Journal of Real-Time Image Processing manuscript No.

Efficient Algorithm for Transferring a Real-Time HEVC Stream with Haptic Data Through the Internet

George Kokkonis, Kostas E. Psannis, Manos Roumeliotis, Yutaka Ishibashi

Abstract It is widely accepted that the growth of Internet and the improvement of Internet's network conditions helped real-time applications to flourish. The demand for Ultra High Definition video is constantly increasing. Apart from video and sound a new kind of real-time data is making its appearance, haptic data. The efficient synchronization of video, audio and haptic data is a rather challenging effort. The new High Efficiency Video Coding is quite promising for real-time ultra-high definition video transferring through the Internet. This paper presents related work on High Efficiency Video Coding. It points out the challenges and the synchronization techniques that have being proposed for synchronizing video and haptic data. Comparative tests between H.264 and HEVC are undertaken. Measurements for the network conditions of the Internet are carried out. The equations for the transferring delay of all the inter prediction configurations of the HEVC are defined. Finally, it proposes a new efficient algorithm for transferring a real-time High Efficient Video Coding stream with haptic data through the Internet.

Keywords Haptics, HEVC, High Efficiency Video Coding, Algorithm, Synchronization techniques, Inter prediction, Haptic data.

G. Kokkonis, K. E. Psannis, M. Roumeliotis
Department of Applied Informatics,
University of Macedonia,
156 EgnatiaStreet,Thessaloniki 54006 , Greece.
E-mail: {gkokkonis, kpsannis, manos} @uom.gr

Y. Ishibashi
Department of Scientific and Engineering Simulation,
Nagoya Institute of Technology,
Nagoya 466-8555, Japan.
E-mail: ishibasi@nitech.ac.jp

1 Introduction

The increasing demand for real-time applications with high and ultra-high definition video urged the ITU-T and the ISO/IEC to join their forces to develop the next-generation video coding standard. The Joint Collaborative Team on Video Coding (JCT-VC) has been created. The new coding standard that has been produced is known as High Efficiency Video Coding (HEVC). The proposed HEVC standard fulfilled its target to achieve more than 50% improvement in video compression over the existing H.264 Advanced Video Coding standard, keeping comparable image quality, at the expense of increased computational complexity. HEVC targets a wide variety of high definition video applications such as the 4k television with screen resolution of 4096 × 2160 and the Ultra High Definition Television (UHDTV) with screen resolutions of up to 7680×4320.

The 50% of improvement on video compression that HEVC achieves is denatured in 50% lower bit rate on video streaming. This reduction together with the improvement of Internet network conditions, made the streaming of high and ultra-high definition video over the Internet feasible.

Apart from video, another kind of real-time data is now trying to travel through the Internet, this is haptic data. The word haptic derives from the Greek "haptikos" meaning "pertaining to the sense of touch".

Since haptics refers to the sense of touch, video refers to the sense of vision and audio refers to the sense of hearing, it is becoming clear that all these streams that try to travel through the Internet should be synchronized, in order to achieve maximum Quality of Experience (QoE) for the users.

The rest of the paper is organized as follows. Section 2 presents the HEVC. Section 3 outlines the characteristics of haptic data. Section 4 describes the synchronization algorithms for inter media synchronization. Section 5 analyzes the proposed algorithm for transferring HEVC video stream enhanced with haptic data through the Internet. Finally section 6 identifies conclusions and future work.

2 High Efficiency Video Coding (HEVC)

HEVC's main target was to increase data compression by 50% over its predecessor H.264, while keeping the same image quality, at the expense of computational cost.

The image quality can be measured with two kinds of perspectives, objective and subjective. The objective video quality assessment is defined as the signal-to-noise ratio (SNR) and peak-to-noise ratio (PSNR) between the original video signal and the video signal after the encoding and the decoding process. The subjective method is based on the Mean Opinion Score (MOS). Videos are shown to a group of viewers and their opinions are recorded and averaged to evaluate the quality of each video.

Many studies [1] - [5] have shown that the increase by 50% of the data compressions has been achieved. Table 1 shows the bit rate reduction of HEVC over the AVC for similar video quality, for three videos with different content [2]. Similarly, Jens-Rainer Ohm et al.in [1] showed that the average bit rate saving for entertainment applications is 35,4% measured with the objective PSNR method and 49,3% measured with the subjective MOS method.

Table 1 Bit Rate Reduction of HEVC OverAVC For Similar Video Quality[2]

	Bit Rate Reduction	
Content	***Objective (PSNR)***	***Subjective (MOS)***
People On the Street	27.5%	50.8%
Traffic	27.5%	74.0%
Sintel2	68.0%	74.7%
Average	**44.4%**	**66.5%**

Depending on the application scenario, HEVC offers many configurations modes for efficiency, computational complexity, processing delay, parallelization and error resilience techniques [6].

The two main encoding complexity configurations are the "High Efficiency" and the "Low complexity" modes. The former offers a high efficiency encoding at the expense of computational cost while "Low complexity" offers reasonably high efficiency while trying to keep the encoder complexity as low as possible [7].

As far as the temporal prediction structure is concerned, there are three prediction modes. The first mode is the "intra-only" configuration, where each picture is encoded independently and no temporal prediction is used. The second mode is the "Low-Delay configuration", where only the first picture of the video sequence will be used as an Instantaneous Decoder Refresh (IDR) coded picture, all the other pictures are encoded as Generalized P and B Pictures (GPB), in mandatory Low-Delay test condition, or as P Pictures, which is called non-normative Low-Delay condition. The third mode is the "Random-Access" mode, where the first picture in a Group of Pictures (GOP), which lasts for approximately 1 sec, is encoded as IDR picture and all the other pictures inside the GOP are encoded as B or GPB pictures.

Apart from the temporal prediction, HEVC uses inter and intra spatial prediction based on the coding unit (CU) structure, the prediction unit (PU) and the Transform Unit (TU). Each picture is divided into coding tree units (CTU) of up to 64 X 64 luma samples. CTUs are split into CUs with the help of a generic quad-tree segmentation structure. CUs can be further split into PUs and TUs [8].

Another interesting feature of the HEVC encoder is the slice and tile partition operation. With the help of the slice partitioning, the HEVC manages to fragmentize the encoded pictures near the maximum transmission unit (MTU) size commonly found in IP networks. While with the help of the tile partitioning, the HEVC exploits the parallel processing of independent tiles of a picture in multiple cores and processors of a computer [9]. Parallel video coding showed that real-time performance for 1920×1080p/50Hz (53.1 fps) and 2560×1600 (29.5 fps) video resolutions is possible [10].

3 Haptics

Haptics refer to the science of manual sensing and manipulation of surrounding objects and environments through the sense of touch [11]. With the optimization of telerobotics and the improvement of Internet status, there is a constant effort to transfer the sense of touch through the Internet. Through that effort a new kind of data made its appearance in the last decade. This is tele-haptic data.

The main obstacle that impedes tele-haptics from flourishing is the delay and the jitter that is being encountered in the Internet. Table 2 depicts the QoS requirements that have to be fulfilled in order for the users' QoE to be satisfactory.

Table 2 QoS Requirements for Multimedia Streams [12]-[16]

	APPLICATIONS			
QOS	***HAPTICS***	***VIDEO***	***AUDIO***	***GRAPHICS***
JITTER (ms)	≤ 2	≤ 30	≤ 30	≤ 30
DELAY (ms)	≤ 50	≤ 400	≤ 150	≤ 100-300
PACKET LOSS (%)	≤ 10	≤ 1	≤ 1	≤ 10
UPDATE RATE (Hz)	≥ 1000	≥ 30	≥ 50	≥ 30
PACKET SIZE (bytes)	64-128	≤ MTU	160-320	192-5000
THROUGHPUT (Kbps)	512-1024	2500-40000	64-128	45-1200

It can be understood from Table 2 that the QoS for haptic data is by far stricter than other applications, as far as jitter, delay and update rate is concerned.

Several techniques have been proposed for the limitation of the negative effects of the delay and jitter. Some of them are the wave variants [17], the haptic packet prioritization [18], the adaptive buffering [19], and the "deadband control" reduction of the sending rate [20]. The main system architecture for tele-haptic applications is the four channel

(4ch) architecture [21].

The proposed high level architectural design for the synchronization of the media streams is depicted in Figure 1.

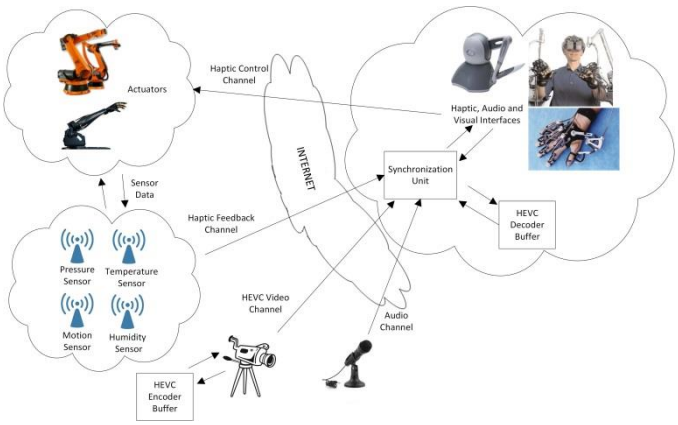

Fig. 1 High level architectural design of the Proposed Haptic System.

It contains four separate communication channels. Those are:

Haptic Control Channel. It carries command queries from the user to the remote Haptic equipment. As Haptic data are very sensitive to network delay and jitter, Haptic data should be transferred by a specific for this purpose transport protocol. Enforcement of strict QoS rules for haptic data of Table 2 should be enforced.

Haptic Feedback Channel. It carries sensor data and response queries from the remote haptic equipment back to the user. As it transfers Haptic data, strict QoS rules should also be enforced.

HEVC Video Channel. It carries an HEVC encoded video stream from the remote environment back to the user. Depending on the resolution of the video, this stream usually occupies the highest percentage of the bandwidth of the communicational channel.

Audio Channel. It carries audio data from the remote environment back to the user. It is the channel with the smallest QoS requirements.

All the above streams transfer their data through the Internet. Audio, Video and Haptic feedback Media Units (MU) arrive at the **Synchronization Unit** with disrupted time intervals compared to the generation ones at the source. The main operation of the Synchronization Unit is to preserve the time relation of the original signal as steady as possible and synchronize the three media streams with each other.

4 Synchronization of Media Streams

One of the negative effects that jitter and delay cause is the desynchronization of the haptic data with the streams of audio and video. This effect is particularly evident in real-time applications that transfer real-time data through the Internet. This is caused mainly by the fact that the end to end delay of each MU is not stable. The MU of each stream may arrive at its destination with a different order than when it left its source [22].

This desynchronization is also accentuated by the following factors:

i. Each of the voice, video and haptic data stream has different size of MU. The MU of haptic data is usually 40 to 64 bytes. The MU for high efficiency video is usually as the Maximum Transmission Unit (MTU) that the IP protocol can transfer, which is 1500 bytes. The size of the MU for a voice stream is usually 160 to 320 bytes.

ii. MUs are usually transmitted by the User Datagram Protocol (UDP) and the Real-Time Protocol (RTP) [23]. Each of the streams has different MU rate. The voice data rate is usually 50 MU/sec and the video MU rate is 30 MU/s. Haptic data have a rather big MU rate of 1000 MU/s.

iii. The average bit rate of each stream is different. The video stream, for a 1920×1080 at 24 fps video resolution, has transmission rate, for an HEVC Intra – Low Complexity encoding 4184 Kbps and for an HEVC Low-Delay – Low Complexity encoding 565 Kbps [4]. The average bit rate of the voice stream in case of a linear Pulse Code Modulation (PCM) sound is usually 64 or 128 Kbps. On the other hand, haptic data have, for MU of 40 bytes, an average bit rate of 320 Kbps [22].

In order all these deviations to be compensated, some rather interesting synchronization algorithms have been proposed [24]. They are divided into two main categories, the Intra-stream and the Inter – stream synchronization algorithms. The former are trying to preserve the time relation inside a single stream, while the latter are trying to keep the temporal relation among multiple streams.

Moreover, all the synchronization techniques can be divided into preventive and reactive techniques. The former are trying to prevent asynchrony, while the later are trying to recover asynchrony (skipping, discarding, shortening and extension of output duration, and virtual time-contraction and time expansion) [25].

The evaluation of all the above techniques could be made either subjectively, with the help of the MOS of volunteers, or objectively by measuring the average MU rate, total pause time, average MU delay, and mean square error of inter-stream synchronization [26].

5 The Proposed Transferring Algorithm

5.1 Synchronization

Quite a lot of research [9, 10] has been done for real-time encoding with an HEVC encoder. All the researchers have come to the conclusion that the real-time encoding with HEVC is feasible, as long as parallel processing with multiple cores is utilized. Apart from the real-time encoding and decoding of the video stream, the audio and the haptic stream

should be transferred through the Internet as well.

As mentioned in the previous section, the video, audio and haptic streams have different data rates from each other. This means that the streams are *loosely coupled*. As a consequence, all the streams have to be synchronized with each other, in order for the maximum QoE to be achieved. Both Intra and Inter-synchronization control should be used.

The intra-stream synchronization keeps the timing relation between MUs of the same stream. It outputs MUs to the destination at the same intervals as the generation ones at the source. On the other hand, inter-stream synchronization tries to reconstruct the temporal relations between the MUs of all the related streams.

The algorithm that it is proposed for the synchronization of the three multimedia streams is the enhanced Virtual-Time Rendering (VTR) media synchronization algorithm [27]. The main difference that the enhanced VTR has over the normal VTR [28] is that the VTR enforces intra stream synchronization on one stream. The enhanced VTR enforces intra stream synchronization on all of the streams separately and an inter synchronization control among the streams. The enhanced VTR has already been enforced between haptic and voice data streams with encouraging results [27].

The first thing that should be defined is which stream is the master stream and which are the slave ones between haptic, audio and video. This mainly depends on the application. If the application is video, or audio, or haptic oriented then the master stream should be either the video, or the audio, or the haptic stream respectively. If the application has neutral interest among the streams, then the master stream can be derived from the QoS that should be enforced in each stream. From Table 2 we can infer that the haptic data are by far more sensitive to delay and jitter than the other streams. This means that the master stream should be the haptic stream. The enhanced VTR will firstly enforce the intra synchronization on all of the streams with the VTR algorithm. Based on the scheduled outputs of the MUs of the haptic stream, it will try to enforce inter synchronization with the other streams.

In the VTR algorithm, the *ideal target output time* [29] x_n of the *n-th* MU (n = 1, 2, ・・・) is defined as the time at which the MU should be output in the case where jitter is always smaller than an estimate J_{max} of the maximum jitter (that is, the buffering time of the first MU [28]). Let T_n, A_n, and D_n denote the generation time, arrival time, and output time, respectively, of the n-th MU.

The ideal target output time x_n is calculated as follows [29]:

$$x_1 = \begin{cases} D_1 (= A_1 + J_{max}), & \text{if } D_1 - T_1 \le \Delta_{al} \\ T_1 + \Delta_{al}, & \text{otherwise} \end{cases} \quad (1)$$

$$x_n = x_1 + (T_n - T_1) \qquad (n \ge 2), \quad (2)$$

where Δ_{al} denotes the *maximum allowable delay*.

When jitter is larger than J_{max}, some MUs cannot be output at their ideal target output time. Therefore, the *target output time* [28] t_n of the n-th MU is introduced, which is calculated by adding/subtracting a delay (i.e., *slide time*) to/from the ideal target output time. Let t_n^* and ΔS_n denote the *modified target output time* and the *slide time*, respectively. Then, t_n and t^*_n are given by

$$t_1 = x_1, \quad (3)$$

$$t_n = x_n + \sum_{i=1}^{n-1} \Delta S_i \qquad (n \ge 2) \quad (4)$$

$$t_n^* = t_n + \Delta S_n \qquad (n \ge 2) \quad (5)$$

where $\Delta S_1 = 0$.

By comparing the arrival time A_n and the target output time t_n, the client determines the *scheduled output time* [28] d_n ($n \ge 2$) as follows:

$$d_n = \begin{cases} t_n^*, & if\ A_n \le t_n \\ A_n, & otherwise \end{cases} \quad (6)$$

In the former case of Eq. (6), the *target output time* is advancing (i.e., the virtual-time contraction, in which $\Delta S_n < 0$); when $\Delta S_n = 0$, we set $D_n = d_n$. The latter case of Eq. (6) delays the *target output time* (i.e., the virtual-time expansion, in which $\Delta S_n > 0$). In the latter, when multiple MUs have the same scheduled output time, the MU which has the largest sequence number among them is outputted and the other MUs are skipped.

An MU (say the n-th MU) which is not skipped has the output time $D_n = d_n$.

(a) Virtual-Time Expansion

When $d_n - t_n > T_{h2} (> 0)$, we set $\Delta S_n = d_n - t_n$, where T_{h2} is a threshold value which we use so as to decide whether the target output time should be delayed or not [28].

(b) Virtual-Time Contraction

When $A_n \le t_n$, the target output time of the *n-th* MU is advanced. $d_n = max(t_n - r, x_n, A_n)$ and $\Delta S_n = -min(r, \sum_{i=1}^{n-1} \Delta S_i)$ [29] when $t_n - T_n > \Delta_{al}$, or when a certain period of time (say $T_{nodelay}$) has elapsed since the last late arrival or the last virtual-time contraction, where r is a positive constant. There is a possibility that $d_n \le D_m$ $(n > m)$ in the case of the virtual-time contraction, where m is the sequence number of the last output MU. In this case, the *n-th* MU is skipped.

After the calculation of the output time of each haptic MU with the VTR algorithm, the inter-synchronization among the other streams should be enforced. The inter-synchronization will be made at the timestamp that each MU carries. The stream with the highest update rate is the haptic stream, that is 1 KHz, which means that there is an output every 1 ms. Audio

and Video streams have much lower output time, which means that the timestamp of their MU could easily be synchronized with timestamps of the haptic MUs.

5.2 Temporal Prediction Structure

An important factor that should be taken into consideration for the real-time synchronization of a haptic stream with an HEVC encoded video is the temporal prediction structure that has to be used for the video encoding. As mentioned in section 2, there are three kinds of temporal prediction structures. The first one is the intra-only configuration. Each picture in this kind of configuration is encoded as an Instantaneous Decoder Refresh (IDR) picture. This means that no temporal reference pictures are used. Each frame is independent of the others. A graphical representation of Intra-only configuration is shown in Figure 2. The number next to each frame indicates the encoded and decoded order of the frame. It is understood that in a real-time tele-system, as the frame is captured from the camcorder, it is instantly encoded; it is transferred through the Internet and is decoded from the destination. The negative aspect of this configuration is that it produces extremely high bit rates, which are prohibitive for real-time transferring of data through the Internet.

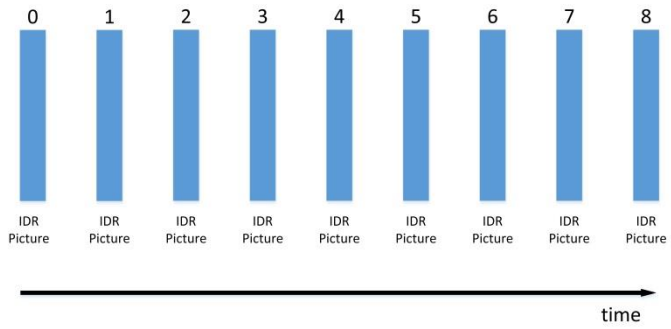

Fig. 2 Graphical presentation of Intra-only configuration

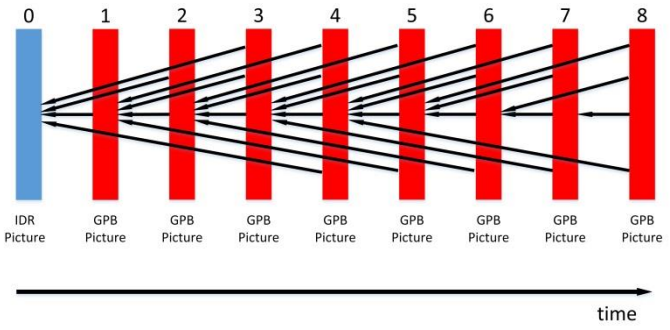

Fig. 3 Graphical presentation of Low-Delay configuration

Another temporal prediction structure of HEVC is the Low-Delay configuration. A graphical presentation of the Low-Delay configuration is depicted in Figure 3. The first picture of this configuration is an IDR frame. It is encoded independently. All the other subsequent frames are encoded based on this frame. The encoded and decoded order of the frames is the same as the display order. This configuration is usually proposed for real-time systems as it exhibits the shortest delay of the video transport. The bit rate of this configuration is lower than the intra configuration.

If we assume for simplicity that the delay time of the network between the source and the destination in a remote system is t_{net}, the mean computational time for the encoding of one frame is t_{en}, and the mean computational time for the decoding of one frame is t_{dec}, then the average delay time $t_{Low-Delay}$ for a video transferring of a Low-Delay configuration is given by the Equation:

$$t_{Low\text{-}Delay}=t_{en\text{-}LD}+t_{net}+t_{dec\text{-}LD} \quad (7)$$

The temporal prediction of the Random-Access configuration is depicted in Figure 4. It is understood that the encoded and the decoded time of the frames is deferent from the display order of the frames.

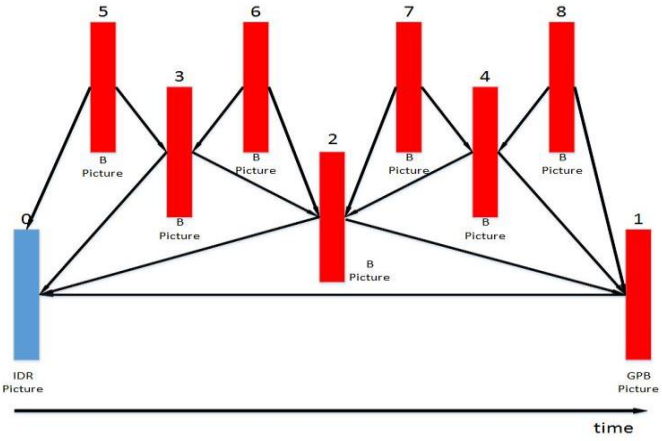

Fig. 4 Graphical presentation of Random-Access configuration

The frames are separated in group of pictures (GOP). If the first frame of the group is encoded as an IDR frame, then the GOP is called *closed GOP*. If the first frame is encoded as a Clean Random Access (CRA) picture then the GOP is called *open GOP*. The strong asset of this configuration is that it demands less encoding time than the other inter-prediction configurations. This feature is very important for applications that transfer data through the Internet. The delay time for this configuration is depended on the encoded process.

If we assume that the size of the GOP is gop_{size} and the interval time between successive frames is $t_{fr}=1/fps$, the delay time $t_{Random\text{-}Access}$ for a video transferring of a Random-Access configuration is given by the Equation:

$$t_{Random\text{-}Access}=gop_{size}*t_{fr}+gop_{size}*t_{en\text{-}RA}+t_{net}+$$
$$+gop_{size}*t_{dec\text{-}RA} \quad (8)$$

If we consider the whole encoding, transferring and decoding process as a pipeline, then the delay time $t_{Random\text{-}Access}$ of (8) can be significantly smaller. The first thing that should be determined is the encoding order of each frame. The frame that has to be encoded first is f_0. The second encoded frame is f_{gop}. The third inevitably encoded frame is $f_{gop/2}$. The next encoded frame could be either $f_{gop/4}$ or $f_{gop3/4}$. If we want to save time, then the frame that has to be encoded next is the $f_{gop/4}$. This is due to the fact that the encoded order is the same with the decoded order. If $f_{gop/4}$ is encoded first, it will be decoded first and it will be available for display on the receiver a little earlier. The next frame that has to be encoded is $f_{gop3/4}$. For gop=4 the graphical presentation of Random-Access configuration is shown in Figure 5.

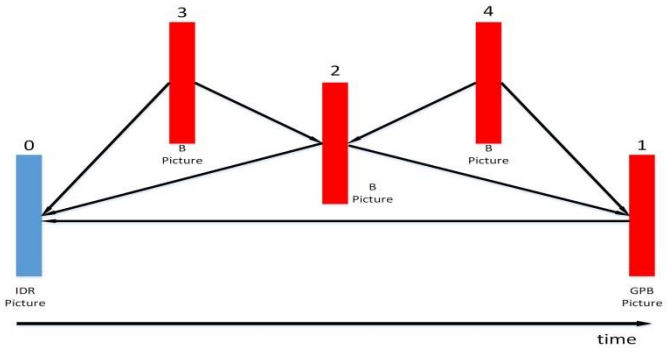

Fig. 5 Graphical presentation of Random-Access configuration for gop=4.

The delay time for gop=4 is given by the Equation:

$$t_{Random\text{-}Access\text{-}4}=3*t_{fr}+3*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (9)$$

The gop should be a power of 2, so that a hierarchical frame prediction structure could be made. With the same process, for gop=8 the graphical presentation of the GOP is shown in Figure 4. The encoded order of the frames is shown as a number next to the frame.

The system will start to show the first decoded frame f_0, $D_1= t_{fr}$ sec before the second frame f_1 of the GOP is decoded. In order for the second frame f_1 to be decoded, all the frames of the GOP have to appear, which means after $D_2=t_{fr}*gop$ time. After all the GOP appear, the encoder will start to encode the frames with the exact order f_8, f_4, f_2, f_6, f_1 as shown in figure 4 (The f_0 frame will already have been encoded during the D_2 interval). This means that the whole system will wait another $D_3=5*t_{en\text{-}RA}$ secs. When the frame f_1 is encoded, it will be sent through the network to the decoder with network delay time $D_4=t_{net}$. The decoder will be ready to show the f_1 frame after $D_5= t_{dec\text{-}RA}$ decoding time.

The summarized delay time for gop=8 is given by the Equation:

$$t_{Random\text{-}Access\text{-}8}=-D_1+D_2+D_3+D_4+D_5$$
$$\equiv$$
$$t_{Random\text{-}Access\text{-}8}=7*t_{fr}+5*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (10)$$

The delay time of Equation (10) may be reduced if we change the order of the encoded frames of figure 4 without changing the correlation between the frames. Since the correlation between the encoded frames is not changing, the encoding time and the data rate of the whole GOP remains the same.

If we try to always encode the left "available" frame of the hierarchical structure, then the encoded order of the frames is shown in Figure 6.

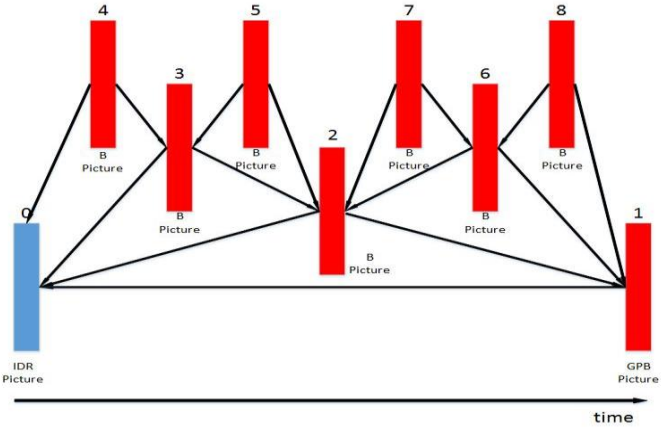

Fig. 6 The proposed encoding order inside the GOP=8

The modified delay time for gop=8 is given by the Equation:

$$t'_{Random\text{-}Access\text{-}8}=7*t_{fr}+4*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (11)$$

$$dt=t_{Random\text{-}Access\text{-}8}-t'_{Random\text{-}Access\text{-}8}=t_{en\text{-}RA} \quad (12)$$

The delay time is reduced by $dt=t_{en\text{-}RA}$, as given by Equation (12). This reduction is due the fact that the encoder will start to encode the frames with the exact order f_8, f_4, f_2, f_1 as shown in figure 6. This means that $D_3'=4*t_{en\text{-}RA}$.

Equation (11) is generalized for gop as:

$$t_{Random\text{-}Access\text{-}gop}=(gop-1)*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+$$
$$+t_{net}+t_{dec\text{-}RA} \quad (13)$$

Equation (13) can be theoretically explained as follows.

When the GOP starts, the first frame (f_0) appears. The encoder starts to encode f_0. When the encoding of the frame f_0 finishes, the encoded frame is sent through the Internet to the decoder. Until f_{gop} appears at t=gop*t_{fr}=gop/fps, the encoder buffers the frames f_1, f_2,..,f_{gop-1}. When f_{gop} appears, the encoder starts to encode the buffered frames with the following order: f_{gop}, $f_{gop/2}$, $f_{gop/4}$, $f_{gop/8}$, …,$f_{gop/gop=1}$. In the meantime every frame that is encoded is sent through the Internet to the decoder.

At t_2, the frame f_1 has been encoded, has travelled through the Internet, has reached the decoder and has been decoded.

$$t_2=gop*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (14)$$

The frame f_0 should reach the decoder and be decoded at

$$t_1=t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA} \quad (15)$$

In case of a real-time encoding we have to accept that

$$t_{fr}>t_{en\text{-}RA}+t_{dec\text{-}RA} \quad (16)$$

For the Random-Access configuration we have

$$gop>1 \quad (17)$$

which means that Equations (16) and (17) lead to

$$t_{en\text{-}RA}<gop*t_{fr}$$

$$\equiv$$

$$t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA}<gop*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+$$

$$+t_{net}+t_{dec\text{-}RA}$$

$$\equiv$$

$$t_1<t_2 \quad (18)$$

The frame f_0 has to come out of the decoder buffer at least t_{fr} time before the decoding of the f_1 at t_2. Therefore, we should have

$$t_1<t_2-t_{fr}$$

$$\equiv$$

$$t_{en\text{-}RA}+t_{net}+t_{dec\text{-}RA}+t_{fr}<gop*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+$$

$$+t_{net}+t_{dec\text{-}RA}$$

$$\equiv$$

$$t_{fr}<gop*t_{fr}+(\log_2(gop))*t_{en\text{-}RA}$$

$$\equiv$$

$$0<(gop-1)*t_{fr}+(\log_2(gop))*t_{en\text{-}RA} \quad (19)$$

Inequality (19) is true as both of the totalizers are positive. This means that the video stream will starts at

$$t_{Random\text{-}Access\text{-}gop}=t_2-t_{fr}$$

$$\equiv$$

$$t_{Random\text{-}Access\text{-}gop}=(gop-1)*t_{fr}+(\log_2(gop)+1)*t_{en\text{-}RA}+$$

$$+t_{net}+t_{dec\text{-}RA} \quad (20)$$

Analyzing equations (7) and (20) we come to the conclusion that the temporal prediction of the HEVC that will be chosen for a tele-haptic system is depended on the available bandwidth of the network, the delay of the network t_{net}, the mean encoding t_{en} and decoding t_{dec} time for each configuration, the frame rate (fps) of the video stream, the data rate that each configuration produces, and the QoS that the specific tele-haptic application requires.

5.3 Computational Cost

In order to estimate the level of the encoding time, a personal pc Intel core i3 2100 at 3.1 GHz with 4 GB RAM has been used. For HEVC encoding, the HEVC Test Model HM 16.2 has been used. The video sample was the mobile_cif YUV series [30] with 352×288 resolution at 24 Hz with a duration time of 10 sec. The encoding time and the data rate of the video sample are shown in Table 3.

Table 3 Computational Cost of HEVC

	COMPUTATIONAL COST of HEVC (for 10 sec video)				
Inter Prediction	***QP***	***RAM (MB)***	***Data Rate (Kbps)***	***Encoding Time (sec)***	***PSNR***
Intra-Only	32	20.9	3514	341	33.18
Low-Delay	32	47.8	269	1209	31.18
Random Access	32	64.5	290	807	31.66
Intra-Only	27	21.1	5266	374	37.26
Low-Delay	27	47.4	641	1491	34.60
Random Access	27	64.7	584	1012	34.76

From Table 3 it is understood that the data rate and computational time of HEVC are strongly dependent on the Quantization Parameter (QP) and the inter prediction configuration. They are inversely proportional. The Low-Delay configuration has much higher encoding t_{en} time than the other configurations. The Intra-Only configuration has the smallest encoding time but the higher data rate. The Random-Access configuration has neutral results regarding encoding time and data rate. The encoding time t_{en} of all the configurations is much higher than the time gap between the successive frames t_{fr}, which means that the real-time encoding is not feasible. The real-time encoding can only be succeeded with the help of parallel processing, a case study that is not in the interest of this paper.

The Intra-Only configuration has the biggest output of data rate from all the configurations. The Low-Delay configuration outputs a higher data rate than Random-Access. On the other

hand the Intra-Only configuration has the smallest encoding time from the other configurations. A more thorough investigation of the HEVC encoding and decoding can be found in [6].

In order to compare the computational cost and the produced data rate of the HEVC with the H.264/AVC, the same video, on the same computer, with the same QP for both encoders were used. The configuration of the H.264 was altered in order to resemble the inter prediction configurations of the HEVC. The outcomes of the H.264/AVC are presented in Table 4.

Both encoders have similar PSNR, which means that the quality of the decoded video is similar. The encoding time of the H.264 is, in most of the cases, by far longer than the HEVC's. It can be noticed that the encoding time of the H.264 is not inversely proportional to the factor QP. H.264 uses a lot more RAM from the HEVC. The data rate of the HEVC, in most cases, is smaller than the H.264's. The only case that the data rate of the H.264 is a little smaller than the HEVC is for the random-Access inter-prediction with QP 32. In that case, the encoding time of the H.264 is 1181% longer than the HEVC, and the RAM usage 173% bigger. For the Low-Delay configurations, the data rate of the HEVC is more than 50% smaller than the H.264.

Table 4 Computational Cost of H.264/AVC

	COMPUTATIONAL COST of H.264/AVC (for 10 sec video)				
Inter Prediction	QP	***RAM (MB)***	***Data Rate (Kbps)***	***Encoding Time (sec)***	***PSNR***
Intra-Only	32	32.3	3948	1191	33.99
Low-Delay	32	95.9	652	2983	33.37
Random Access	32	111.8	270	9528	32.58
Intra-Only	27	32.3	5974	1416	37.61
Low-Delay	27	80.3	1490	2702	36.70
Random Access	27	112.7	644	6487	35.43

5.4 Data Rate Reduction

In the case of a tele-haptic application that is sensitive to the data rate, the Intra-Only configuration should be avoided. In that case, apart from video, the data rate of the haptic stream should be minimized as well. Most of the haptic interfaces [31] produce haptic MU at a rate of 1 KHz. This rather high packet rate is often difficult to transfer through the Internet.

One interesting technique that is proposed for data rate reduction is the dead-reckoning technique [20]. Dead-reckoning can keep the output rate of the haptic media at 1 KHz by prediction and convergence. The haptic source compares the position of the haptic interface to the predicted one. If the difference between the two positions becomes larger than a threshold value, the real position information of the haptic interface is transmitted. This technique has encouraging results in the case of network congestion.

An additional technique, for packet rate reduction, is the packetization interval of the haptic MU [32]. If packetization interval is enforced every P ms on the haptic stream, then the packet rate of the haptic stream can be reduced by a factor of P=8 or P=16, depending on the delay of the system. To achieve information compression, differential coding and quantization is enforced inside the intervals. Each MU has strong correlation with its nearby MU. The packet size of its interval could be reduced from 20+24P bytes to 20+24+3(P-1) bytes. This means that for P=8, the data rate reduction will be at 69.34%. Of course, the packetization interval of the haptic MU adds an extra delay of P ms to the haptic stream. If the delay of the video stream given from Equations (7) or (13), is longer than the delay of the packetized haptic stream, then a packetization interval could be applied.

5.5 Network Conditions Of The Internet

In order to infer the network conditions of the Internet, and determine the network delay t_{net}, 3000 ICMP packets were sent through the Internet between two destinations in different regions. The destinations that were chosen were the city of Thessaloniki, Greece and the city of Grevena, Greece. The distance between the two cities is 170 Km. In order to eliminate the relevance of the time of day, we repeated the experiment every 3 hours for 24 hours. In order to eliminate the relevance of the client ISP connection of the two destinations, we sent the ICMP packet through two different ISP. In the first experiment, the first client connection was a simple adsl connection of 24 Mbps, while the other client was connected to a private optical network, GRNET [33], part of the pan-European GEANT network with speeds up to 4×10 Gbps. In the second experiment both of the clients were directly connected to the private optical network, GRNET. We managed to measure the mean end to end Delay between the two destinations, the Jitter, the packet loss, and the number of Hops between the two destinations. The results of these experiments are shown in Table 5.

Table 5 Internet Network Conditions Between Thessaloniki-Grevena, Greece

	INTERNET STATUS			
CONNECTED CITIES	***AVG. DELAY (ms)***	***Standard DELAY Deviation (ms)***	***PACKET LOSS (%)***	***No. HOPS***
GREVENA – THESSALONIKH THROUGH GRNET	19.12	1.70	0	5
GREVENA – THESSALONIKH THROUGH ADSL LINE	53.19	5.31	0.11	8

It is understood that when the source and the destination are connected directly to the GRNET, the network conditions satisfy all the requirements of Table 2. In the case of the

simple adsl connection, all multimedia streams can travel through the Internet easily, except of the haptic stream, as it is very sensitive to delay and jitter.

5.6 Flowchart of HEVC Encoding for Real-Time Transferring of Video, Audio, and Haptics

If all the variables t_{fr}, t_{en}, t_{net}, t_{dec} are known from the above procedures, then the appropriate inter-prediction configuration can be chosen from Equations (7) and (13) and the limits of Table 2. All the above procedures are integrated in the flowchart of Figure 7.

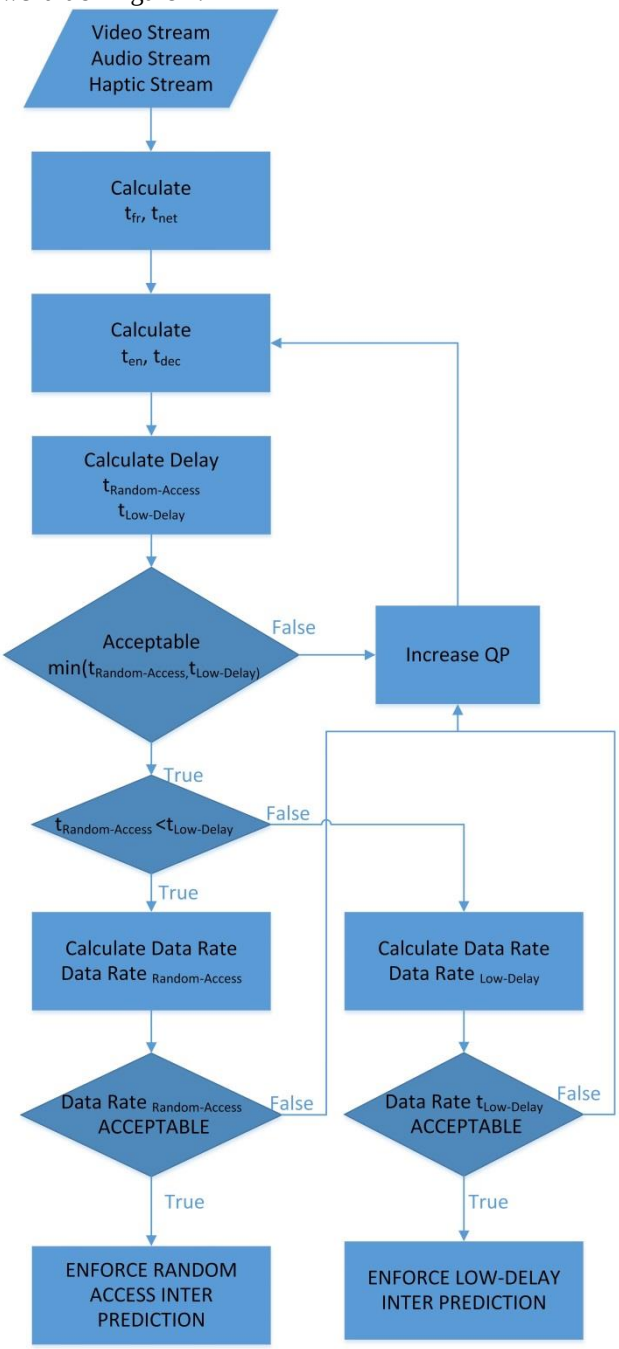

Fig. 7 Flowchart of HEVC encoding for real-time transferring of video, audio, and haptics.

6. Conclusions And Future Work

It is obvious that the synchronization of real-time media streams that travel through the Internet is rather a challenging process. The new HEVC technique shows rather promising results as it manages to reduce the bit rate to 50% comparing to its predecessor H.264/AVC. The available Internet bandwidth is no longer an obstacle. The only obstacles in live media streaming is the end to end delay and the jitter of the network. These obstacles are rather obvious when one of the media streams is haptic data.

This paper proposed an efficient algorithm for transferring a real-time HEVC stream with haptic data through the Internet. The H.264 and the HEVC are compared. The network conditions of the Internet were measured. The transferring delay of all the inter prediction configurations of the HEVC were defined.

If the QoS of the network, the encoding and decoding time of the HEVC are known, then the correct temporal prediction of the HEVC could be chosen. It has been proven that the encoding order of the frames inside a GOP could play a major role in the delay of the system. Comparative tests between H.264 and HEVC have been undertaken.

The synchronization techniques that are proposed in this paper compensate the barrier of the low tolerance that haptic data have to jitter and the delay.

It has already been scheduled to evaluate the algorithm for real-time transferring through the Internet of HEVC video, audio and haptic streams in real word experiments.

References

1. J. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, pp. 1669–1684, Dec 2012.
2. P. Hanhart, M. Rerabek, F. De Simone, and T. Ebrahimi, "Subjective quality evaluation of the upcoming hevc video compression standard," *Proc. SPIE*, vol. 8499, pp. 84990V–84990V–13, Oct 2012.
3. R. Garcia and H. Kalva, "Subjective evaluation of hevc and avc/h.264 in mobile environments," *IEEE Tran. on Consumer Electronics*, vol. 60, pp. 116–123, Feb 2014.
4. J. Nightingale, Q. Wang, and C. Grecos, "Benchmarking real-time HEVC streaming," *Proc. SPIE*, vol. 8437, pp. 84370D–84370D–14, Jun 2012.
5. A. Panayides, Z. Antoniou, M. Pattichis, and C. Pattichis, "The use of h.264/avc and the emerging high efficiency video coding (HEVC) standard for developing wireless ultrasound video telemedicine systems," in *Proc. Forty Sixth Asilomar Conf. on Signals, Systems and Computers (ASILOMAR)*, Nov 2012, pp. 337–341.
6. F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Tran. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.
7. I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, and H. Woo-Jin, "Hm9: High efficiency video coding (HEVC) test model 9 encoder description," in *9th JCT-VC Meeting, Switzerland*, 2012, July, pp. 10–11.
8. G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, Dec 2012.

9. C. C. Chi, M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel scalability and efficiency of HEVC parallelization approaches," *IEEE Tran. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1827–1838, Dec 2012.
10. M. Alvarez-Mesa, C. Chi, B. Juurlink, V. George, and T. Schierl, "Parallel video decoding in the emerging HEVC standard," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2012, pp. 1545–1548.
11. E. Saddik, "The potential of haptics technologies," *IEEE Instrumentation Measurement Magazine*, vol. 10, pp. 10–17, Feb 2007.
12. M. Eid, J. Cha, and A. El Saddik, "Admux: An adaptive multiplexer for haptic-audio-visual data communication," *IEEE Tran. Instrument. and Measurement*, vol. 60, pp. 21–31, Jan 2011.
13. K. Iwata, Y. Ishibashi, N. Fukushima, and S. Sugawara, "QoE assessment in haptic media, sound, and video transmission: Effect of playout buffering control," *Comput. Entertain.*, vol. 8, pp. 12:1–12:14, Dec 2010.
14. N. Suzuki and S. Katsura, "Evaluation of QoS in haptic communication based on bilateral control," in *IEEE Int. Conf. on Mechatronics (ICM)*, Feb 2013, pp. 886–891.
15. E. Isomura, S. Tasaka, and T. Nunome, "A multidimensional qoe monitoring system for audiovisual and haptic interactive IP communications," in *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2013, pp. 196–202.
16. A. Hamam and A. El Saddik, "Toward a mathematical model for quality of experience evaluation of haptic applications," *IEEE Tran. Instrument. and Measurement*, vol. 62, pp. 3315–3322, Dec 2013.
17. H. Morimitsu, S. Katsura, and M. Tomizuka, "Design of force compensator with variable gain for bilateral control system under time delay," in *IEEE Int. Symposium on Ind. Electron. (ISIE)*, May 2013, pp. 1–6.
18. A. Alturki and M. Alnuem, "Study of the effects of using some QoS mechanisms on haptic transmission - using opnet modeler," in *IEEE Int. Workshop on Haptic Audio Visual Environments and Games (HAVE)*, Oct 2011, pp. 94–101.
19. Y. Ishibashi, S. Tasaka, and Y. Tachibana, "Adaptive causality and media synchronization control for networked multimedia applications," in *IEEE Int. Conf. on Communications*, vol. 3, Jun 2001, pp. 952–958 vol.3.
20. A. Bartl, M. Diaz-Cacho, A. Barreiro, and E. Delgado, "Passivity framework and traffic reduction for the teleoperation of a gantry crane," in *39th Ann. Conf. IEEE Ind. Electron. Society (IECON)*, Nov 2013, pp. 3675–3680.
21. S. Sakaino, T. Sato, and K. Ohnishi, "Precise position/force hybrid control with modal mass decoupling and bilateral communication between different structures," *IEEE Tran. on Industrial Informatics*, vol. 7, pp. 266–276, May 2011.
22. Q. Zeng, Y. Ishibashi, N. Fukushima, S. Sugawara, and K. Psannis, "Influences of inter-stream synchronization errors among haptic media, sound, and video on quality of experience in networked ensemble," in *IEEE 2nd Global Conf. on Consumer Electronics (GCCE)*, Oct 2013, pp. 466–470.
23. T. Schierl, M. Hannuksela, Y.-K. Wang, and S. Wenger, "System layer integration of high efficiency video coding," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 22, pp. 1871–1884, Dec 2012.
24. Y. Ishibashi and S. Tasaka, "A comparative survey of synchronization algorithms for continuous media in network environments," in *Proc. 25th Ann. IEEE Conf. on Local Computer Networks*, Nov 2000, pp. 337–348.
25. Y. Ishibashi, S. Tasaka, and H. Ogawa, "A comparison of media synchronization quality among reactive control schemes," in *IEEE Proc. 14th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Apr 2001, pp. 77–84 vol.1.
26. Y. Ishibashi, S. Tasaka, and A. Tsuji, "Measured performance of a live media synchronization mechanism in an atm network," in *IEEE Int. Conf. on Communications*, vol. 3, Jun 1996, pp. 1348–1354.
27. Y. Ishibashi, T. Kanbara, and S. Tasaka, "Inter-stream synchronization between haptic media and voice in collaborative virtual environments," in *Proc. 12th Ann. ACM Int.l Conf. on Multimedia*, New York, 2004, pp. 604–611.
28. Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," in *IEEE Proc. 14th Annual Joint Conf. of the IEEE Computer and Communications Societies. Bringing Information to People (INFOCOM)*, Apr 1995, pp. 1010–1019 vol.3.
29. S. Tasaka, T. Nunome, and Y. Ishibashi, "Live media synchronization quality of a retransmission-based error recovery scheme," In Conference Record of IEEE ICC'00, pages 1535‑1541, June 2000.
30. F. Fitzek and M. Reisslein. Video traces for network performance evaluation: Yuv 4:2:0 video sequences, http://trace.eas.asu.edu/yuv/yuv.html
31. A. Silva, O. Ramirez, V. Vega, and J. Oliver, "Phantom Omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conf. (CERMA)*, Sept 2009, pp. 193–198.
32. M. Fujimoto and Y. Ishibashi, "Packetization Interval of haptic media in networked virtual environments," in *Proc. 4th ACM SIGCOMM Workshop on Network and System Support for Games*, New York, 2005, pp. 1–6.
33. The Greek Research and Technology Network - GRNET S.A., https://www.grnet.gr/en.