmission, it is immediately notified through the sequence numbers stored in the header of other video frames previously received. The SEC scheme brings back the  $(N - 1)^{th}$  video frame and waits for the  $(N + 1)^{th}$  video frame to arrive. Once the  $(N + 1)^{th}$  video frame arrives, the concealment procedure starts immediately by partitioning the  $(N + 1)^{th}$  video frame into blocks of size  $16 \times 16$ . The  $(N + 1)^{th}$  video frame is considered as the current video frame and its blocks are denoted as Input Blocks (IBs), while the  $(N - 1)^{th}$  video frame is considered as the reference video frame and its blocks are denoted as Reference Blocks (RBs).

#### C. Construction of Searching Windows for Block Matching

In this phase, for each input block of size  $16 \times 16$ , a searching window (SW) of size  $64 \times 64$  is constructed in the reference video frame in such a way that the location of the input block is located in the middle of the corresponding searching window. All SWs have the same size. In determining the size of SWs, we assume that the motion in the consecutive video frames is not very fast. Therefore, we set the size of an SW to be four times (i.e.,  $64 \times 64$ ) of the size of an IB [29].

#### D. Searching Patterns

In this phase, a process to find the best match through searching is performed between the current and the reference video frames. This process works on the block level. The best match for blocks in the current video frame is searched in the reference video frame.

In the SEC scheme, TZS is modified. By default, TZS starts with an 8-point Diamond Search (DS). After finding a best match, it still goes further for three more rounds. In the SEC

scheme, a 16-point DS is proposed to cover a bigger area without the three extra rounds. The modified TZS works under the assumption of slow and linear motion in the consecutive video frames. Secondly, the searching pace is changed from two pixels of TZS to one pixel in this paper in the searching direction, in order to better detect the minor changes. Thirdly, the searching process is performed in parallel using multiple threads in all the blocks within the searching window.

In the SEC scheme, multiple Working Threads (WTs) are generated during the ME process and are assigned to each IB. Each WT processes one row of the IB. The size of each IB is  $16 \times 16$ , so 16 WTs are assigned to each IB. To further speed up the searching process, an adaptive threshold is set to terminate the DS process. It changes every time when a new searching process triggers. Each RB is defined as a block inside searching window, and the sizes of RBs and IBs are the same. In the beginning, a random but a small value is set as a termination threshold. The matching process is based on one pixel movement. If the best match is found and the distance (i.e., the SAD value) is lower than the termination threshold, the searching process immediately terminates. If the best match is found and the distance is equal to or higher than the termination threshold, the searching process continues with the raster search to find an optimum match. The distance between an IB and the optimum match is used as a termination threshold for the searching process of next IB.

### E. Motion Vector Estimation

During the searching patterns phase, SAD is calculated for each pixel, and then compared with the termination threshold. There are three possible scenarios. Firstly, if the computed SAD is greater than the termination threshold in three unit steps (i.e., three pixel paces), the searching process leaves that row of pixels by assuming that there is no match in this row. Secondly, if the computed SAD is equal to the termination threshold, the searching process continues to find a better match up to three unit steps. Thirdly, if the computed SAD is less than the termination threshold, the searching process stops immediately. This process is repeated for all the possible locations in the targeted SW. This process may produce more than one better match, thus creating a set of best matches. Next, the  $(X-Y)$  coordinates of the pixels representing the best matches are obtained. The  $(X-Y)$  coordinates in the reference and the current video frames represent the tail and the head of the desired MVs, respectively. In the end, another search is performed to find an MV having a minimum length from the set of the estimated MVs. The MV with the minimum length is used to represent the motion of a block of pixels from the reference to the current video frames. This process is known as Forward Motion Estimation (FME) and the MVs estimated in this process are known as Forward Motion Vectors (FMVs). The same process is repeated from Subsections III-B to III-E by transforming the  $(N+1)^{th}$  and the  $(N-1)^{th}$  video frames as the reference and the current video frames, respectively. This process is known as Backward Motion Estimation (BME) and the MVs estimated in this process are known as Backward Motion Vectors (BMVs).

### F. Frame Interpolation

Let frame  $N$  be the frame to be concealed,  $F_i$  represents the  $i^{th}$  frame and  $F_i(x, y)$  represent the intensity at pixel location  $(x, y)$  of  $F_i$  ( $i = 1, 2, \dots$ ), and  $D$  represent the image domain. Then, the intensity values are initialized using the simple interpolation algorithm in Eq. 1 [30].

$$F_N(x, y) = 0.5F_{N-1}(x, y) + 0.5F_{N+1}(x, y), (x, y) \in D \quad (1)$$

Let  $V_{x,y}$  represent the MV of the block containing  $(x, y)$ , and  $(x_b, y_b)$  and  $(x_f, y_f)$  represent the tail and the head of the MV respectively. Then, it is obvious that

$$\begin{aligned} (x_b, y_b) &= (x, y) - 0.5V_{x,y}, \\ (x_f, y_f) &= (x, y) + 0.5V_{x,y}. \end{aligned} \quad (2)$$

Then, the intensity value at pixel location  $(x, y)$  of the missing video frame (i.e., frame  $N$ ) is updated using a modified interpolation algorithm, taking into account the block MV, as Eq. 3

$$F_N(x, y) = 0.5F_{N-1}(x_b, y_b) + 0.5F_{N+1}(x_f, y_f), \quad (3)$$

when both  $(x_b, y_b)$  and  $(x_f, y_f)$  are in  $D$ .

### G. Adaptive Filtering

Thin black-lines commonly appear in block-based approaches. The Arithmetic Mean Filter (AMF) [26] that is averaging pixel values in each pixel's neighborhood can be used to suppress this problem of thin black-lines, but it will also have blurring effects at image contours and edges. To propose a method that can smooth the mis-concealed pixel values while keeping the sharpness of edges and contours, an Adaptive Filter (AF) is applied and computed at each pixel location  $(x, y)$  by

$$AF(x, y) = \begin{cases} CP_{AMF}(x, y), & \text{if } \Delta_1 - \Delta_{(1)} > T_h \\ CP(x, y), & \text{otherwise} \end{cases} \quad (4)$$

where  $AF(x, y)$  is the output pixel value at  $(x, y)$  generated by an AF,  $CP(x, y)$  is the concealed pixel at  $(x, y)$  computed by Eq. 3,  $CP_{AMF}(x, y)$  is the pixel value at  $(x, y)$  concealed using AMF,  $\Delta_1$  is the cumulatively weighted distance (CWD) allocated to the central pixel of a filtering window, and  $\Delta_{(1)}$  is the smallest CWD within the window, and  $T_h$  is a threshold value. Detailed description of  $\Delta_1$  and  $\Delta_{(1)}$  and the selection of  $T_h$  value can be referred to [31]. Algorithm 1 summarizes the entire SEC scheme detailed above in this section.

## IV. EXPERIMENTAL SETUP AND RESULTS

### A. Experimental Setting

Test video sequences are chosen from an SHVC standard documentation [32]. Six test video sequences are used for simulation purpose in our experiments. Details of the test video sequences, such as resolution, total number of frames and frame rate, are listed in Table I.

**Algorithm 1** Scalable Error Concealment

```

1: procedure SCALABLE-ERROR-CONCEALMENT
2:   IV: Input Video, IB: Input Block, SW: Search Window, RB: Reference Block
3:   for Missing  $F_N \in IV$  do
4:     if  $F_N \in BL$  then
5:       goto Concealment
6:     else
7:       Find Master Frame
8:       goto Concealment
9:     end if
10:    goto Filtering
11:  end for
12: end procedure
13: Concealment:
14: while  $F_{N-1}, F_N, F_{N+1} \in IV$  do
15:   Read  $F_{N+1}$ 
16:   Read  $F_{N-1}$ 
17:   goto FME
18:   goto BME
19:   Conceal  $F_N$  using Eq. 3
20: end while
21: FME:
22: while  $IB \subset F_{N-1}$  do
23:   Construct SW in  $F_{N+1}$ 
24:   for  $RB \subset SW$  do
25:     Generate Working Threads (WTs)
26:     Perform Mapping between IB and RB Using WTs
27:     Compute FMV for IB
28:   end for
29: end while
30: BME:
31: while  $IB \subset F_{N+1}$  do
32:   Construct SW in  $F_{N-1}$ 
33:   for  $RB \subset SW$  do
34:     Generate WTs
35:     Perform Mapping between IB and RB Using WTs
36:     Compute BMV for IB
37:   end for
38: end while
39: Filtering:
40: while True do
41:   Generate WTs
42:   Apply Adaptive Filter on Concealed  $F_N$  Using WTs
43: end while

```

Encoding details, such as quantization parameters (QPs) used during the encoding process and the bit-rates obtained at the end of the encoding process, are tabulated in Table II. The QPs regulate how much detail within a frame is saved. The bit-rate means the number of bits that make up a section of a video per unit of time, i.e., a bit-rate of  $1kbs$  means that every second of video is represented by 1 thousand bits. In SHVC encoding, QP values always come in pairs. In a QP pair, first value is used to encode BL and second value is used to encode EL. QP pairs are chosen according to the SHVC standard

Sequence	Resolution	Total Number of Frames	Frame Rate
PeopleOnstreet	$2560 \times 1600$	150	30
BasketballDrive	$1920 \times 1080$	500	50
BQTerrace	$1920 \times 1080$	600	60
Cactus	$1920 \times 1080$	500	50
Kimono	$1920 \times 1080$	240	24
ParkScene	$1920 \times 1080$	240	24

TABLE I. Test Video Sequences

documentation [32]. The test videos are encoded using the standard settings of SHVC in an Unsliced mode. Group of Pictures (GoP) is set to 1 because of the Intra coding mode [32].

Sequences	QP Pairs				
	(20,22)	(24,26)	(28,30)	(32,34)	(36,38)
PeopleOnStreet	152.799	711.312	505.382	319.452	205.801
BasketballDrive	102.077	412.136	214.05	129.638	805.679
BQTerrace	191.936	123.574	675.929	406.758	249.337
Cactus	156.169	744.634	412.003	250.373	151.744
Kimono	234.06	135.395	861.646	561.255	357.66
ParkScene	646.129	394.476	235.31	137.842	775.943

TABLE II. QP Pairs and Video Sequences' Bitrates

Hardware	CPU: Intel Core <sup>TM</sup> i5-4590 CPU @ 3.30 GHz RAM: 16 GB
Software	SHVC Codec SHM-10.0-Dev Matlab R2015b
Video Format	4:2:0
QP	20 to 38
PLR	1%, 3%, 5%
Methods	FC, BMA, Proposed technique

TABLE III. Simulation Environment

Details of the experimental platform are summarized in Table III. The experiments and performance evaluations of the SEC scheme are conducted using SHM-Dev 10.0 [33] and Matlab R2015a. To simulate packet-losses, H.265 RTP loss model proposed in [34] is adopted. In our experiments, we selected PLRs of 1%, 3% and 5%. For comparison purposes, we also implemented the FC [11] and the BMA [12] techniques under the same experimental settings.

The test videos are encoded in the cloud environment, provided by Matlab. Current version of Matlab offers Amazon EC2 cloud facility and programmers can easily use built-in functions to perform cloud-based computing [35]. In our experiments, the Amazon cluster consists of 20 virtual machines of type medium, having 2 virtual CPUs and 16GB RAM. SHVC encoder is executed on one virtual machine of type medium. The maximum size of a heap is set to 8GB.





Sequence	Algorithm	1% PLR			3% PLR			5% PLR		
		QP (20,22)	QP (28,30)	QP (36,38)	QP (20,22)	QP (28,30)	QP (36,38)	QP (20,22)	QP (28,30)	QP (36,38)
PeopleOnStreet	BMA-FS	88.63	90.39	98.93	160.77	152.97	120.66	140.62	130.32	120.72
	BMA-TZS	75.83	77.31	76.86	135.83	129.49	102.76	115.72	105.85	100.82
	Proposed-TZS	0.2103	0.2111	0.2103	0.2333	0.2119	0.2003	0.2321	0.2121	0.2133
BasketballDrive	BMA-FS	83.55	85.87	80.22	148.29	135.62	105.53	115.83	110.39	90.29
	BMA-TZS	61.77	66.66	62.37	129.34	107.75	77.26	90.91	88.55	70.73
	Proposed-TZS	0.2072	0.2122	0.2999	0.2243	0.2191	0.2133	0.2121	0.2201	0.2103
BQTerrace	BMA-FS	77.46	82.88	78.21	146.26	132.69	93.67	111.86	107.79	85.94
	BMA-TZS	60.63	64.87	61.22	114.48	103.86	73.32	87.55	84.37	67.27
	Proposed-TZS	0.2108	0.2111	0.209	0.2113	0.2096	0.2115	0.211	0.2106	0.2103
Cactus	BMA-FS	99.14	99.14	89.65	125.6	115.07	61.72	112.37	107.1	75.68
	BMA-TZS	77.59	77.59	70.17	98.31	90.01	48.31	87.95	83.83	59.24
	Proposed-TZS	0.2123	0.2123	0.2068	0.2084	0.2091	0.2109	0.2103	0.2107	0.2089
Kimono	BMA-FS	82.52	86.15	82.52	146.78	158.18	101.6	114.65	122.17	92.06
	BMA-TZS	64.59	67.43	64.59	114.89	123.81	79.53	89.74	95.63	72.06
	Proposed-TZS	0.2121	0.2173	0.2121	0.2143	0.2114	0.2097	0.2132	0.2143	0.2109
ParkScene	BMA-FS	84.88	85.01	83.39	147.33	155.23	110.43	109.23	125.88	100.52
	BMA-TZS	62.73	63.11	64.84	139.66	119.7	80.39	91.88	93.06	70.56
	Proposed-TZS	0.2109	0.2045	0.2111	0.2211	0.2009	0.2056	0.2101	0.2078	0.2177

TABLE VI. Average Computational Time (Minutes) of Test Video Sequences

requires approximately 99.75% to 99.76% less time under a 1% PLR, 99.76% to 99.85% less time under a 3% PLR and 99.75% to 99.82% less time under a 5% PLR, compared to the original TZS and the BMA approaches. Hence, the SEC scheme performs better in terms of both the average PSNRs and the computational times. The computational time remains a major target in the SEC scheme, as the main motive is to support the real-time processing of SHVC video streams. We cannot expect high computational power and unlimited memory at an end-user's mobile device. Delays are always critical in real-time applications, so light-weight processing and requirement for less hardware resources are recommended.

For a visual comparison, we randomly select three different video sequences from the test videos dataset. The visual results of the referenced and the SEC schemes are depicted in Fig. 7 under different QPs and PLRs. In Fig. 7, first column (from left to right) represents the original video frame lost during a transmission and second, third and fourth columns represent the outputs of the FC, the BMA and the SEC schemes, respectively. The sample images are taken from the test video sequences, having different types of motions and are encoded with different QPs. The first sample is taken from the video sequence BQTerrace with QP pair (20, 22), the second one is taken from the video sequence Cactus with QP pair (28, 30) and the third one is taken from the video sequence Kimono with QP pair (36, 38). These comparisons show clearly that there is not much visible difference between the original video frame and the outputs of the FC and the SEC schemes. However, blurred lines are quite visible in the outputs of the BMA technique. In the consecutive video frames, the motion of objects is usually slow and the FC does not show noticeable differences. However, in the experiments with higher PLRs, visible fluctuations and video freezing effects are experienced when the FC technique is applied. Because of their large sizes, the videos showing the effects are not possible to be uploaded

in this paper to show the effects of errors, so we have included only the specific video frames instead.

The average PSNRs and the average computational times, under different QP pairs, of the three video sequences are plotted in Fig. 8 and Fig. 9, respectively. Begin from the left, the three columns in the sequential order, represent 1%, 3% and 5% PLRs, respectively. It is clear, in both figures, that the SEC scheme outperforms both FC and the BMA techniques in terms of average PSNRs and computational times. The significantly smaller computational times of the proposed SEC proves its suitability for real-time processing of SHVC video streams.

Our results in Fig. 7 and Fig. 8 demonstrate that, by applying the proposed SEC scheme, we can greatly enhance the QoE of the users. The subjective gains are shown in Fig. 7 and the objective gains are shown in Fig. 8 and Fig. 9 in terms of PSNRs and computational costs using the proposed SEC scheme. They demonstrate that the SEC scheme not only achieves much better quality but also has much lower computational costs.

### C. Computational Complexity

The execution time of the proposed SEC scheme is much less than the standard classic BMA technique. It is demonstrated, in Table V and Fig 9, that our proposed SEC scheme achieves approximately several hundred times faster speed than the standard BMA technique. Note that the values quoted demonstrate only the 'relative' reduction in terms of computation costs. The experiments are performed on a computer using Matlab, and not on a DSP/DIP processor using C or other hardware languages. In real applications with optimization, the computational costs can be in micro or nano seconds and the information of implementing the video algorithms on a hardware kit with timing details can be found in [36], [37]. Therefore, from the mobile devices point of view, the

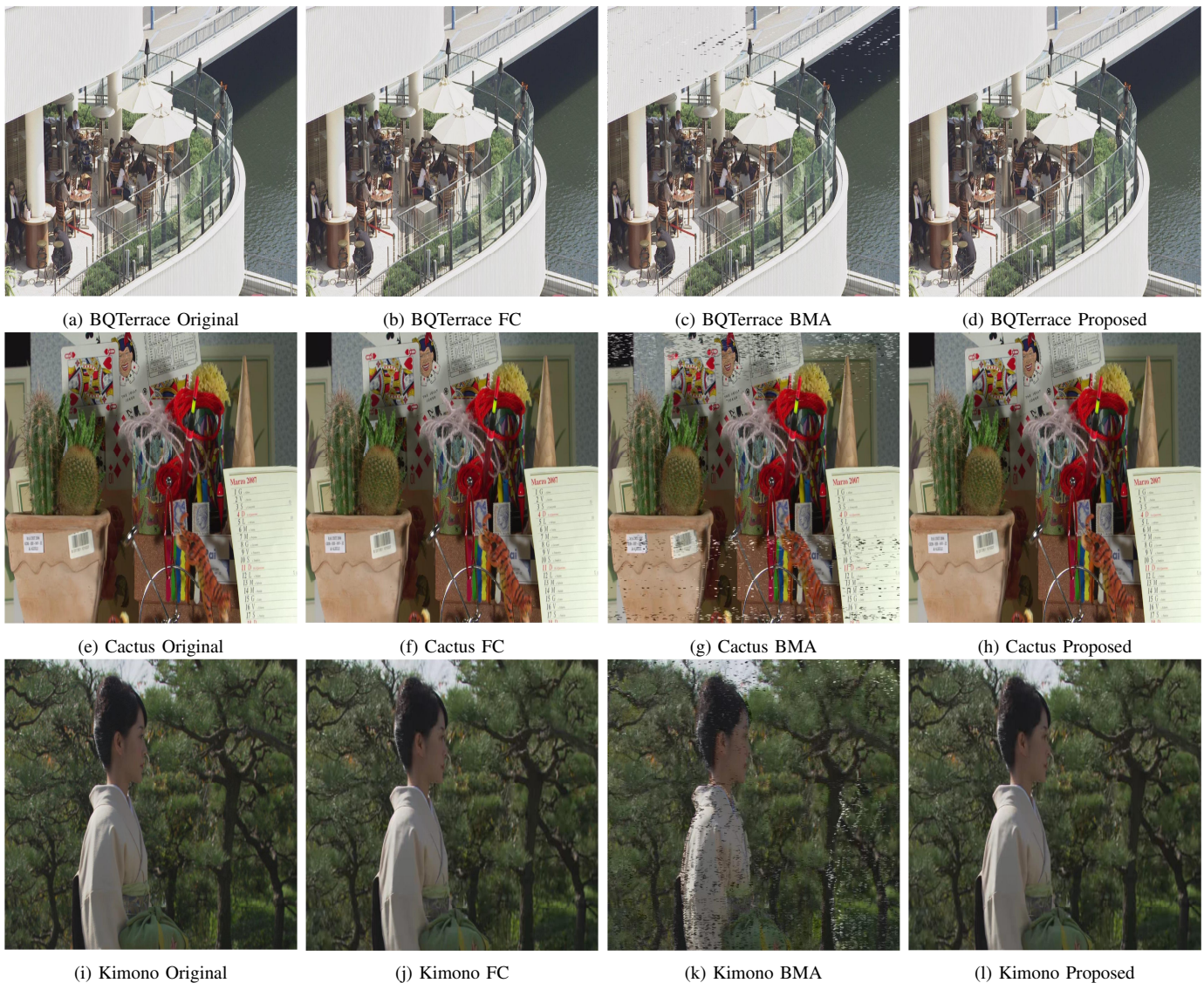


Fig. 7. Visual Comparison

overhead caused by the computational costs is imperceptible as the proposed scheme just involves implementing a motion estimation algorithm and an adaptive filter when it receives an alert for a missing video frame. Modern mobile devices are easily able to encode and decode videos at a speed up to 60fps, so adoption of the proposed scheme is not an issue for them at all. It will be our future work to provide those details of implementing this proposed scheme (whose concepts are proved and verified by the experiments in this paper) on an embedded kit and code in a low level programming language such as C.

Let  $m \times m$  be the block size in the SEC scheme. Then, the time complexity of SAD is  $O(m^2)$ , as stated in [38], [39]. Note that, in a BMA based on FS or TZS. TZS is a combination of a diamond search and a raster search. The raster search goes through every pixel. Furthermore, each pixel in an RB is searched and the RB and the IB have the same size. Therefore, in the worst case, the computational complexity of raster search is  $m^2$ , and the computation complexity of BMA

is  $O(m^2 \times m^2)$ , i.e.,  $O(m^4)$ .

On the other hand, in the SEC scheme, the search does not go till the end of an RB. The search ends in the first few steps once the nearest match is found, so the computational complexity of the searching process in the SEC scheme is  $O(p)$ , where  $p$  represents the total number of unit steps and  $p < m^2$ . Furthermore, the matching process based on SAD in the SEC scheme is performed in parallel in rows using multiple threads, so the time complexity of performing SAD in the SEC is  $O(m)$ . Therefore, the computation complexity of SEC is  $O(p \times m)$ .

## V. CONCLUSION

On-demand and self-service features of cloud computing provide a facility to a VSP to reduce costs of hardware installation and maintenance. In this paper, we have studied problems of big video data processing, storage and quality of transmitted videos. We have proposed a scheme, in which HD and UHD videos are encoded using the SHVC standard and

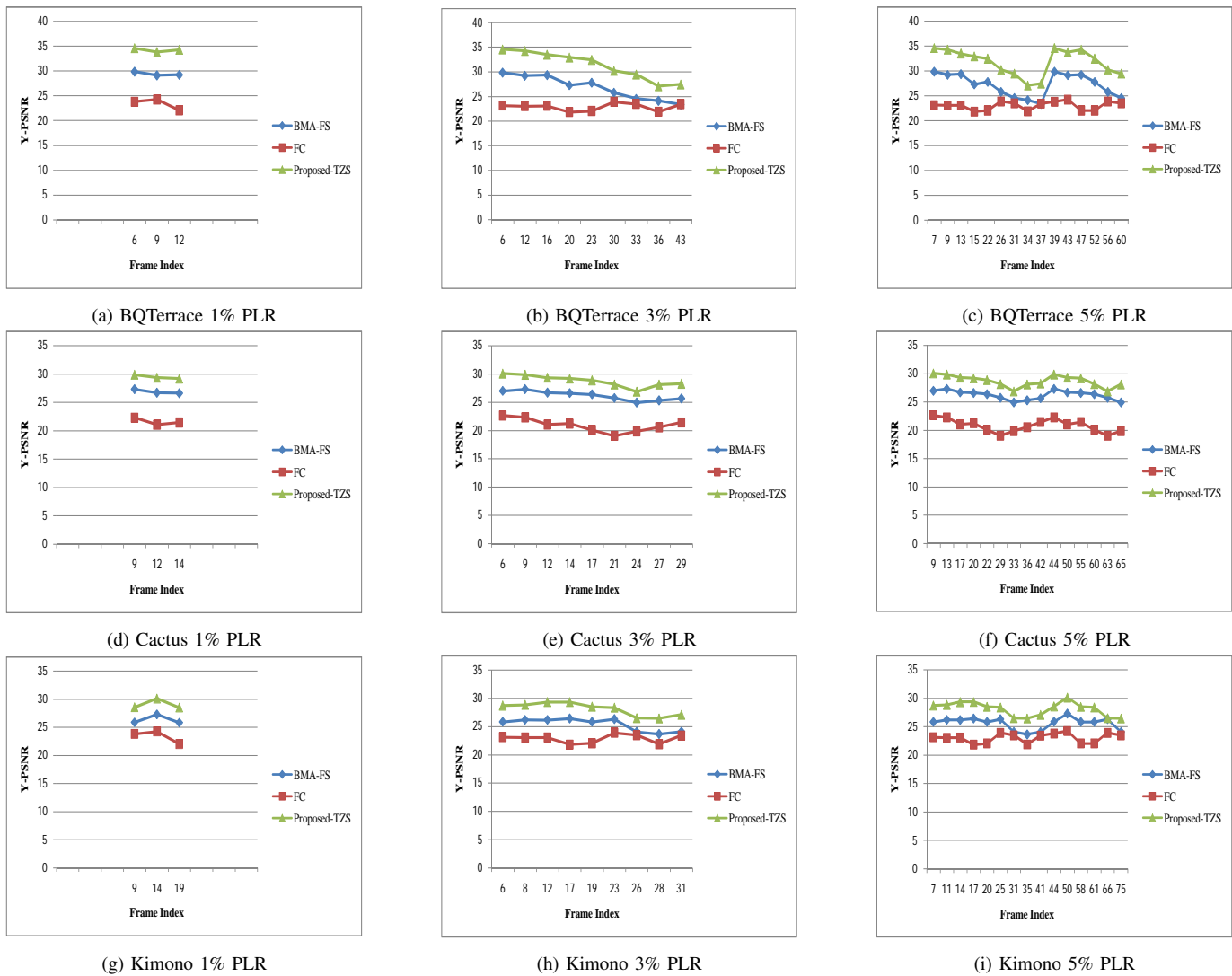


Fig. 8. PSNR Comparison

stored at cloud servers, formally known as media clouds. The video streaming can be either live or stored-file-based from a cloud platform. To face the challenge of maintained quality of transmitted videos, we have also proposed a fast and an efficient SEC scheme for scalable video streams. The SEC scheme uses a multi-threading concept to perform multiple computations in parallel to save the computational time. We have evaluated the SEC scheme and demonstrated it better performances in terms of both average computational times and PSNRs. In the experiments, we have used HD and UHD videos with different types of motions. The great reduction in the average computational time by the SEC scheme proves that it can be applied to low power end-user mobile devices. Regarding future works, we shall use the simulation results produced in this paper to perform further enhancements in the SEC scheme. Moreover, we shall use 4k and 8k videos to evaluate the performance of the SEC scheme.

## REFERENCES

- [1] Y. Liu, F. Li, L. Guo, B. Shen, S. Chen, and Y. Lan, "Measurement and analysis of an internet streaming service to mobile devices," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2240–2250, November 2013.
- [2] A. Seetharam, P. Dutta, V. Arya, J. Kurose, M. Chetlur, and S. Kalyanaraman, "On managing quality of experience of multiple video streams in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 03, pp. 619–631, March 2015.
- [3] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, December 2012.
- [4] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramanian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 01, pp. 20–34, January 2016.
- [5] X. Liu, D. Zhao, L. Xu, W. Zhang, J. Yin, and X. Chen, "A distributed video management cloud platform using hadoop," *IEEE Access*, vol. 13, no. 09, pp. 2637–2643, September 2015.
- [6] Y. Mo, J. Chen, X. Xie, C. Luo, and L. Yang, "Cloud-based mobile multimedia recommendation system with user behavior information," *IEEE Systems Journal*, vol. 08, no. 01, pp. 184–193, March 2014.
- [7] M. Usman, M. A. Jan, and X. He, "Cryptography-based secure data storage and sharing using hevc and public clouds," *Information Sciences*, 2016.
- [8] E. Dede, B. Sendir, P. Kuzlu, J. Weachock, M. Govindaraju, and L. Ramakrishnan, "Processing cassandra datasets with hadoop-streaming based approaches," *IEEE Transactions on Services Computing*, vol. PP, no. 99, June 2015.

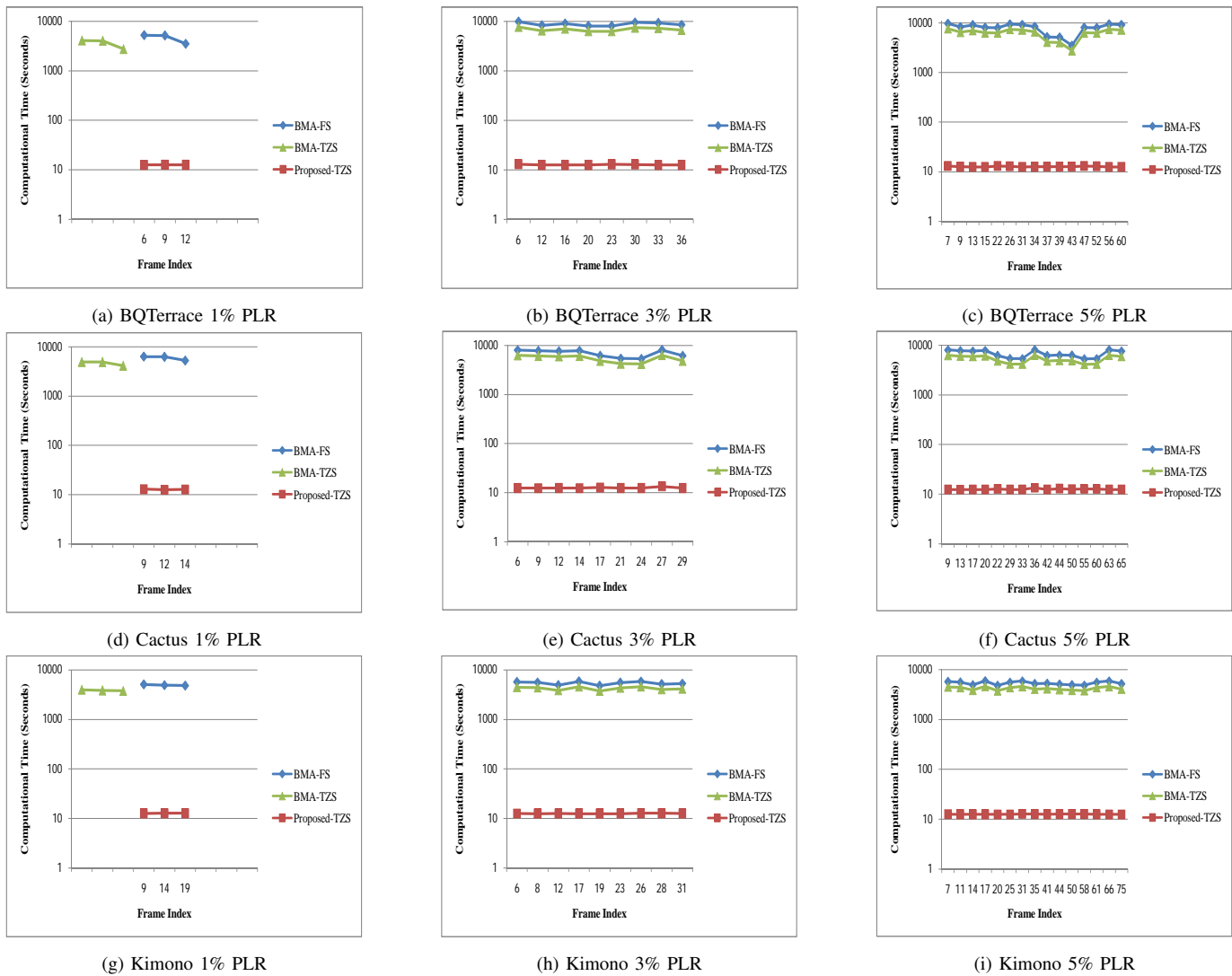


Fig. 9. Computational Time Comparison

- [9] S. Deng, L. Huang, J. Taheri, and A. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317–3329, December 2015.
- [10] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 01, pp. 369–392, May 2013.
- [11] E. Yaacoub, F. Filali, and A. Dayya, "Qoe enhancement of svc video streaming over vehicular networks using cooperative lte/802.11p communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 01, pp. 37–49, February 2015.
- [12] J. Zhou, B. Yan, and H. Gharavi, "Efficient motion vector interpolation for error concealment of h.264/avc," *IEEE Transactions on Broadcasting*, vol. 57, no. 01, pp. 75–80, March 2011.
- [13] Wikipedia, "Wikipedia block matching algorithms," 2016. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350>
- [14] Y. Wen, X. Zhu, C. Chen, and J. Rodrigues, "Mobile cloud media: Reflections and outlook," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 885–902, 2014.
- [15] G. Gao, W. Zhang, Y. Wen, Z. Wang, and W. Zhu, "Towards cost-efficient video transcoding in media cloud: Insights learned from user viewing patterns," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1286–1296, 2015.
- [16] J. Hwang, J. Lee, N. Choi, and C. Yoo, "Havs: Hybrid adaptive video streaming for mobile devices," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 02, pp. 210–216, May 2014.
- [17] Y. Ye, Y. He, and X. Xiu, "Manipulating ultra-high definition video traffic," *IEEE Multimedia*, vol. 22, no. 03, pp. 73–81, January 2015.
- [18] S. Tang and P. Alface, "Impact of random and burst packet losses on h.264 scalable video coding," *IEEE Transactions on Multimedia*, vol. 16, no. 08, pp. 2256–2269, December 2014.
- [19] L. Yu, H. Li, and w. Li, "Wireless scalable video coding using a hybrid digital-analog scheme," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 02, pp. 331–345, February 2014.
- [20] M. Usman, X. He, M. Xu, and K. M. Lam, "Survey of error concealment techniques: research directions and open issues," in *Picture Coding Symposium (PCS), 2015*. IEEE, 2015, pp. 233–238.
- [21] P. Zimmermann and R. Brent, "Modern computer arithmetic," *Cambridge University Press*, December 2010.
- [22] J. Koloda, A. Peinado, and V. Sanchez, "Kernel-based mmse multimedia signal reconstruction and its application to spatial error concealment," *IEEE Transactions on Multimedia*, vol. 16, no. 06, pp. 1729–1738, October 2014.
- [23] S. Yang, C. Chang, and C. Chan, "An objetc-based error concealment technique for h.264 coded video," *Multimedia Tools and Applications*, vol. 74, no. 23, pp. 10785–10800, December 2015.
- [24] W. Lie, C. Lee, C. Yeh, and Z. Gao, "Motion vector recovery for video error concealment by using iterative dynamic-programming optimization," *IEEE Transactions on Multimedia*, vol. 16, no. 01, pp. 216–227, January 2014.
- [25] D. Zhang, H. Li, and C. Chen, "Robust transmission of scalable video coding bitstream over heterogeneous networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 02, pp. 300–313,

- February 2015.
- [26] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.
- [27] LG, "Lg tv," 2016. [Online]. Available: <http://www.lg.com/us/tvs>
- [28] W. Choi, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," *Proceedings of IEEE International Conference on Image Processing*, vol. 03, pp. 371–374, September 2003.
- [29] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the hevc standard," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012.
- [30] "Motion interpolation," 2010. [Online]. Available: <http://monochrome.sutic.nu/2010/10/12/motion-interpolation.html>
- [31] B. Smolka, K. Malik, and D. Malik, "Adaptive rank weighted switching filter for impulsive noise removal in color images," *Journal of Real-Time Image Processing*, vol. 10, no. 2, pp. 289–311, 2015.
- [32] G. Sullivan and V. Seregin, "Shvc draft," 2015. [Online]. Available: <https://hevc.hhi.fraunhofer.de/shvc>
- [33] "Shm-10.0-dev," 2015. [Online]. Available: [https://hevc.hhi.fraunhofer.de/svn/svn\\_SHVCSoftware/branches/](https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/branches/)
- [34] W. Stephan, "Nal unit loss software," *JCT-VC Document, JCTVH0072*, February 2012.
- [35] M. Corporation, "Matlab parallel computing," 2015. [Online]. Available: <http://www.mathworks.com/products/parallel-computing/parallel-computing-on-the-cloud/?requestedDomain=www.mathworks.com&refresh=true>
- [36] M. U. Shahid, A. Ahmed, M. Martina, G. Masera, and E. Magli, "Parallel h. 264/avc fast rate-distortion optimized motion estimation by using a graphics processing unit and dedicated hardware," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 701–715, 2015.
- [37] A. Aysu, G. Sayilar, and I. Hamzaoglu, "A low energy adaptive hardware for h. 264 multiple reference frame motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, 2011.
- [38] M. Al-Mualla, C. N. Canagarajah, and D. R. Bull, *Video coding for mobile communications: efficiency, complexity and resilience*. Academic Press, 2002.
- [39] S. Metkar and S. Talbar, *Motion estimation techniques for digital video coding*. Springer, 2013.



**Muhammad Usman** is currently pursuing the Ph.D. in Computer Systems from the School of Computing and Communications, University of Technology Sydney, Australia. His research interests include audio, image and video processing, compression, communication, security in wireless networks, fast video encoding algorithms and error concealment techniques.



**Xiangjian He** received a Ph.D. in computing sciences from the University of Technology, Sydney, Australia, in 1999. Since 1999, he has been with the University of Technology, Sydney. He is currently a full professor and the director of Computer Vision and Pattern Recognition Laboratory there.



**Kin-Man Lam** received the Ph.D. degree from the Department of Electrical Engineering, University of Sydney, Sydney, Australia, in August 1996. Since 1990, he has been with The Hong Kong Polytechnic University, Hong Kong. He is currently a full professor. His current research interests include human face recognition, image and video processing, and computer vision.



**Min Xu** received the Ph.D. degree from University of Newcastle, Australia in 2010. She is currently a Senior Lecturer at University of Technology, Sydney, Australia. Her research interests include multimedia content analysis, multimedia affective computing, social multimedia, pattern recognition and computer vision.



**Syed Mohsin Matloob Bokhari** received the Ph.D. degree in Electrical and Electronics Engineering from the University of Bristol, Bristol, U.K. in 2012. Since 2012, he has been with University of Engineering and Technology Peshawar, Pakistan. He is currently an assistant professor. His research interests include image and video processing, compression, and communication.



**Jinjun Chen** is a Professor from Faculty of Science, Engineering and IT, Swinburne University of Technology, Australia. He holds a PhD in Computer Science and Software Engineering (2007) from Swinburne University of Technology, Melbourne, Australia. His research interests include cloud computing, big data and data intensive systems. He has published more than 130 papers in high quality journals and conferences.



**Mian Ahmad Jan** is an Assistant Professor at the Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan. He holds a Ph.D. in Computer Systems from University of Technology Sydney, Australia. His research interests include cluster-based hierarchical routing protocols, congestion detection and mitigation and intrusion and malicious attack detections in wireless sensor networks, Internet and web of things.