

Optimal QoE Scheduling in MPEG-DASH Video Streaming

Shin-Hung Chang^{1*}, Min-Lun Tsai¹, Meng-Huang Lee³, Jan-Ming Ho²

¹ Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taipei (Taiwan)

² Institute of Information Science, Academia Sinica, Taipei (Taiwan)

³ Department of Information Technology and Management, Shih Chien University, Taipei (Taiwan)

Received 30 October 2020 | Accepted 13 January 2021 | Published 25 June 2021



ABSTRACT

DASH is a popular technology for video streaming over the Internet. However, the quality of experience (QoE), a measure of humans' perceived satisfaction of the quality of these streamed videos, is their subjective opinion, which is difficult to evaluate. Previous studies only considered network-based indices and focused on them to provide smooth video playback instead of improving the true QoE experienced by humans. In this study, we designed a series of click density experiments to verify whether different resolutions could affect the QoE for different video scenes. We observed that, in a single video segment, different scenes with the same resolution could affect the viewer's QoE differently. It is true that the user's satisfaction as a result of watching high-resolution video segments is always greater than that when watching low-resolution video segments of the same scenes. However, the most important observation is that low-resolution video segments yield higher viewing QoE gain in slow motion scenes than in fast motion scenes. Thus, the inclusion of more high-resolution segments in the fast motion scenes and more low-resolution segments in the slow motion scenes would be expected to maximize the user's viewing QoE. In this study, to evaluate the user's true experience, we convert the viewing QoE into a satisfaction quality score, termed the Q-score, for scenes with different resolutions in each video segment. Additionally, we developed an optimal segment assignment (OSA) algorithm for Q-score optimization in environments characterized by a constrained network bandwidth. Our experimental results show that application of the OSA algorithm to the playback schedule significantly improved users' viewing satisfaction.

KEYWORDS

Integer Programming, MPEG-DASH, Motion Vector, Quality Of Experience (QoE), Video Streaming.

DOI: 10.9781/ijimai.2021.06.003

I. INTRODUCTION

RECENTLY, the rapid increase in network bandwidth has enabled Internet services to offer users more diverse choices with the result that Internet services have become inseparable from daily life. Owing to the changes in the habits and customs of users seeking entertainment, video streaming services have become part of life. For example, people can instantly receive the latest news, watch movies, or listen to music online. Although the increase in network bandwidth has brought additional Internet services, network congestion has become more problematic, especially for video streaming services. HTTP is a popular protocol for sending and receiving web pages. The reasons for adopting HTTP include its ability to easily penetrate client firewalls because network port 80 is always open for browsing webpages. Moreover, HTTP is compatible with content delivery networks (CDNs). Thus, it is helpful for deploying CDNs [1]. The well-known video streaming technology based on HTTP is dynamic adaptive streaming over HTTP (DASH), also known as MPEG-DASH [2], [3], [4]. DASH technology divides entire streaming videos into

a series of small video segments and sequentially transmits each video segment to the users. The DASH client selects different video segments according to the network bandwidth conditions, and the length of each video segment is fixed.

With the advent of the 5G era, network bandwidth has increased tremendously, and video streaming providers such as YouTube, Netflix, Hulu, and Tudu have not only provided users with faster and more convenient services, but also improved quality of service (QoS) [5]. Although DASH streaming can provide different video resolutions to users according to the network bandwidth, it cannot ensure user viewing satisfaction. Therefore, video streaming service providers can improve user satisfaction by reducing the influence of network congestion. Although a certain QoS can be maintained, user satisfaction is not ensured. Quality of experience (QoE) is another indicator that defines user satisfaction, which reflects the subjective feelings of the users and originates directly from the user. Thus, QoE can be used to provide services that meet user expectations. The factors influencing QoE can be divided into three types: human, system, and context [6]. Briefly, the service either satisfies or disappoints the user.

Traditional DASH video playback scheduling algorithms often focus on smooth playback of the entire video when the network bandwidth is constrained [7], [8], [9]. These algorithms cannot adjust the video playback scheduling according to the user's true

* Corresponding author.

E-mail address: shchang@csie.fju.edu.tw

Please cite this article in press as:

S. H. Chang, M. L. Tsai, M. H. Lee, J. M. Ho. Optimal QoE Scheduling in MPEG-DASH Video Streaming, International Journal of Interactive Multimedia and Artificial Intelligence, (2021), <http://dx.doi.org/10.9781/ijimai.2021.06.003>

viewing QoE indicators. In this paper, we discuss the influence of different video resolutions on user viewing satisfaction based on two evaluation methods. The first method involves a series of click density (CD) experiments that can be performed to collect factors with which users are dissatisfied and define a quality score, termed the Q-score, based on the unsatisfactory click counts. However, many different types of videos need to be quantized into Q-scores, and it is impractical to perform CD experiments for each video. Instead, we proposed a second method in which the Q-score can be automatically matched with the amounts of motion vector variations in different scenes with different resolutions, thereby improving the efficiency of Q-score quantification.

Finally, we propose an algorithm, the optimal segment assignment (OSA) algorithm, which uses integer programming to schedule DASH video streaming for QoE optimization [10]. The main objective of OSA in terms of video playback scheduling is that when a large Q-score difference exists between resolutions, the OSA algorithm assigns a high-resolution video segment, and vice versa. This design concept can enhance the overall satisfaction of users and effectively utilize network bandwidth.

The remainder of this paper is organized as follows. Section II introduces related work. Section III presents the problem formulation and proposed algorithms, including the click density (CD) experimental environment and method, motion vector variation measurement using the block matching algorithm, distribution of unsatisfied click counts with different resolutions, the method for quantifying the Q-score, and the proposed OSA algorithm. Section IV presents the experimental results for user satisfaction obtained using our proposed method and the traditional popular algorithm. Finally, Section V presents our conclusions and topics for future work.

II. RELATED WORK

A. User Feeling Measurement

Many factors affect the feelings of users, and it is not easy to collect and quantify these subjective feelings. Many previous researchers have focused on mapping network-related factors, such as the average playback bit rate, resolution switch count, buffer status, and download video quality, to judge users' quality of experience. Sakamoto et al. [11] proposed a bitrate selection method for adaptive video streaming for MPEG-DASH to improve the QoE by minimizing the bit rate fluctuation. Cetinkaya et al. [12] constructed a software-defined network (SDN) to reroute DASH flows to provide a fairness streaming service among DASH clients. Gao et al. [13] provided a deep learning model to characterize personalized QoE with temporal, spatial, and periodic correlations. With this classification of characteristics, they claimed that they could improve the personalized QoE for each specific user. They claimed that the results were more effective for QoE evaluation. Bentalb et al. [14] proposed a software-defined network, named SDNDASH, based on dynamic network resource allocation and management. This architecture can manage and allocate network resources dynamically to improve the QoE for each user. Li et al. [15] proposed a QoE-driven mobile edge caching placement mechanism for dynamic adaptive video streaming with different rate-distortion characteristics. Zhao et al. [16] provided a robust adaptive algorithm at the client side for smoothing the streaming experience. They claimed that this mechanism can work stably under different network conditions. Lee et al. [17] proposed a segment-adjusted scheme based on the playback buffer status and network characteristics of the content. Cao et al. [18] proposed a QoE-friendly resolution-adaptation method that switches resolution less frequently and achieves smooth changes in resolution. Muller et al. [7] proposed a buffer level (BL)

algorithm, which sets a 30s buffer to compensate for large bandwidth variations. The BL algorithm determines the resolution of the next requested video segment according to the state of the buffer occupancy of the client. When the buffer occupancy is at a lower level, a higher-resolution segment is retrieved during the next segment-retrieving cycle. On the other hand, when the buffer occupancy is at a higher level, a lower-resolution segment is retrieved. Alzahrani et al. [19] applied a machine-learning model to handle rate control to select the best network quality. Huang et al. [20] defined an integrated user QoE model and optimally controlled playback freezing, bitrate switch, and video playback quality by stabilizing the client buffer state. Yin et al. [21] used a theoretical approach for controlling video streaming over HTTP. Xin et al. [22] proposed a trunk-based request strategy to guarantee QoE in a P2P-VOD system.

Generally, if network traffic is sufficient to stream smooth video playback to each user, the user will have the best video viewing quality. However, these network-related factors may not truly reflect users' QoE. Chen et al. [23] proposed the OneClick experiment to describe the influences of various network factors, such as bandwidth, loss rate, and delay, on users' listening satisfaction. The original audio clip was divided into different clips according to various network factors, and each clip was cut into several audio segments. To provide the sound clip material to the user as a test, the first sound segment is the original sound segment, and the subsequent audio segments correspond to different situations and are randomly arranged to form a brand new testing sound clip, which does not overlap with the other segments. Because the experiment is intended to test sound satisfaction, the user must wear headphones to listen to the sounds to eliminate interference in the form of external sound, and the experiments are conducted through a keyboard and computer screen display. If the user is dissatisfied with the sound quality of the current segment, the blank key on the keyboard can be tapped, which is designated as an unsatisfactory click record. After the test, the database collected the satisfaction of different users for different audio segments. Because the user reacts by clicking after listening to the sound, there is a slight delay. Therefore, using a modified feedback delay time, it is possible to determine the actual time point of user dissatisfaction. The OneClick experiment is convenient because it can be performed on any computer at any time, as long as the computer system includes a keyboard, monitor, and headphones. In addition, the participants in the experiment did not need training beforehand; rather, they only needed to click on the appropriate key to indicate that the sound was unsatisfactory. Therefore, the OneClick experiment can reflect user listening satisfaction.

Additionally, many previous researchers have studied users' QoE based on the video content and users' viewing characteristics. Yue et al. [24] proposed a hybrid neural network model that integrates a deep neural network (DNN) and a recurrent neural network model (RNN) to learn an attention mechanism for user behavior analysis. Dimopoulos et al. [25] proposed a mechanism for detecting users' QoE degradations with three key influencing factors: stalling, average video quality, and quality variations. Zhao et al. [26] reviewed the QoE strategy for video transmission, including context and human factors. Engelke et al. [27] reviewed several psychophysiology-based QoE assessment methods. Hu et al. [28] proposed a semantic-aware adaption scheme termed SMA-PANDA to adapt video segments to DASH streaming. They used the k-means algorithm to classify motions into three types of video: slow moving, general walking, and rapid moving in a soccer sport movie. They mentioned that three types of video segments can be scheduled in a video playback to achieve a high QoE for users. However, except for the OneClick experiment, previous studies might not touch the users' true feelings. In this paper, we propose a segment assignment algorithm and test real users' viewing QoE using a series of sensory experiments.

B. Motion Vector in Motion Estimation Algorithm

Barjatya et al. [29] compared seven different block-matching algorithms to predict the movement of macroblocks while compressing a video. Among the motion estimation methods, the exhaustive search approach is the most accurate; however, the processing time is relatively long, and the motion estimation search process is employed. Motion estimation involves calculation of the change in the position of an object between two successive frames and the background in a macroblock to corresponding positions, and the calculated amount and direction of movement are recorded in a matrix to form a motion vector. Considering that the time cost is limited, a macroblock cannot be searched in the range of an entire frame. Therefore, the search range must be fixed; generally, the search parameter extends 7 pixels out of a macroblock with a length and width of 16 pixels. The search range will be formed as a rectangle area with a length and width of 25 pixels.

A larger number of search parameters increases the search cost. A macroblock is similar to another macroblock and is calculated on the basis of the cost function. The cost is minimized when the macroblock most closely corresponds to the current macroblock, that is, the most similar macroblock. There are many measures of cost, the most widely known and inexpensive being the mean absolute difference and mean squared error. Hosur et al. [30] used a motion vector for fast-motion estimation. Tourapis et al. [31] proposed an enhanced block-based search algorithm for motion estimation. Arora et al. [32] identified the initial search center dismisses that will be applied in any fast block-matching algorithm to find the motion vectors, rather than using a fixed search pattern. Kamble et al. [33] proposed a modified diamond search algorithm, which employed a small diamond-shaped search pattern in the initial step and a large diamond shape in further steps for handling fast motion estimation.

III. PROBLEM FORMULATION AND PROPOSED ALGORITHM

A. Problem Formulation

In this paper, we propose a measurement method for evaluating users' QoE, namely the Q-score. In this section, we formulate the problem to maximize users' QoE and propose an algorithm for determining a playback schedule that maximizes the Q-score. To define the problem clearly, we present some notations, as listed in Table I.

TABLE I. NOTATIONS FOR PROBLEM FORMULATION AND MODEL

Symbol	Definition
L	Startup latency.
N	Number of video segments in a DASH video.
ΔT	Playback duration of each video segment.
K	Number of different resolutions in a DASH video.
s_i	i^{th} -resolution video segment, where $1 \leq i \leq K$.
$ s_i $	Size of the i^{th} -resolution video segment, where $1 \leq i \leq K$ and $ s_1 > s_2 > \dots > s_K $.
T	A DASH playback schedule consisting of t_j $T = [t_j 1 \leq j \leq N]$, where $t_j \leftarrow s_i$ and $1 \leq i \leq K$.
q_{ij}	The specific Q-score gain while the i^{th} -resolution video segment, s_i , is scheduled into t_j of T .
$Q(T)$	The accumulated Q-score gain in a playback schedule T . $Q(T) = \sum_{j=1}^N q_{ij}$, where $1 \leq i \leq K$.

In this study, the measured network bandwidth R was first assumed to be limited and fixed. The time required to download a video was divided into three periods. The first is the startup latency L , which involves preloading a video to start playback. Second, we assume

that the video has N segments for playback and that the duration of each video segment is ΔT . Therefore, the total download time is $L + (N - 1) \times \Delta T$. Note that the last video segment should be downloaded successfully before it plays back.

In DASH video streaming technology, segments with the same video resolution are encoded with the same data size. We assume that there are K resolutions in total, and the video segments of each resolution are denoted as s_i and $|s_i|$ indicates the size of the segment s_i , where $1 \leq i \leq K$. The K^{th} resolution has the lowest resolution, thus $|s_1| > \dots > |s_K|$. A playback schedule, T , consists of arranging a video segment, s_i , into each time slot, t_j , and formed as $T = [t_j | 1 \leq j \leq N]$, where $t_j \leftarrow s_i$, where $1 \leq i \leq K$. Additionally, while each video segment, s_i , is scheduled into t_j , the specific Q-score gain, q_{ij} , is accumulated in $Q(T)$. Therefore, $Q(T) = \sum_{j=1}^N q_{ij}$, where $1 \leq i \leq K$. The major goal of this study is to find a DASH playback schedule, T , which maximizes $Q(T)$.

To distinguish different user experience levels, we convert users' QoE into a Q-score and use the proposed algorithm to determine a playback schedule that maximizes the Q-score gain. We observed that the Q-score of a high-resolution video segment is greater than that of a low-resolution video segment for the same scenes, as confirmed by our proposed Click Density (CD) experiment, presented in the following sections.

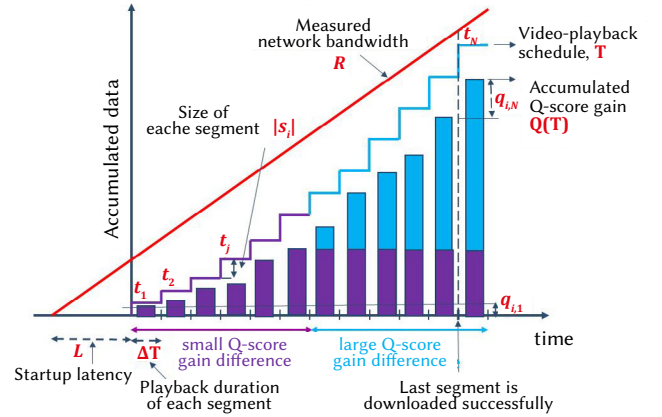


Fig. 1. Video-playback schedule with accumulated Q-score.

Fig. 1 presents an example of the video-playback schedule, T , in which the proposed algorithm allocates a different Q-score gain to video segments with a different resolution in a distinct scene. With the measured network bandwidth, R , low-resolution or high-resolution video segments occur in each playback duration, ΔT , of the playback schedule. In Fig. 1, we assume that the difference in the Q-score gain between high-resolution and low-resolution video segments is small in the first half of the video and large in the second half of the video. Then, the proposed OSA algorithm assigns low-resolution video segments in the area in which the Q-score gain difference between resolutions is smaller. Conversely, high-resolution video segments are chosen when the Q-score gain difference between resolutions is larger. Thus, the proposed OSA algorithm is designed to maximize users' Q-score gain.

Fig. 2 presents our framework for the designed CD experiment and the proposed OSA algorithm for calculating playback scheduling while maximizing the Q-score for video stream V . In our sensory experiments, we merged several different types of video into a test video (as shown in ①) and designed a click density (CD) experiment (as shown in ②) to obtain a Q-score model (as shown in ④) by a subjective test (as shown in ③). Then, we designed a motion vector estimation algorithm to calculate the average motion vector of this test video (as shown in ⑤).

Using regression analysis, we obtained a set of Q-score equations of different resolutions (as shown in ⑥), which transfers the Q-score of different resolutions obtained from the subject test results, QoE(CD), to that obtained by calculating the average motion vector, QoE(MV), (as shown in ⑦). Therefore, we obtain the Q-score of each segment with different resolutions, instead of performing subjective tests for each video stream. Furthermore, given a network condition R , we obtain the optimal playback scheduling with the maximum Q-score, $Q(T)$, by the integer programming algorithm (as shown in ⑧). Hereafter, given a new video stream V (as shown in ⑨) and a network constraint R (as shown in ⑩), the playback schedule with maximum Q-score (as shown in ⑪) is arranged by calculating the average motion vectors (as shown in ⑫) instead of starting the subjective tests all over again.

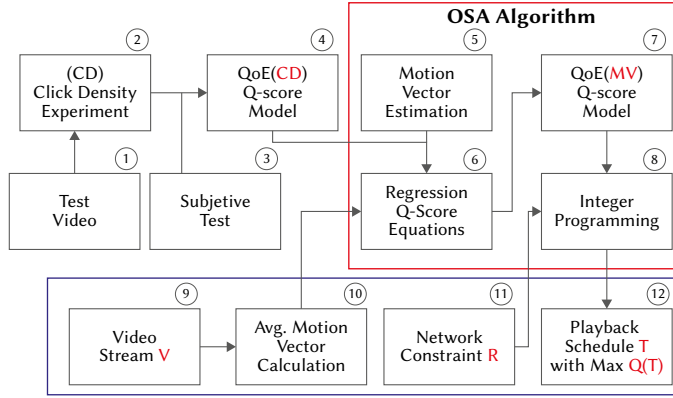


Fig. 2. Framework for the CD experiment and the OSA algorithm.

B. Test Video Sequence for Click Density (CD) Experiment

YouTube is currently one of the largest platforms for video streaming and includes a wide variety of video types [35]. For example, movies can be divided into science fiction films, action adventure films, literary and romantic films, and so on. In addition, YouTube uses a 16:9 aspect ratio player to match computer limitations, and the video resolution is proportional to the recommended aspect ratio. Because different video types correspond to different individual bitrates, different resolutions have different bitrates. To verify the user satisfaction based on the QoE in different video scenes, we chose three different categories—action adventure, sport, and family love films—as test materials. Action adventure films include fast-motion scenes, and the shots move quickly. Family films involve slow-motion scenes and fixed shots. Although sports and action adventure films both include fast-motion scenes, they differ in terms of shot distance. All of the videos were sourced from YouTube: the action adventure films were “Casino Royale” and “The Avengers,” the family love film was “Why Him,” and the sports film was “Lionel Messi - Skills & Goals.”

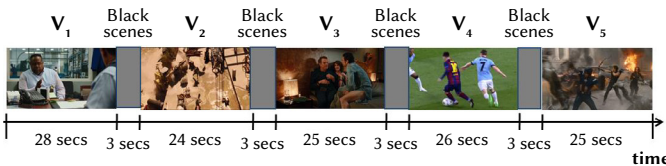


Fig. 3. Test video sequence composed of different types of video.

We merged these different types of video into a test video (as presented in Fig. 3) and inserted fast motion scenes between the slow motion scenes. To prevent adverse effects as a result of watching a video clip from a fast motion scene followed by one from a slow motion scene or conversely, we inserted black scenes with a duration of 3 seconds between consecutive video clips. Because the impact of

resolution on the QoE for different types of videos was the focus of our study, we excluded the two most important influencing factors: sound and subtitles. Sound is an important influencing factor in videos; for example, horror movies are often paired with thrilling sound effects, and sound effects can integrate users into the story. In addition, subtitles significantly influence the visuals at different resolutions. For example, the degree to which the edges of subtitles are blurred is more dramatic in low-resolution videos. Thus, the test video sequence was composed without sound or subtitles. We used FFmpeg open-source tools to cut the video clips to produce the corresponding scenes and merge them into a test video sequence. The test video sequence consisted of five video clips that were presented at 30 fps. Details of the video clips are listed in Table II.

TABLE II. TEST VIDEO SEQUENCE INFORMATION

Video Clip no.	Video Source	Type	Duration (secs)	Movement	
				Camera	Object
v1	Why Him	Family	28	Fixed	Slow
v2	Casino Royale	Adventure	24	Moving	Fast
v3	Why Him	Family	25	Fixed	Slow
v4	Lionel Messi - Skills & Goals	Sport	26	Moving	Fast
v5	The Avengers	Adventure	25	Moving	Fast

Most videos on YouTube have a maximum resolution of 1080p, so we chose resolutions of 1080p, 720p, 480p, and 360p as the test resolutions, and the corresponding bitrates are provided in Table III. The bitrates were chosen to enable users to perceive the differences between the resolutions.

TABLE III. RESOLUTIONS AND CORRESPONDING BITRATES

Resolution	Video size	Bitrate (kbps)
1080p	1920×1080	3000
720p	1280×720	1500
480p	854×480	500
360p	640×360	200

C. Proposed Click Density (CD) Experiment

We followed the concept of the OneClick experiment and modified it as the click density experiment for the video quality of the user experience test. Our click density experiment was modified to include two improvements. First, we changed the response device from the keyboard to the mouse. The click sensitivity with the mouse is higher than the keyboard effect, thus the differences in user satisfaction during the video test could be measured more accurately. Second, to retain the ability to conduct the test on any computer and facilitate the process for the test subjects, we set up a video test website. The following steps were performed to instruct the test subjects on how to use the video test website. Initially, a silent test video with a length of 2 min 20 s was played twice. First, the test video was played exactly as obtained from the source, which means that the highest bitrate and resolution were employed. The second time, one of the four test resolutions was randomly selected. Then, if the subject was dissatisfied with the current video segment during the video playback, the subject could click on the “unsatisfied button.” If the unsatisfied button was clicked consecutively, it turned red to remind the subjects that they were highly dissatisfied with the current state and to facilitate user distinction of the level of dissatisfaction during the test. Finally, we reminded the subjects to use a monitor with at least 1024×768 resolution when performing the video test experiments and recommended not conducting the test on devices such as mobile phones or tablet PCs.

Considering that individuals have different levels of subjective

judgment, we used at least 30 test subjects to obtain data for each resolution. Then, we removed the black scenes between the video clips and modified the response delay times for the subjects. Finally, we aligned the unsatisfied click counts such that they corresponded with each video segment. The basic unit of the video was a video segment, each video segment lasted for 2 s, and the total length of the video was 64 segments.

Fig. 4 shows the average normalized click-count distributions corresponding to the four resolutions. There were 30 test subjects for 1080p, 30 for 720p, 37 for 480p, and 42 for 360p. Video segments 1–14 and 27–39 were slow-motion scenes, whereas video segments 15–26, 40–52, and 53–64 were fast motion scenes. As shown in Fig. 4, the average number of unsatisfied clicks for the high-resolution and low-resolution cases contrast each other. In other words, the lower the resolution, the higher the number of unsatisfied clicks. Moreover, with a low resolution, such as 360p or 480p, the average number of unsatisfied clicks was significantly higher for the fast motion scenes (video segments 15–26, 40–52, and 53–64) than in the slow motion scenes (video segments 1–14 and 27–39). With a high resolution of 720p ps, the same phenomenon is evident. However, the difference at 1080p was insignificant. To summarize, different resolutions have different QoE ranges for different scenes.

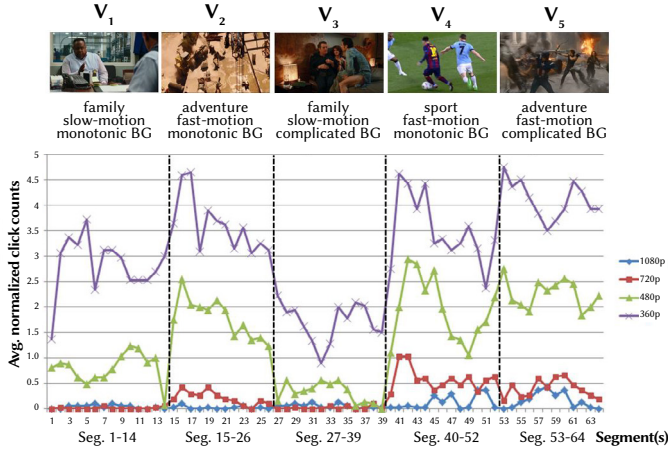


Fig. 4. Normalized dissatisfied click counts distribution.

D. QoE Quantification Q-score

In this section, we present the method for converting the user's QoE into a Q-score, which is defined in this paper. The absolute category rating (ACR) is a method for assessing differences in the QoE level in video or audio tests, and it consists of five levels: excellent, good, fair, poor, and bad. The number of unsatisfied clicks was converted into ACRs and 30 people were randomly selected from the test samples, thus the total number of samples from the four resolutions was 120. Because situations with zero clicks are equivalent to no dissatisfaction, they were evaluated as excellent.

As shown in Fig. 5, in addition to the zero-click case, the other unsatisfied clicks were sorted from smallest to largest according to the satisfaction level, and in the sample used in our CD experiment, the minimum number of clicks was 1 and the maximum was 19.

We then used the quartile method to match the rating quality. Zero unsatisfied clicks are equivalent to the highest Q-score of 5 points (as indicated in Table IV), and the remaining one to four points are allocated by different percentiles defined by the first quartile ($Q1 = 2$), second quartile ($Q2 = 5$), and third quartile ($Q3 = 9$).

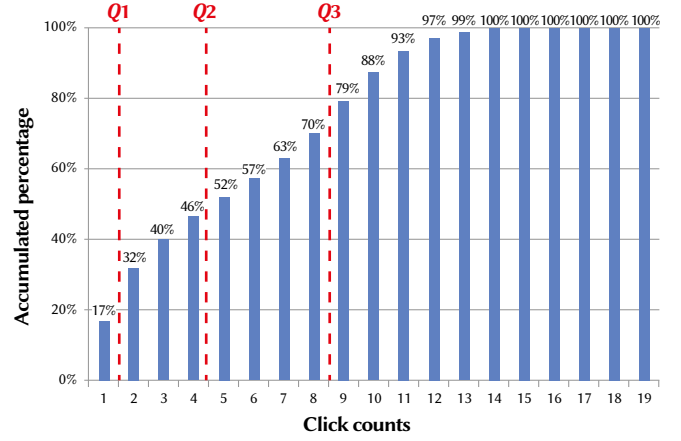


Fig. 5. Distribution of number of clicks excluding the zero click case.

TABLE IV. CORRESPONDENCE BETWEEN NUMBER OF CLICKS AND ACR RATING

Q-score	Number of clicks (x)
5	0
4	$x < 2$
3	$2 \leq x < 5$
2	$5 \leq x < 9$
1	$x \geq 9$

Fig. 6 presents the average Q-score of each video segment with four resolutions. It is evident that in the slow motion scenes (video segments 1–14 and 27–39), the Q-scores corresponding to 1080p and 720p resolution fall between 5 and 4.9 points, whereas in the fast motion scenes (video segments 15–26, 40–52, and 53–64), the Q-score corresponding to the 720p resolution is obviously lower. However, the Q-score corresponding to a 1080p resolution exhibits no obvious change. Conversely, the Q-scores corresponding to the 480p and 360p resolutions were significantly higher in the slow motion scenes than in the fast motion scenes. Additionally, the difference in the Q-scores in the fast motion scenes was greater than that in the slow motion scenes. In other words, the effect is the same as that in the click distribution from the CD experimental results.

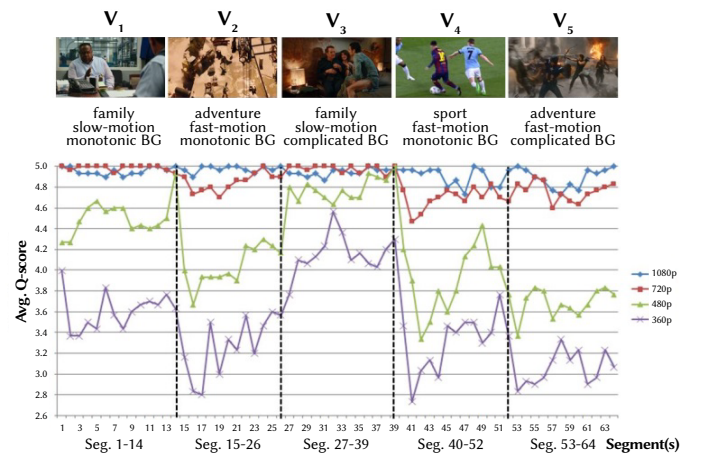


Fig. 6. Average Q-score distribution applied in our study.

E. Q-score Prediction By Variation in the Motion Vector

Conducting a CD experiment to obtain the user's Q-score before scheduling the playback of each video is impractical. Therefore, we used the exhaustive search method in the block-matching algorithm to perform motion vector calculations for each macroblock in a video

frame. We performed calculations for different resolutions of the motion vectors. Each video segment was 2 s long and the video was played at 30 fps, thus a video segment generated 60 frames. The length of the video segments was 64, and there were 3840 frames in total. Because our objective was to calculate the motion vector variation between frames, we compared the current frame with the frame that was two frames away from the current frame. For example, the 1st frame was compared with the 3rd frame, the 2nd frame was compared with the 4th frame, and so on (as presented in Fig. 7).

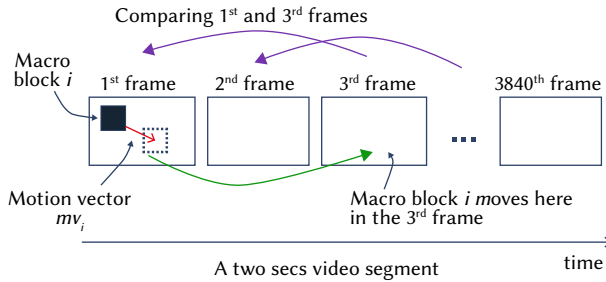


Fig. 7. Calculation of the motion vector, mv_i , between two frames.

After calculating all the motion vectors of the current frame, we averaged the vector length $|mv_i|$, for the current frame. The average motion vector variation is calculated using (1):

$$\frac{\sum_{i=1}^M |mv_i|}{M} \quad (1)$$

where M indicates the number of macro blocks in a video frame.

By comparing the corresponding frames in sequence, we obtained an average motion vector length of 3838 frames at four different resolutions. Fig. 8 compares the average motion variations of frames with two different resolutions, and the former resolution (indicated by “r1”) and the latter resolution (indicated by “r2”). Taking the comparison of the 1080p and 720p resolutions as an example, the number of larger average motion vectors of frames at 1080p resolution is 3643 more than that at 720p resolution (represented as “r1 > r2”), which accounts for 95% of the total. Conversely, the number of smaller average motion vectors of the frames in the 1080p resolution video is 195 less than that in the 720p resolution video (represented as “r1 < r2”), accounting for 5% of the total. In Fig. 8, we demonstrate that the average motion vector length in high-resolution video is always larger than that in low-resolution video. Thus, the average motion vector length of the video segments can be used to represent the video quality among videos with different resolutions.

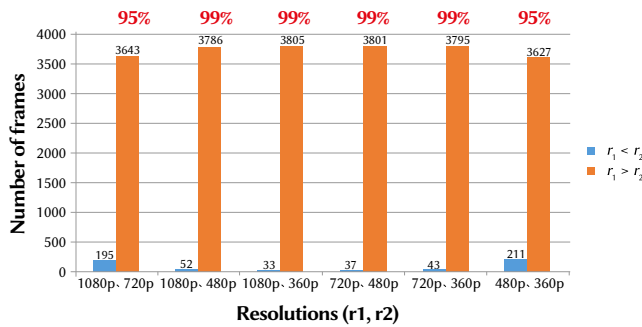


Fig. 8. Comparison of the average motion vector length of corresponding frames between two different resolutions.

Fig. 9 shows the average motion vectors of video segments with the four different resolutions. The frames were assigned to the video segments as follows: frames 1–60 constituted the first video segment,

frames 61–120 were the second video segment, and the final frames (frames 3781–3838) formed the 64th video segment, which had only 58 frames. Therefore, Fig. 9 also shows that the higher the resolution, the higher the average motion vector length regardless of whether the scene contained fast or slow motion actions.

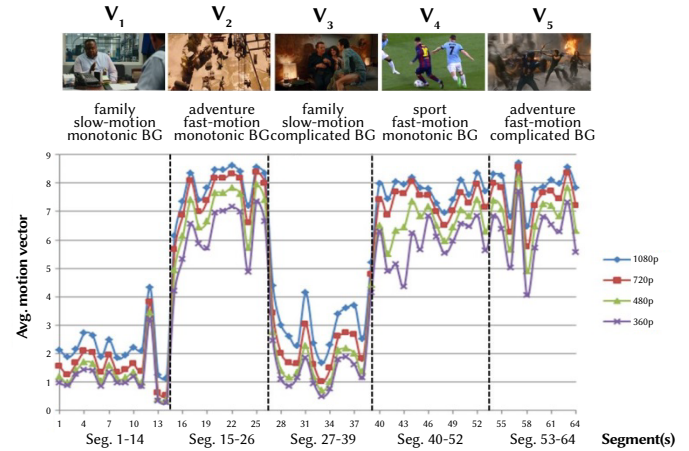


Fig. 9. Average length of motion vectors of video segments with different resolutions.

Furthermore, to evaluate whether the average motion vector length of video segments with four different resolutions was significant for the Q-score prediction model, we performed multiple regression and the Q-score was quantified by conducting our proposed CD experiment. Tables V–VIII contain the results of the ANOVA analysis of the motion vector and Q-score. Because multiple regression equations are established with different influencing factors, they are either highly or poorly correlated. In addition to the Q-scores corresponding to 1080p resolution, the others are significant for the Q-score prediction model. However, the p-values between groups indicate no significant difference, which means that the influencing factors may affect each other and cause the model to inaccurately predict the Q-score. Thus, we excluded the lowest relevant influencing factors and performed regression analysis of the other influencing factors until all of the influencing factors exhibited significant differences.

TABLE V. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH FOUR RESOLUTIONS AND Q-SCORE FOR 1080P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	4	0.013	0.003		
Residual	59	0.294	0.005	0.631	0.643
Total	63	0.307	-		

	Coefficient	Standard error	t-statistic	p-value
Intercept	4.964	0.052	94.883	3.3×10^{-66}
1080p avg. mv	-0.028	0.075	-0.369	0.714
720p avg. mv	0.063	0.131	0.48	0.633
480p avg. mv	-0.038	0.111	-0.343	0.733
360p avg. mv	-0.005	0.048	-0.103	0.919

TABLE VI. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH FOUR RESOLUTIONS AND Q-SCORE FOR 720P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	4	0.817	0.204		
Residual	59	0.4	0.007	30.15	1.14×10^{-13}
Total	63	1.217	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	4.894	0.061	80.237	6.05×10^{-62}	
1080p avg. mv	0.32	0.088	3.644	5.69×10^{-4}	
720p avg. mv	-0.7	0.152	-4.594	2.34×10^{-5}	
480p avg. mv	0.473	0.129	3.664	5.33×10^{-4}	
360p avg. mv	-0.103	0.056	-1.826	7.29×10^{-2}	

TABLE VII. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH FOUR RESOLUTIONS AND Q-SCORE FOR 480P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	4	8.983	2.246		
Residual	59	3.851	0.065	34.402	8.21×10^{-5}
Total	63	12.834	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	4.473	0.189	23.625	9.11×10^{-32}	
1080p avg. mv	0.644	0.272	2.366	2.13×10^{-2}	
720p avg. mv	-1.109	0.473	-2.346	0.022	
480p avg. mv	0.194	0.401	0.484	0.63	
360p avg. mv	0.219	0.174	1.254	0.215	

TABLE VIII. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH FOUR RESOLUTIONS AND Q-SCORE FOR 360P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	4	6.598	1.649		
Residual	59	5.685	0.096	17.118	2.27×10^{-9}
Total	63	12.283	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	3.562	0.23	15.485	1.9×10^{-22}	
1080p avg. mv	0.836	0.331	2.528	0.014	
720p avg. mv	-1.497	0.574	-2.608	0.012	
480p avg. mv	0.648	0.487	1.331	0.189	
360p avg. mv	-0.018	0.212	-0.086	0.932	

The Q-score model prediction for the 1080p resolution is of little relevance to the other influencing factors, and the Q-score for 1080p resolution was always greater than 4.7 points; in other words, most users were satisfied with the 1080p resolution. Therefore, we directly set the Q-score prediction model for 1080p resolution to a maximum of five points. Tables IX–XI present the results of the ANOVA analysis for the other three resolutions and their Q-scores, and all of the influencing factors have significant differences at the $\alpha = 0.05$ level. The coefficients of determination R^2 for the other three resolutions were 0.653, 0.699, and 0.537 for the 720p, 480p, and 360p resolutions, respectively.

TABLE IX. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH THREE RESOLUTIONS AND Q-SCORE FOR 720P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	3	0.795	0.265		
Residual	60	0.422	0.007	37.63	8.3×10^{-14}
Total	63	1.217	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	4.917	0.061	80.679	6.8×10^{-63}	
1080p avg. mv	0.261	0.083	3.138	2.63×10^{-3}	
720p avg. mv	-0.538	0.126	-4.256	7.41×10^{-5}	
480p avg. mv	0.268	0.065	4.135	1.12×10^{-4}	

TABLE X. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH THREE RESOLUTIONS AND Q-SCORE FOR 480P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	3	8.968	2.989		
Residual	60	3.867	0.064	46.384	1.23×10^{-15}
Total	63	12.834	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	4.504	0.177	25.431	7.89×10^{-34}	
1080p avg. mv	0.565	0.217	2.609	1.14×10^{-2}	
720p avg. mv	-0.914	0.244	-3.741	4.13×10^{-4}	
360p avg. mv	0.292	0.085	3.424	1.12×10^{-3}	

TABLE XI. RESULTS OF ANOVA ANALYSIS OF MOTION VECTORS WITH THREE RESOLUTIONS AND Q-SCORE FOR 360P RESOLUTION

ANOVA analysis	Degree of freedom (DF)	Sum of squares (SS)	Mean of sum (MS)	F	Significance F
Regression	3	6.597	2.199		
Residual	60	5.686	0.095	23.206	4.28×10^{-10}
Total	63	12.283	-		
	Coefficient	Standard error	t-statistic	p-value	
Intercept	3.566	0.224	15.948	3×10^{-23}	
1080p avg. mv	0.826	0.306	2.702	8.95×10^{-3}	
720p avg. mv	-1.469	0.464	-3.165	2.44×10^{-3}	
480p avg. mv	0.611	0.238	2.573	1.26×10^{-2}	

In the regression model, the predicted Q-score is y , the influencing factors corresponding to 1080p, 720p, 480p, and 360p resolutions are x_1 , x_2 , x_3 , and x_4 , respectively, and α_i is the coefficient of the influencing factor. The regression formula is presented in (2):

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 \quad (2)$$

The results led us to conclude that the variation in the average motion vector is higher, regardless of whether the scenes were contained in fast or slow motion segments. To evaluate whether the average motion vector variation of video segments with four different resolutions was significant for the Q-score prediction model, we performed multiple regression and ANOVA analyses. In the regression model, the predicted Q-score of the video segment was y . The calculated average motion vector variations in the 1080p, 720p, 480p,

and 360p resolutions are x_1 , x_2 , x_3 , and x_4 , respectively. We obtained regression equations to predict the Q-score of a video segment with different resolutions (as presented in Table XII).

TABLE XII. EQUATIONS FOR Q-SCORE PREDICTION WITH FOUR RESOLUTIONS

Resolution	Q-score regression equation (y)
1080p	$y=5$
720p	$y=4.917+0.261x_1-0.538x_2+0.268x_3$
480p	$y=4.504+0.565x_1-0.914x_2+0.292x_4$
360p	$y=3.566+0.826x_1-1.469x_2+0.611x_3$

Fig. 10 shows the multiple regression results for the Q-score prediction model with four resolutions. In the slow-motion scenes, the Q-score differences between resolutions were small, whereas they were larger in the fast motion scenes. This phenomenon is the same as that in the original Q-score distribution obtained from the CD experiment. Therefore, we also used the correlation coefficient to analyze the difference between the predicted and original Q-scores. Fig. 10 shows the multiple regression results for the Q-score prediction model with four resolutions, which are similar to the results presented in Fig. 6.

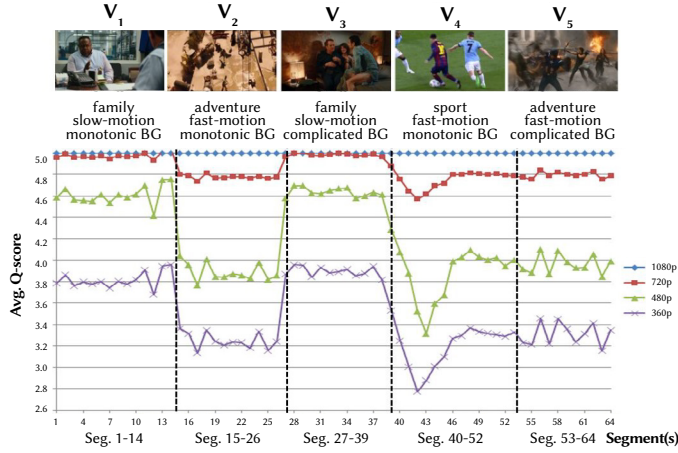
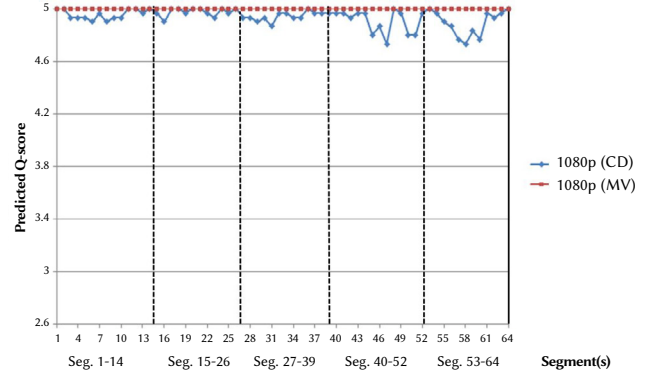


Fig. 10. Average Q-score distribution based on multiple regression analysis.

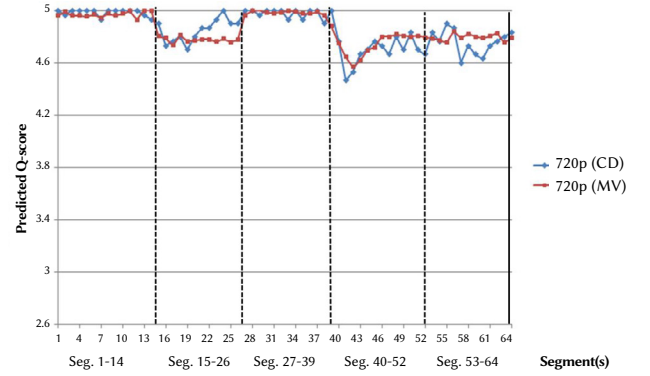
Fig. 11 shows the predicted Q-scores with the click density (CD) and motion vector (MV) at four resolutions. Because the Q-score predictions from motion vectors in the video with 1080p resolution always equal 5 points, the correlation calculation could not be applied. However, Fig. 11(a) shows that the Q-score traces between these two methods are similar. Fig. 11(b), Fig. 11(c), and Fig. 11(d) present the results for the 720p (correlation = 0.812), 480p (correlation = 0.836), and 360p (correlation = 0.733) resolutions, respectively. From these results, we conclude that the predicted Q-score from the motion vector method is close to the true user QoE.

F. Integer Programming Algorithm

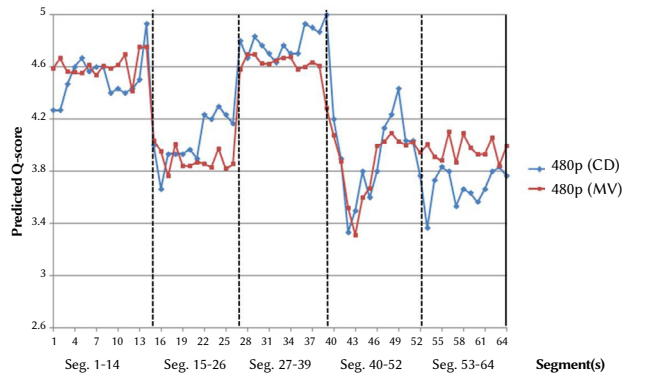
The differences in the Q-scores between the different resolutions in the slow-motion scenes calculated with the proposed Q-score prediction model were small, but were larger for the fast motion scenes. This finding is the same as that for the original Q-score distribution obtained from the CD experiment. To schedule video playback in which high-resolution video segments are assigned to fast motion scenes and low-resolution video segments to slow motion scenes, we applied integer programming to maximize the overall average Q-score gain [10]. The playback schedule T , is a combination of N video segments. We present this playback scheduling, T , in (3). Each video segment selects only one resolution among K resolutions.



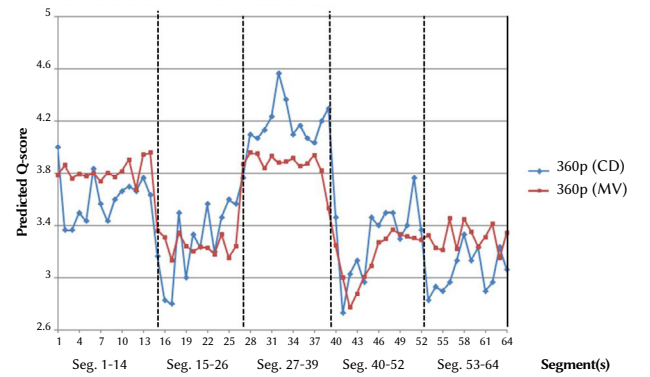
(a) Video resolution 1080p



(b) Video Resolution 720p



(c) Video resolution 480p



(d) Video resolution 360p

Fig. 11. Comparison of predicted Q-scores calculated with the click density (CD) and motion vector (MV) methods for four different video resolutions.

The $\lambda_{i,j}$ indicates that video segment i selects the j^{th} resolution and has a choice of 0 or 1.

$$T = \begin{bmatrix} \lambda_{1,1} & \lambda_{2,1} & \dots & \lambda_{N,1} \\ \lambda_{1,2} & \lambda_{2,2} & \dots & \lambda_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{1,K} & \lambda_{2,K} & \dots & \lambda_{N,K} \end{bmatrix} \quad (3)$$

If the j^{th} resolution of segment i is chosen, the Q-score gain, $q_{i,j}$, is accumulated in $Q(T)$. Because the objective is to maximize the average Q-score gain, $Q(T)$, the target function is expressed by (4):

$$\text{Maximize } Q(T) = \sum_{i=1}^N \sum_{j=1}^K q_{i,j} \times \lambda_{i,j} \quad (4)$$

When each video segment is selected, there are K resolutions to choose from, and each resolution has its own Q-score. However, under different network bandwidth conditions, R , the selection is subject to two restrictions:

$$\sum_{j=1}^K \lambda_{i,j} = 1, 1 \leq i \leq N \quad (5)$$

$$R \times (L + (i - 1) \times \Delta T) - \sum_{x=1}^i |s_x| \geq 0, 1 \leq i \leq N \quad (6)$$

Equation (5) is the first restriction when playing a video, and only one resolution is selected for each video segment, $\sum_{j=1}^K \lambda_{i,j} = 1$. Equation (6) presents the second restriction when the resolution suitable for video playback is selected in the video segment. Additionally, the entire amount of data transmitted, $\sum_{x=1}^i |s_x|$, are prevented from exceeding the total data size the given network bandwidth is able to transmit, $R \times (L + (i - 1) \times \Delta T)$. We used the integer programming method and proposed the OSA algorithm to determine the best solution for video playback with a video length of 64 segments. With limited network bandwidth, the OSA algorithm can be used to optimize video playback.

IV. EXPERIMENTAL RESULTS

To compare the video playback schedules that were edited using the different algorithms, we set up a video test platform to provide subjects with video satisfaction tests. BL is a popular scheduling algorithm for MPEG-DASH streaming applications [7]. Therefore, we applied this BL algorithm to test the effectiveness of our proposed OSA algorithm to improve the QoE of the user. The steps of the experiment were as follows. Two test videos were used: the first was scheduled using the OSA algorithm and the other using the BL algorithm.

Although the two videos contained the same movie scenes, the video segments of different resolutions were arranged in different ways in the video playback. Subjects who were participating in the experiment viewed the two videos in randomly generated order to prevent the subject from having an established impression. After the first video had finished playing, to prevent the subject from forgetting the current feeling of satisfaction with the video, we asked the user to provide a rating regarding the feeling. Moreover, the rating was referenced by ACR, the five-level rating scale for different QoE levels, and the second video was shown using the same process as before. Finally, once each subject had completed the two different video tests, we asked the subject to again indicate which one they were more satisfied with, to confirm the consistency of the answers. There were three options to choose from: the first video, the second video, or no difference. The user satisfaction tests were conducted with four different resolutions and with playback bitrates of 3000 (1080p), 1500 (720p), 500 (480p), and 200 (360p) kbps. The playback duration of each video segment was 2 s, and the total number of segments in each video was 64. Therefore, the video length was 2 min 8 s, and the videos did not include sounds or subtitles. For a fair comparison, the startup

latency for running each algorithm was set to 2 s with a constant network bandwidth, and none of the videos froze. To distinguish which QoE metrics applied to the playback schedule, OSA_C indicates that the OSA algorithm applied QoE metrics from click density (CD) experiments and OSA_M represents that the OSA algorithm applied QoE metrics from the motion vector (MV) method.

A. Comparison of OSA_C and BL Algorithms

In this section, we compare the user satisfaction results of the BL algorithm with a 30s buffer with those obtained with the OSA_C algorithm. Furthermore, the results obtained with a network bandwidth of 900 kbps are presented. The experimental results for the different videos are shown in Fig. 12. We demonstrate that, among the two algorithms, the OSA_C algorithm has the largest amount of accumulated data and an average bitrate of 896.875 kbps, followed by the BL algorithm with a 30s buffer and an average bitrate of 628.125 kbps. The OSA_C algorithm processes the playback schedule based on the QoE metrics from the click density (CD) experiments. The results in Fig. 12 show that the OSA_C algorithm allocates high-resolution video segments starting from segments 15 and 46, because the gap in user satisfaction becomes wider. Therefore, the amount of accumulated data increased sharply between video segments 15 and 46. Conversely, the BL algorithm processes the resolution adaption only on the buffer occupation, and it does not adapt the video resolution in these time slots.

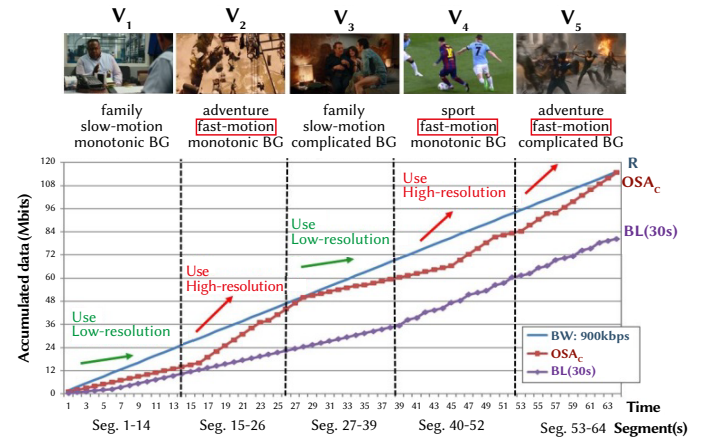


Fig. 12. Two playback schedules processed by the OSA_C and BL algorithms with a network bandwidth constraint of 900 kbps.

The data presented in Fig. 13 were obtained from 42 test subjects. The distributions can be divided into three parts: above the line, below the line, and on the line. The line represents the score at which the results for the OSA_C and BL algorithms were the same. The distributions above and below the line correspond to subjects who preferred the video playback schedules edited by either the BL or the OSA_C algorithm, respectively. User satisfaction was rated from 1 to 5 points, and the different colors represent the situations the subjects selected when reconfirming their satisfaction status. Finally, the size of each pie chart represents the number of people who chose the score in that case. For example, in the case of the largest pie chart, which represents eight subjects, five subjects chose the BL algorithm with two points and the OSA_C algorithm with three points. Upon reconfirmation, this result indicated that user satisfaction with the OSA_C algorithm was higher than that with the BL algorithm. The three different situations represented in the bar chart in Fig. 14 correspond to the number of subjects who assigned higher scores to either the OSA_C or BL algorithm or assigned the same score to both. Additionally, the subjects chose which playback schedule they preferred. Eighteen users scored the schedule derived using the OSA_C algorithm higher than

that produced by the BL algorithm, and they considered the former algorithm to yield superior results. The experimental results shown in Fig. 14 indicate that most of the subjects felt that the video playback scheduled using the OSA_c algorithm provided a more satisfactory visual experience.

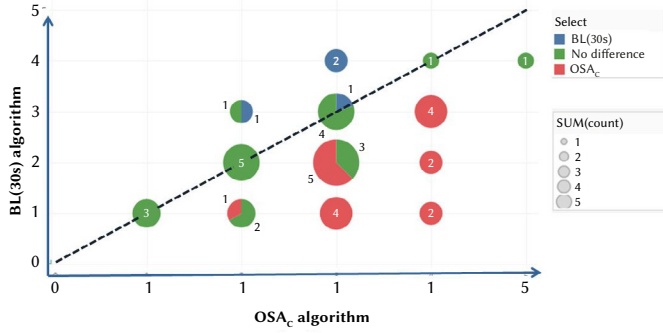


Fig. 13. Pie chart of user satisfaction with the OSA_c and BL algorithms (buffer = 30s).

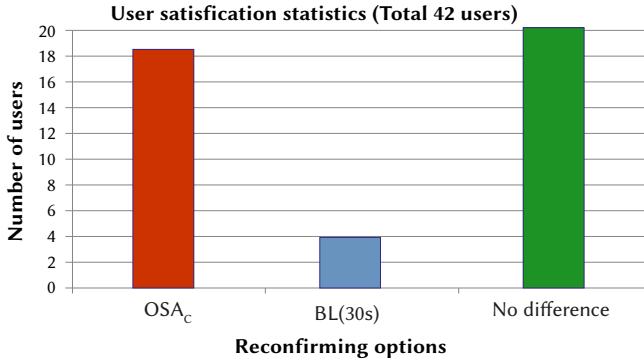


Fig. 14. User satisfaction statistics for the OSA_c and BL algorithms (buffer = 30s).

Fig. 14 shows that the satisfaction counts obtained by applying the OSA_c algorithm are higher than those obtained by applying the BL algorithm with a 30-s buffer. To confirm whether the two different video playbacks were distinct, we conducted a paired sample t-test to analyze the significance of the satisfaction scores. The null hypothesis was that the effect of the two algorithms was insignificant. Table XIII demonstrates that the differences between the experiments were significant at the $\alpha = 0.01$ level. The p-values from the first and second experiments were both less than 0.01, refuting the null hypothesis. Thus, the video playback scheduled using the OSA_c algorithm provides a more satisfactory user experience.

TABLE XIII. T-TEST OF SATISFACTION SCORE: OSA_c Vs. BL (BUFFER = 30S)

Algorithm	Sum	Average	Variance	p-value
OSA_c	121	2.88	0.8391	3.39×10^{-5}
BL	91	2.17	0.9228	

B. Comparison of OSA_M and BL Algorithms

The experimental results presented in Section IV-A show that video playback scheduled with QoE metrics from the CD experiment delivers higher streaming performance. In this section, we applied the QoE metrics obtained from the motion vector (MV) method and arranged a playback schedule using the OSA_M algorithm. We then performed video playback scheduling using the OSA_M algorithm and the original approach using the BL algorithm with a 30-s buffer to determine which result is more appealing to users. Therefore, we chose a network bandwidth of 900 kbps, where the video playback was

composed of high-resolution and low-resolution video segments. Fig. 15 depicts the results of the two different video playback schedules with a network bandwidth of 900 kbps. The average bitrate of the OSA_M algorithm is 875 kbps, which is slightly lower than that of the OSA_c algorithm but much higher than that of the BL algorithm with a 30-s buffer. Additionally, the OSA_M algorithm uses high-resolution video segments starting from video segments 15 and 46 because of the fast motion scenes.

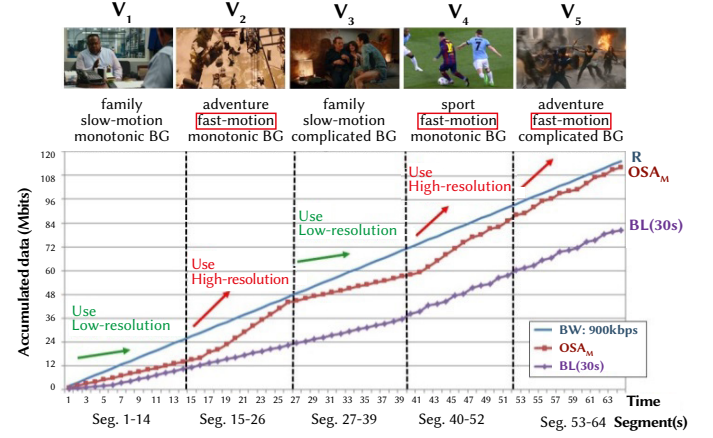


Fig. 15. Two playback schedules handled by OSA_M and BL algorithms with a network bandwidth of 900 kbps.

The user feedback, shown in Fig. 16, is that of 30 participants who participated in the test. The experimental results indicate that most users rated the OSA algorithm higher than the BL algorithm with a 30-s buffer. Moreover, the larger pie charts are mostly below the line, which means that the subjects were more satisfied with the OSA algorithm. Fig. 17 demonstrates that more than half of the subjects tended to choose the OSA algorithm instead of the BL algorithm with a 30-s buffer. Thus, with a network bandwidth of 900 kbps, the QoE gain scheduled using the OSA algorithm was significantly more satisfactory to the test users.

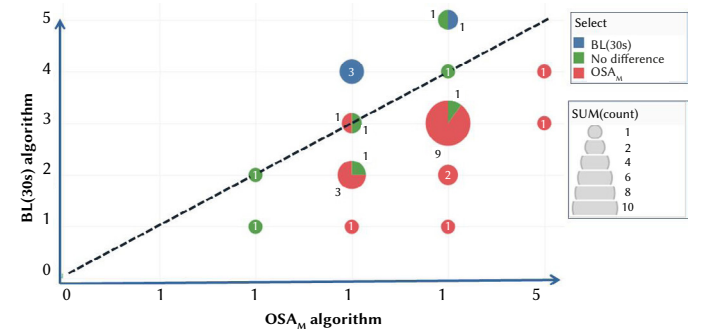


Fig. 16. Pie chart of user satisfaction with the OSA_M and BL algorithms (buffer = 30s).

Because of the mechanism of the BL algorithm, it always chooses low-resolution video segments at the beginning, and when the buffer occupancy is sufficient, the BL algorithm switches to high-resolution video segments. In contrast, the OSA_M algorithm always uses high-resolution video segments for video streaming. To confirm whether the two different video playbacks were distinct, we conducted a paired sample t-test to analyze the significance of the satisfaction scores. Fig. 17 shows that the satisfaction counts obtained by applying the OSA_M algorithm are higher than those obtained by applying the BL algorithm with 30-s buffering. The null hypothesis was that the effect of the two algorithms was insignificant. The results in Table XIV demonstrate

that both experiments have significant differences at the $\alpha = 0.01$ level. The p-values from the first and second experiments were both less than 0.01, refuting the null hypothesis. Thus, the video playback scheduled using the OSA_M algorithm provides a more satisfactory user experience.

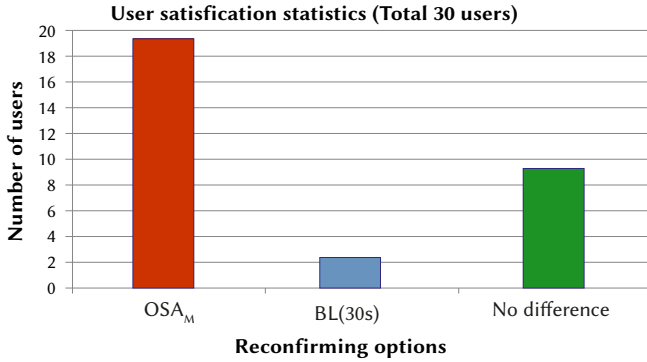


Fig. 17. User satisfaction statistics for the OSA_M and BL algorithms (buffer = 30s).

TABLE XIV. T-TEST OF SATISFACTION SCORE: OSA_M Vs. BL (BUFFER = 30S)

Algorithm	Sum	Average	Variance	p-value
OSA_M	108	3.6	0.5241	4.48×10^{-4}
BL	86	2.87	1.085	

V. CONCLUSIONS AND FUTURE WORK

DASH streaming is a popular method for video streaming over the Internet. The QoE score measures the extent to which the user is satisfied with the viewing experience but is difficult to evaluate, and the factors influencing the QoE, including the user's subjective feeling and sound environment, are complicated. The current DASH streaming scheduling algorithms only consider the network constraints, and the primary objective of these algorithms is to provide stable video playback, instead of focusing on the QoE. To evaluate the factors influencing the QoE, we designed a series of click density (CD) experiments to collect unsatisfactory click counts in different scenes with different video resolutions. The click distributions in the CD experiment indicated that the test subjects were usually unsatisfied with fast motion scenes and relatively satisfied with slow motion scenes at the same video resolution. In other words, the use of high-resolution video segments in high-motion scenes would significantly improve the QoE. Additionally, we observed that the difference between the two levels of resolution was greater in high-motion scenes than in low-motion scenes. Therefore, we defined an ACR five-level Q-score for rating the quality of different QoE levels, applied integer programming, and proposed the OSA algorithm to generate video playback schedules that maximized the Q-score gain with network bandwidth constraints. Then, we designed a subjective experiment to test user satisfaction with our proposed OSA algorithm and the most popular DASH streaming algorithm, the BL algorithm. In addition to converting clicks into Q-scores, we applied multiple regression to derive the correlation of the motion vector variations with different resolutions with the Q-scores. The experimental results show that the playback videos generated using the OSA algorithm with the motion vector model also increased user satisfaction. In this study, we only used fast or slow motion to investigate user satisfaction during different scenes of different video resolutions, where motion speed is an influencing factor. Thus, we aim to incorporate other influencing factors and modify our proposed Q-score prediction algorithms for video playback scheduling in the near future.

REFERENCES

- [1] A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, 2010.
- [2] I. Sodagar, "MPEG-DASH standard for multimedia streaming over the internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, 2011.
- [3] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP - Standards and design principles," in *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems*, pp. 133–144, 2011.
- [4] Pantos and May, Apple, Inc., "HTTP live streaming," IETF RFC 8216, July, 2017. [online] Available: <http://tools.ietf.org/html/draft-pantos-http-live-streaming-23>
- [5] A. M. Tekalp, "Digital Video Processing," 2nd edition, Prentice Hall Press Upper Saddle River, NJ, USA, 2015.
- [6] U. Reiter, K. Brunnström, K. D. Moor, M. C. Larabi, M. Pereira, A. Pinheiro, J. You, and A. Zgank, "Factors influencing quality of experience," *Quality of Experience, Advanced Concepts, Applications and Methods*, pp. 55–72, 2014.
- [7] C. Müller, S. Lederer and C. Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments," in *Proceedings of the 4th ACM International Multimedia Conference on Mobile Video*, pp. 37–42, 2012.
- [8] A. Zambelli, "IIS smooth streaming technical overview," *Microsoft documentation, Microsoft Inc.*, vol. 3, 2009.
- [9] M. Levkov, "Video encoding and transcoding recommendations for HTTP dynamic streaming on the Adobe Flash platform," *White Paper, Adobe Systems, Inc.*, 2010.
- [10] A. Schrijver, "Theory of Linear and Integer Programming," *John Wiley & Sons Inc.*, ISBN 978-0-471-90854-8, 1986.
- [11] R. Sakamoto, T. Shobudani, R. Hotchi, and R. Kubo, "QoE-Aware Stable Adaptive Video Streaming Using Proportional Derivative Controller for MPEG-DASH," *IEICE Transactions on Communications*, vol. E104.B no. 3 pp. 286–294, 2021.
- [12] C. Cetinkaya, K. Herguner, C. Hellge, and M. Sayit, "Segment-aware dynamic routing for DASH flows over software-defined networks," *International Journal of Network Management*, vol. 30, issue 4, 2020.
- [13] Y. Gao, X. Wei, L. Zhou, "Personalized QoE improvement for networking video service," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2311–2323, 2020.
- [14] Bentaleb, Abdelhak, A. C. Begen, and R. Zimmermann. "SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking," in *Proceedings of the 24th ACM International Conference on Multimedia*, pp.1296–1305, 2016.
- [15] C. L. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard, "QoE-driven mobile edge caching placement for adaptive video streaming," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 965–984, 2017.
- [16] S. Zhao, Z. Li, D. Medhi, P. L. Lai, and S. Liu, "Study of user QoE improvement for dynamic adaptive streaming over HTTP (MPEG-DASH)," in *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)*, pp. 566–570, 2017.
- [17] S. H. Lee, E. Lee, and H. W. Lee, "Quality adaptation scheme for improving QoE of MPEG DASH," in *Proceedings of IEEE International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 368–370, 2016.
- [18] Y. M. Cao, X. Q. You, J. Wang, and L. Song, "A QoE Friendly Rate Adaption Method for DASH," in *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1–6, 2014.
- [19] I. R. Alzahrani, N. Ramzan, S. Katsigiannis, and A. Amira, "Use of Machine Learning for Rate Adaptation in MPEG-DASH for Quality of Experience Improvement," in *Proceedings of the 5th International Symposium on Data Mining Applications*, pp. 3–11, 2018.
- [20] W. W. Huang, Y. P. Zhou, X. Y. Xie, D. Wu, M. Chen, and E. Ngai, "Buffer State is Enough: Simplifying the Design of QoE-Aware HTTP Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 590–601, 2018.
- [21] X. Q. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proceedings of the ACM SIGCOMM*, pp. 325–338, 2015.
- [22] W. Xin, P. Chuan, L. Zhou and Y. Qian, "QoE Oriented Chunk Scheduling

in P2P-VoD Streaming System,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8012–8025, 2019.

- [23] K. T. Chen, C. C. Tu, and W. C. Xiao, “OneClick: A framework for measuring network quality of experience,” in *Proceedings of IEEE INFOCOM*, pp. 702–710, 2009.
- [24] T. Yue, H. Wang, S. Chen, and J. Shao, “Deep learning based QoE evaluation for internet video,” *Neurocomputing*, vol. 386, pp.179–190, 2020.
- [25] G. Dimopoulos, I. Leontiadis, P. B. Ros, and K. Papagiannaki, “Measuring video QoE from encrypted traffic,” in *Proceedings of the Internet Measurement Conference*, pp. 513–526, 2016.
- [26] T. Zhao, Q. Liu, and C. W. Chen, “QoE in video transmission: A user experience-driven strategy,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285–302, 2017.
- [27] U. Engelke, D. P. Darcy, G. H. Mulliken, S. Bosse, M. G. Martini, S. Arndt, J. N. Antons, K. Y. Chan, N. Ramzan, and K. Brunnstro, “Psychophysiology-based QoE assessment: A survey,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 6–21, 2016.
- [28] S. H. Hu, L. F. Sun, C. X. Xiao, and C. Gui, “Semantic-aware adaptation scheme for soccer video over MPEG-DASH,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 493–498, 2017.
- [29] A. Barjatya, “Block matching algorithms for motion estimation,” *IEEE Transactions on Evolution Computation*, vol. 8, pp. 225–239, 2004.
- [30] Hosur, P. Irappa, and K. K. Ma. “Motion vector field adaptive fast motion estimation,” in *Proceedings of the 2nd International Conference on Information, Communications and Signal Processing (ICICS’99)*, pp. 7–10, 1999.
- [31] A. M. Tourapis, O. C. L. Au, and M. L. Liou, “Predictive motion vector field adaptive search technique (PMVFAST): enhancing block-based motion estimation,” *Visual Communications and Image Processing*, vol. 4310, pp. 883–892, 2001.
- [32] S. Arora, K. Khanna, and N. Rajpal, “A Novel Hybrid Approach for Fast Block Based Motion Estimation,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 6, pp. 24–30, 2017.
- [33] S. Kamble, N. Thakur, A. Samdurkar, and A. Patharkar, “Object Detection and Tracking using Modified Diamond Search Block Matching Motion Estimation Algorithm,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 1, pp. 73–85, 2018.
- [34] D. K. Krishnappa, D. Bhat, and M. Zink, “DASHing YouTube: An analysis of using DASH in YouTube video service,” in *Proceedings of the 38th Annual IEEE Conference on Local Computer Networks*, pp. 407–415, 2013.
- [35] YouTube genres and categories. [online] Available:https://developers.google.com/youtube/v3/live/guides/encoding-with-dash



Shin-Hung Chang

Shin-Hung Chang received his Ph.D. and M.S. degrees in Computer Science and Information Engineering from National Taiwan University in 2005 and 1998, respectively. He received his B.S. degree in Computer Science and Information Engineering from Fu Jen Catholic University in 1996. He joined the Institute of Information Science (IIS), Academia Sinica as a research assistant from 1998 to

2005 and as a post-doctoral fellow from 2005 to 2006. Dr. Chang is responsible for patent matters in the Computer System and Communication Laboratory (CSCL) of the IIS, and he holds one United States patent and four Taiwanese patents in computer screen recording technology. In the IIS, Dr. Chang was also responsible for transferring several technologies to industry. After his postdoctoral work, Dr. Chang joined an American company, MagnetoX Co. Ltd., as an R&D team manager. He led several multimedia application development projects and handled patent matters from 2006 to 2007. Dr. Chang joined Fu Jen Catholic University as a professor in 2007. Dr. Chang coached students to attend many ICPC programming contests and hosted several Ministry of Science and Technology (MOST) research projects during his teaching period at Fu Jen Catholic University. Additionally, Dr. Chang is an IEEE member and IEEE computer society member. Dr. Chang’s research interests cover the integration of theory and application, including machine-learning algorithms, network protocol design, multimedia applications, media streaming, media codec algorithms, slot machine game model construction, and real-time strategy (RTS) game AI.



Min-Lun Tsai

Min-Lun Tsai received his M.S. and B.S. degrees in Computer Science and Information Engineering from Fu Jen Catholic University in 2018 and 2016, respectively. Mr. Tsai joined the Institute of Information Science (IIS), Academia Sinica, as a research intern in 2017. After earning his M.S. degree, he accepted a position as software engineer at ACTi Corporation, where he is currently employed, and develops IP camera applications, such as intelligent retail solutions, digital signage with age/gender recognition, and window applications. Mr. Tsai’s research interests cover media streaming, multimedia applications, stock analysis, app development, and recognition algorithms.



Meng-Huang Lee

Meng-Huang Lee received his B.S. and M.S. degrees in Electrical Engineering from National Cheng Kung University in 1987 and 1989, respectively, and his Ph.D. degree in Computer Science and Information Engineering from National Taiwan University in 1996. He is currently a professor at the Department of Information Technology and Management, Shih Chien University. His research interests include multimedia systems, IPTV, and computer networks.



Jan-Ming Ho

Jan-Ming Ho received his Ph.D. degree in electrical engineering and computer science from Northwestern University in 1989. He received his M.S. degree at the Institute of Electronics of National Chiao Tung University in 1980 and his B.S. degree in electrical engineering from National Cheng Kung University in 1978. Dr. Ho joined the Institute of Information Science, Academia Sinica, as an Associate Research Fellow in 1989, and was promoted to Research Fellow in 1994. During the years 2000-2003, he served as the Deputy Director of the institute. In 2004-2006, he served as Deputy Director of IIS, Academia Sinica in 2000-2003, and as Director General of the Division of Planning and Evaluation, National Science Council. During his term as Deputy Director of the IIS, he was responsible for the planning and operation of the National Digital Archive Program, organized by the Summer Institute on Bioinformatics in 2001. He also founded the Open Source Foundry and organized a series of events promoting open sources. He co-founded Foresight Taiwan and the Germination Initiative and Functional Units with Dr. Eugene Wong in 2007-2012. He served as a reviewer and think tank member of the Germination Initiative ever since. He also served as an advisor to the Ministry of Education and its Division of IT education in 2012-2016. He served as Chair of the Office of Science & Technology Program Executive Review Board in 2017-2018. Dr. Ho’s research interests cover the integration of theory and applications, including combinatorial optimization, information retrieval and extraction, multimedia network protocols, bioinformatics, and digital library and archive technologies. Dr. Ho also published results in the field of VLSI/CAD physical design.