Reducing Perceptual Discontinuity P2P Live Video Streaming Using Adaptive Scheme Based on Frame Loss Visibility

Ricky Christanto[1], Shaou-Gang Miaou[2], Ting-Lan Lin[3]

Chung Yuan Christian University, Taiwan
[1]richrist81@gmail.com, [2]miaou@cycu.edu.tw, [3]tinglan@cycu.edu.tw

Abstract: When a live streaming video is delivered over a peer-to-peer (P2P) network, the video is divided into several chunks and distributed to peers. Each chunk has its own playback time deadline. A chunk delay can be a problem of great concern because a serious chunk delay can produce obvious discontinuity of the streaming video, resulting in low satisfaction for a viewer. The playback time of a delayed chunk is overlapped with that of the next chunk. Traditionally, fast forwarding and uniform frame skipping are the two methods used to deal with the chunk delay problem. However, they may easily produce the effect of perceptual discontinuity and perceived by the viewer. This study proposes a frame dropping method based on frame loss visibility information in order to maintain visual continuity of the video. The frame with lower loss visibility will be dropped first. In a previous work, the idea of loss visibility was applied by a router to develop a frame dropping strategy when network congestion occurs. In this work, we apply it to the chunk-level playback controller in P2P network. The number of frames to be dropped is dependent on the value of chunk delay. Given a fixed chunk delay, the main difference between the proposed approach and the uniform frame skipping method is the way of choosing frames for dropping (fast forwarding method does not drop any frame but play all the frames at a faster speed). A viewer's satisfaction is evaluated by using the way of subjective video quality assessment. The proposed method can produce higher MOS (mean opinion score) than that of the traditional methods (fast forwarding and uniform frame skipping), demonstrating the effectiveness of the proposed approach.

Keywords: Video streaming, P2P networks, perceptual discontinuity, frame loss visibility.

1. **Introduction**

Delivering multimedia data over a peer-to-peer (P2P) network receives more and more demand. P2P downloads data files in a parallel manner from other peers rather than from a single data transfer node in the client server topology. The file transferred by a P2P system is divided into several chunks. The system distributes the chunks to the peers who are interested in the chunks. The peers are coordinated to achieve the goal of file exchange. In the P2P system, a seeder is known as a peer that already has the complete file and it can give the chunks of file to other peers. In contrast, a peer that still needs some chunks from other peers is called a leecher. Video chunk is not just used for local consumption (with playback time deadline), but also for uploading to other peers.

The network impairment to video playback in P2P streaming is rarely packet loss, but only delayed chunk. In this study, we propose a method based on data loss visibility to reduce the bad perceptual effect associated with the chunk delay problem. We propose a video playback design to improve the viewer's satisfaction in visual quality whenever the delay chunk problem occurs. We evaluate the video quality resulted from each method by conducting a subjective experiment. This research proposes a frame dropping method based on frame loss visibility information in order to maintain visual continuity of the video. In addition, this research also proposes an adaptive frame dropping system that performs the task at different locations in chunk file. The performance of this

work is compared with that of other methods such as fast forwarding and uniform frame skipping. The paper is organized as follows. We start with the review of relevant literature works, including the P2P network, video streaming, and perceptual discontinuity related to chunk level impairment. Then, we describe the fast forwarding, frame skipping, and the proposed method, including the description on the methodology to design the subjective tests. Next, we give the experimental results and discussion. Finally we make conclusions and point out possible future research directions.

2. Literature Review

A. *P2P Live Video Streaming*

Combining video streaming and P2P network as P2P video streaming becomes a demanding application in the future. The P2P approach divides the video content into chunks (or substreams); and it relies on the chunks travelling along different paths in order to achieve the desired streaming rate. The paths taken by different chunks may be constantly changing, due to the strategies used to balance the load on peers. In P2P systems, the most noticeable impairments are not bit-level distortions, nor packet losses (as they get repaired), but video chunk or frame level delays [1].

For the P2P systems that deliver 3L-videos (low-bit rate, low-frame rate and low-resolution video), the minimum processing unit is no longer a packet but a chunk. In P2P streaming applications, almost all important building blocks and components are designed based on chunks, e.g., chunk selection algorithm, peer selection algorithm (based on chunk bit-map information exchanged among neighbor peers), local chunk buffer management, and so on [1].

There are four important factors on chunk-level impairment systems [2]. The first factor is chunk size. It is a key parameter that determines the trade-off between chunk-level transmission, control overhead, and chunk-level playback delay. The second factor is chunk-level distortion. The third factor is playback controller, which will be the main focus of this paper. The fourth factor is average playback continuity.

Live video streaming has a desired playback time for real time video applications. Video on P2P is divided into several chunks of file and distributed to P2P networks. Each chunk of file has its own playback time. Figure 1 shows chunks of file with their desired playback time.

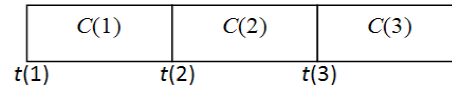| $C(1)$	$C(2)$	$C(3)$
$t(1)$	$t(2)$	$t(3)$

Figure1. Illustration of video chunks $C(i)$ along with desired playback time $t(i)$.

The playback time is defined in order to maintain the continuity of the video streaming on the network. Some research works discussed the issue of how to meet the desired playback time requirement [3-6]. It is an important factor for a user to watch video as a real-time application.

A.1. *Perceptual Discontinuity: Delayed Chunk*

Temporal quality is a terminology to describe the relation among frames inside the video. The length of the delay is an important factor that affects the perceptual discontinuity of the video. In general, the longer time of the delay, the more frames need to be dropped, and when more frames are dropped, the perceptual discontinuity of the video will be more serious. In this case, the viewer will feel that more frames on the video disappear and the continuity of the frames within the video becomes degraded. The content of video dominates the subjective viewing experiment results [7]. The two types of frame loss are regular and irregular. A motion mapping mechanism associated with scene cut detection has been studied in [8] as an objective temporal quality metric called *Perceptual Temporal Quality Metric* (*PTQM*). The detection and analysis of transition between shots of video sequences is investigated in order to know the content of video. The frame dropping effects on users' quality perception have been investigated [9].

Our method attempts to maintain the temporal quality of the video even when the system needs to drop a lot of frames in a video. With a subjective assessment based on the Mean Opinion Score (MOS), viewers are asked to determine the quality of the video.

The impact of P2P distribution on video quality is not completely understood yet, especially in live streaming situations. This work addresses the impact of P2P distribution when the delay of the play is out of the limit, as it must be in any true live TV service [10]. Chunk availability varies as a function of the number of users in the system [11]. The bandwidth heterogeneity among peers can be exploited to significantly improve the delay performance of all peers [12].

On P2P video streaming, each chunk $C(i)$ comes to the chunk-level playback at the user side from another peer with desired playback time $t(i)$ as shown in Figure1 for $i = 1, 2$ and 3. Figure 2 shows an example when a chunk has a delay, where the delay time d is given to simulate the chunk delay pattern on live video streaming.
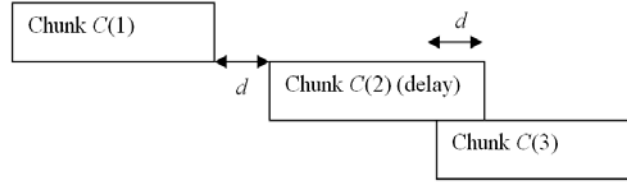

Figure 2. An example of chunk delay on P2P streaming network.

Between Chunk $C(1)$ and Chunk $C(2)$ the playback controller will use the last frame of Chunk $C(1)$ as a frozen frame until Chunk $C(2)$ is ready to play. Furthermore, the delay of Chunk $C(2)$ will create an overlapping area between Chunk $C(2)$ and Chunk $C(3)$. This delay is related to the arrival of chunk file on video streaming system over throughput–limited paths [8]. We focus on this type of delay rather than the startup delay on P2P streaming system that occurs in the beginning of the playback video because it occurs quite often in P2P systems for a long playback of video. At the desired playback time of Chunk $C(3)$, Chunk $C(2)$ still has some remaining frames to play until it is finished. In this case, the system needs to sacrifice some frames from Chunk $C(2)$ or Chunk $C(3)$ in order to keep the continuity of the video.

For a popular P2P live streaming channel, the player pop-up delay (the delay from the instant when one channel is selected until the streaming player pops up) is in general 10 to 15 seconds and the player buffering delay (the delay from the instant when the player pops up until the playback actually starts) is 10 to 15 seconds. Therefore, the total start-up delay is from 20 to 30 seconds. These delays are significantly longer than what are provided by traditional television broadcasting [13]. The delay of our concern in this paper is related to the arrival of chunk, which is around 0 to 4 seconds [1].

A.2. *Chunk-Level Playback Controller*

The size of a chunk is implementation dependent, but definitely much larger than one packet [1]. The chunk size is typically much larger than the maximum payload size of a TCP segment (typically 1460 bytes). For example, in PPLive [14], the chunk size is larger than 14 KBytes (the exact chunk size depends on the bitrate) [15]. The size of a chunk is about 2 Mbytes in [16]. The chunk size is a key parameter that determines the trade-offs between chunk-level transmission, control overhead, and chunk-level playback delay. There are two extreme cases: when a P2P system sets the smallest value of the chunk size (equals to the packet size), it becomes the traditional Client-Server mode streaming; on the other hand, when we set the largest vale (equals to the size of the whole video content), it is similar to the P2P file downloading [2]. Chunk size can have significant impact on diffusion rate (chunk miss ratio), diffusion delay, and overhead. P2P size does matter, with the optimal size being not too small. For large chunks (greater than a few hundred kilobits), there are no missing chunks. As the chunk size goes below a certain critical value, chunks start to miss, roughly proportional to the logarithm of the chunk size. The delay is also proportional to the chunk size [17]. Using smaller chunks in video streaming leads to smaller chunk transmission delay [18].

Some previous works on P2P focus more on how a playback controller can determine whether the player should wait (hence freeze) for additional content to arrive when (part of) a chunk has a delay or is missing [1]. However, a good and complete design should also consider how users react to different frame delay scenarios and playback decisions. To the best of our knowledge, no method was proposed to manipulate specific areas of the chunk, which may be necessarily to achieve such a design.

B. Previous Works
B.1. Fast Forwarding

Fast forwarding is used in some video players. Perceptual discontinuity is one of the indicators to show how smoothly the video can play. This scheme does not drop any frame, but increase the frame rate on some parts of the chunk. Fast forwarding begins from the front part of Chunk $C(2)$. The frame rate at fast forwarding part is twice ($k = 2$) of that at other parts. Let d be the length of delay and FR_n be the normal frame rate. The total number of frames that are played in the fast forwarding area can be expressed as:

$$TF_{ff} = k \times d \times FR_n \qquad (1)$$

The delay time is used to determine the frame rate for some part of the chunk. If the delay is 0.5 seconds, the number of skipping frames is 15 for the frame rate of 30 frames per second (fps). To catch up these 15 delayed frames, fast forwarding needs to play them plus 15 additional (normal) frames in 0.5 seconds. Therefore, we play the first 30 frames of Chunk 2 in 60 fps frame rate. The illustration of fast forwarding method is shown in Figure3.
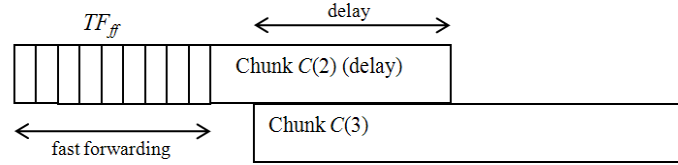

Figure 3. Fast forwarding process on delayed chunk video.

For the 4.5-second delay, we play all frames of Chunk 1 and some frames in the first part of Chunk 2 (a total of 270 frames) in 60 fps frame rate. The rest of the frames are played normally in 30 fps. Figure 4 shows an example of fast forwarding for the 4.5-second delay scenario.
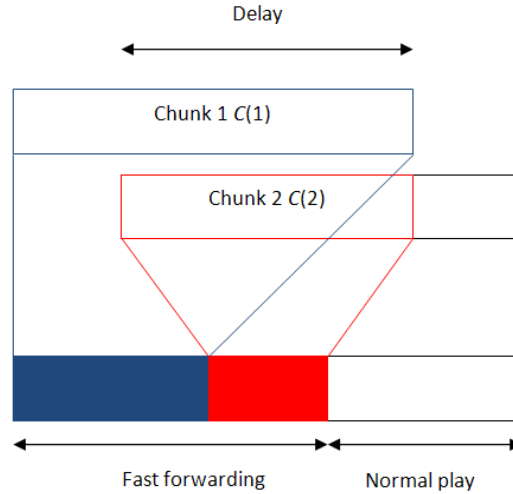

Figure 4. A fast forwarding example for the 4.5-second delay scenario.

The total length of video play in a normal condition is 10 seconds (300 frames). Since the delay considered here is 4.5 seconds, 135 frames are skipped if played in 30 fps. These delayed frames

plus the same amount of additional frames need to be played in 4.5 seconds. In other words, we need to do the fast forwarding for 270 frames using the frame rate of 60 fps. The rest of the frames are played normally in 1 second using the frame rate of 30 fps.

B.2. Uniform Frame Skipping

The perceptual impact by frame freezing and frame skipping is highly content dependent. Viewers prefer the scenario with a single long segment of freezing frames than the scenario with multiple short segments of freezing frames [19]. In frame skipping, some frames will be skipped in order to reduce the number of frames. This is the simplest way to reduce the number of frames. In our experiment, we start from the first frame of Chunk *C*(2), jump over one frame to the third frame, and so on until the required number of frame to be dropped is reached. The illustration of such a uniform frame skipping scheme is shown in Figure5. The dropping process does not consider the content of the frame but only follows the uniform dropping pattern. It will drop 15 and 135 frames for 0.5- and 4.5-second delay scenarios, respectively.
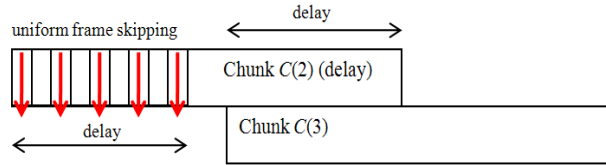

Figure 5. Uniform frame skipping process on delayed chunk video.

C. Proposed Approach

C.1. Frame Loss Visibility Information

Packet loss visibility was proposed by some researches and addressed in [20] - [22]. It provides important information to determine which packet can be dropped when network congestion occurs. The factors for packet loss visibility prediction are given in [22], which were first proposed by the authors in [20]. They developed a model for the visibility prediction using three subjective experiment data sets that span various encoding standards (H.264 and MPEG-2), group-of-picture structures, and decoder's error concealment choices. They considered the factors not only within a packet, but also in its vicinity, to account for possible temporal and spatial masking effects. They discovered that the factors of scene cuts, camera motion, and reference distance are highly significant to the packet loss visibility. They applied their visibility model to packet prioritization for a video stream; when the network gets congested at an intermediate router, the router is able to decide which packets to drop such that visual quality of the video is minimally impacted.

The visibility model is used for in-network quality monitoring of transmitted video. The visibility model is computed for the specific loss pattern that is observed in the network. Packet loss visibility can be computed based on the modeling process using Generalized Linear Model (GLM) [23] to determine the model of frame loss. GLMs are an extension of classical linear models.

The parameter p_i is modeled as a function of P factors. Let X represent a matrix of size N x P, where each row i contains the P factors influencing the corresponding parameter p_i. Let x_{ij} be the element of X located at (i, j). A generalized linear model can be represented as:

$$g(p_i) = \gamma + \sum_{j=1}^{P} x_{ij} \beta_j \tag{2}$$

where $g(\cdot)$ is called the link function, which is typically nonlinear, and β_1, $\beta_{1,....}$ β_P, are the coefficients of the factors. Coefficients β_j and the constant term γ are usually unknown and need to be estimated from the data. For logistic regression, the link function is the logit function, which is the canonical link function for the binomial distribution. The logit function is defined in [22].

The loss of whole frame is less noticeable by a viewer than packet loss (partial loss of a frame) [24]. Dropping the whole frame can reduce the probability that a viewer notices the artifact when a

packet is lost. Thus, the whole frame loss visibility is considered here. The frame loss visibility of bitstream video is described as a probability value as shown in Figure6, where a lower probability corresponds to higher dropping priority. Factors for predicting whole frame loss visibility are shown in Table 1.

Table 1. Factors for Predicting Whole Frame Loss Visibility [24]

Order	Factors	Coefficients
α	I	-2.3502
1	MeanMotM	8.5907e-2
2	VarMotY	-2.4423e-3
3	$\log(\text{MaxRSENGY} + 10^{-7})$	5.7905e-2
4	VarMotX	-7.5725e-4
5	MeanSliceSize × VarMotY	4.8017e-7
6	NumInterMB	-6.0581e-4
7	MaxMotM	3.6750e-3

The factors that used by the proposed system to generate bitstream with frame loss visibility information were described in [22] and [24], where an experiment of subjective assessment was performed by asking a viewer to notice which frame did the viewer see a glitch. The responses from viewers constitute the basis to construct the probability that a particular frame loss is visible.
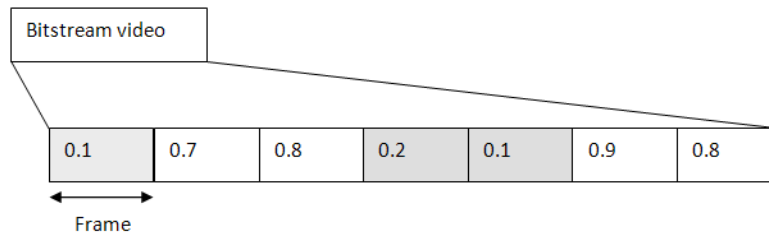

Figure 6. Illustration of frame loss visibility information (a probability value) on the bitsream video.

A model was developed to predict the probability of a lost frame being visible to viewers based on the factors related to some attributes of frame loss [24]. The visibility for the whole frame to be dropped was estimated by the visibility model for single-slice packets. That is, the visibility score for the frame was taken to be simply the sum of the visibility scores for the slices which compose the frame. And those visibility scores for slices came from a model designed using a human observer experiment involving slice loss data [24].

From a frame, it can obtain factors that can be extracted without the need for other frames. Therefore, it is no need to consider initial MSE and other metrics involving operations related to pixel domain reconstruction (as pixel reconstruction would require access to the reference frame). By this, the frame loss visibility can be determined even in the case that does not have the access to other frames. In [24], the authors considered the residual energy distribution of the MBs in a frame, denoted by RSENGY. They took the average of the residual energy of all the MBs in a frame and denoted as MeanRSENGY. Similarly, MaxRSENGY denotes the maximal residual energy after motion compensation among all MBs in a frame; VarRSENGY denotes the variance of the residual energy of MBs in a frame. Two more descriptions of the distribution are the skewness of RSENGY which describes the amount of asymmetry of the RSENGY distribution, denoted as SkewRSENGY, and the entropy of RSENGY which captures the randomness of the RSENGY distribution, denoted as EntRSENGY. In addition to RSENGY, the QP distribution used for each MB is also included. In H.264, the partition of a MB is supported, so the Interparts distribution of MBs in a frame is included as a factor. Another important factor involves motion vectors. MotX and MotY are motion

vector–distributions in *x* and *y* directions of MBs in each frame, respectively. MotM, the motion magnitude distribution of MBs in a frame, was considered. To compute the factors related to phase of motion vectors, they only considered MBs with non-zero motion, for which the phase is well defined. They denoted the phase information distribution of the motion vectors as MotA. The packet size distribution in bits in a frame, denoted as SliceSize, is also included for prediction. For each one of these distributions (QP, Interparts, MotX, MotY, MotM, MotA and SliceSize), they included the Mean, Max, Var, Skew and Ent (as we do for RSENGY) as predictive features in the model. In addition, they were interested in how the way MBs are coded can affect the frame loss visibility, thus we include the number of MBs in a frame that are coded in the mode of INTRA (NumIntraMB), INTER (NumInterMB), DIRECT (NumDirectMB) and SKIP (NumSkipMB) into factor consideration. The residual energy factor after taking the logarithm operation was more correlated to frame loss visibility (where they added 10^{-7} before taking the log to avoid a log of zero problems). More detail information about the factors can be found in [24].

C.2. *Frame Dropping Based on Frame Loss Visibility Information*

The frame loss visibility is described quantitatively as a probability value for each frame to represent the dropping priority of that frame. These values are sorted and the frame with the lowest value is dropped first, followed by the one with the second lowest value, and so on, until the required number of frames is reached. We performed frame dropping on Chunk $C(2)$ and Chunk $C(3)$ based on frame loss visibility information at 3 different areas of the chunk. An illustration of the proposed method for these three different areas is shown in Figure7, 8 and 9, respectively. For convenience, they are called Scheme A, Scheme AB, and Scheme B, respectively.
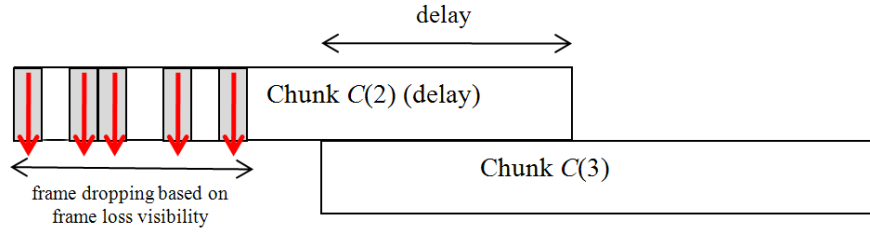

Figure 7. Frame dropping process based on frame loss visibility information on delayed chunk video (Proposed method with Scheme A).
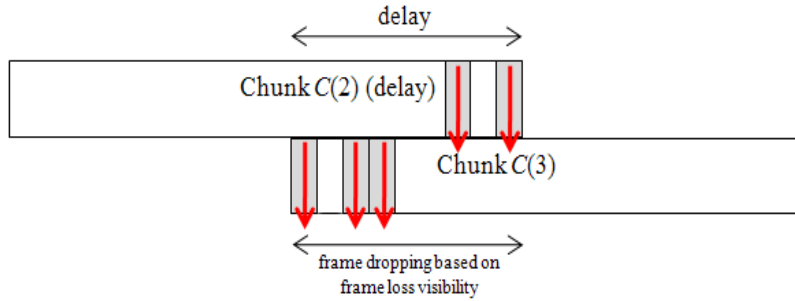

Figure 8. Frame dropping process based on frame loss visibility information on delayed chunk video (Proposed method with Scheme AB).

Assuming that the chunk generator cuts the video file into chunks without knowing the video content, we choose a convenient frame dropping area that consists of half of Chunk $C(2)$ and half of Chunk $C(3)$. Therefore, only the frame loss visibility information on that area will be considered in order to expedite the sorting process.
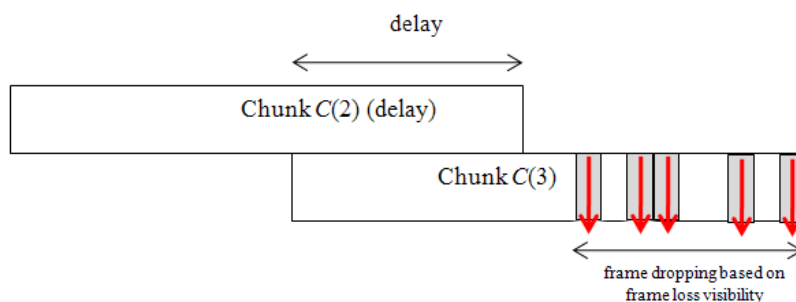
Figure 9. Frame dropping process based on frame loss visibility information on delayed chunk video (Proposed method with Scheme B).

C.3. *An Adaptive Frame Dropping Scheme*

Since different parts of chunk may have different video contents, an adaptive frame dropping scheme is designed to determine which part of the chunk is more suitable for applying frame dropping based on frame loss visibility information.

According to our experimental results for the 3 test sequences with very different characteristics, we found that the proposed method with one of the three schemes does not always perform better than the two traditional methods. However, given different types of test sequence and different parts of chunk, at least one of the schemes of the proposed method performs better than the two traditional methods. Therefore, we propose an adaptive system based on a switching strategy to choose the best suitable scheme of the proposed method for a particular chunk.

In the switching strategy, we are interested in the scheme which can produce the highest MOS in each video sequence and in every delay scenario. The strategy is derived from the distribution of frame loss visibility value in each part of the chunk. For each scheme among Scheme A, AB and B, we calculate the number of frames whose frame loss visibility value is less than or equal to P_r (= 0.4 in our experiment). We define this value as NumLessVis for each scheme. Then we will adaptively choose the scheme whose NumLessVis is the largest among the three schemes as our best method.

D. Experiments and Results
D. 1. Design of Experiments

The number of frames in each chunk is half of the total number of whole video sequence. We use three video test sequences and they represent the video with different levels of motion, i.e., slow, fast, and the mixture of both. All of the sequences have pixel spatial resolutions of 720 by 480. They are all compressed by H.264. Their normal frame rate is 30 fps. They use the I-B-B-P Group of Picture (GOP) structure and the GOP length is less than 15. More detail about the test sequences can be found in [25].

The chunk level distortion of the processed video is generated by dropping several frames of test video sequences that use the three different methods. Each video sequence is divided by two parts as Chunk $C(2)$ and Chunk $C(3)$.

The delay time considered in the experiment is set to be 0.5, 2.5 and 4.5 seconds, where the former is assumed to be the lowest delay and the latter is chosen to be the longest delay that can occur for our simulation given the length of our test sequences. The chunk size is half of the bitstream file for the video sequence (around 1.5 MB).

Viewers compare the proposed method with the two other methods and give scores for each method. The methods for comparison are fast forwarding method and uniform frame skipping method. All methods are implemented in Matlab with suitable signal and image processing techniques [26][27].

We define a *comparison set* to be the rating of fast forwarding, uniform frame skipping and the proposed method (in randomized order) for a given video sequence at a particular delay scenario. We generated 6 comparison sets using the 3 video sequences and the 3 delay scenarios. Each

comparison set was rated by 30 viewers and each viewer took about an hour to complete. None of the viewers is an expert. We had explained the perceptual discontinuity problem and showed the viewers a score table with the five quality levels defined earlier before they performed the experiments.

First, they were asked to watch the first original video sequence. Then they started to watch the video with chunk-level impairment. They watched five impairment videos and one hidden reference video (original video). It repeated for different video sequences. And it repeated for three different delay scenarios, that is, 0.5 seconds, 2.5 seconds, and 4.5 seconds.

D.2. Subjective Assessment Methodology for Video Quality (SAMVIQ)

The subjective and objective quality of video has been studied [28]. The rating procedure for generating a comparison set was realized by SAMVIQ (Subjective Assessment Methodology for Video Quality) [29][30][31].

We used the MSU Perceptual Video Quality Tool [32], which was developed based on SAMVIQ, to perform the subjective experiments. We told the viewer to focus on temporal quality of video. In each video evaluating process, after the viewer watches the video, he or she can move a slider to give a number from 0 to 100 as a score.

D.3. Experiments Results

Previous psychophysical experiments in [33] were designed to investigate the effects of size, location, blur, and contrast on the perceived visual interest of objects within images. The test sequences used here have different objects of interest and image characteristics, and therefore the experimental results are better described and discussed on the basis of individual sequence. Furthermore, since original video always has the highest MOS value as compared with other video test sequences, this part of discussion will be almost skipped in the future. For the lowest delay scenario the video impairment is not serious and the evaluation results for the original video and other sequences are supposed to be similar.

D.3.1 Formula Sequence

Figure 10 shows the MOS result for the Formula test sequence. The original test sequence (denoted as ori) which serves as the hidden reference always has the highest MOS for all delay scenarios. Most viewers can easily recognize the original video which appears in random order in the subjective test.
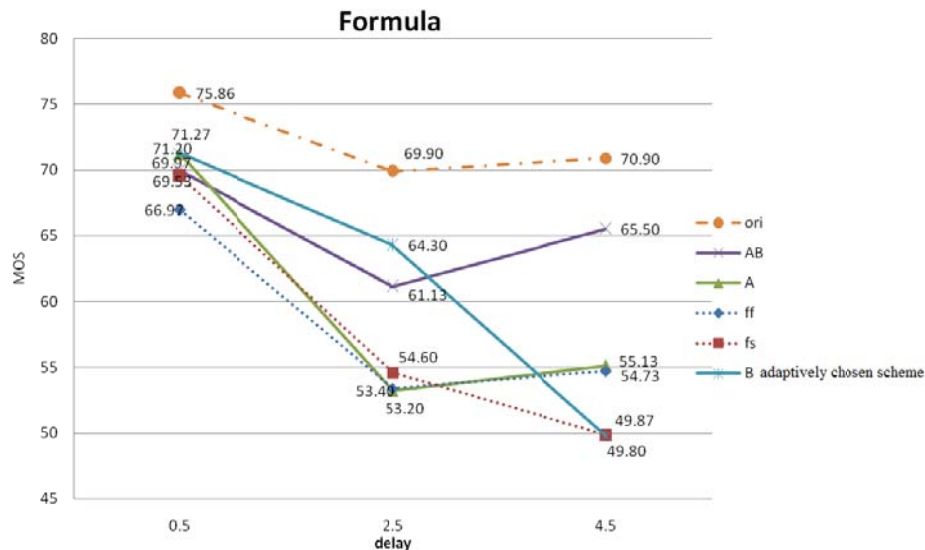

Figure 10. MOS of subjective assessment for Formula sequence.

In the 0.5-second delay scenario, the proposed method with Scheme B performs slightly better than the other two methods. In the 2.5-second delay scenario, the proposed method with Scheme B also performs slightly better than the other two methods. However, in the 4.5-second delay scenario, the proposed method with Scheme AB outperforms the other two methods by a more significant margin. The advantage of the proposed method is that we choose to drop the frames within a fast moving scene that is difficult to recognize by the viewer.

Fast forwarding (ff) performs better than uniform frame skipping (fs) at the higher delay scenario, indicating that fast forwarding is more suitable than uniform frame skipping for high motion scenes. The reason could be that fast forwarding does not drop any frame but play the video with a higher frame rate. Furthermore, the car racing scene in the Formula sequence has fast enough motion such that fast forwarding produces little noticeable artifact.

In almost all performance results, a higher delay corresponds to a lower MOS value. The exceptions are the proposed method with Schemes AB and A, and fast forwarding in the 4.5–second delay scenario. The performance of the proposed method with Scheme B drops significantly in the 4.5-second delay scenario because the final part of Formula sequence contains low motion scene.

D.3.2. New York Sequence

Figure 11 shows the MOS result for the New York test sequence. The uniform frame skipping method performs better than the fast forwarding method for the low motion scene. In the 0.5-second delay scenario, the proposed method with Scheme B performs better than the other two methods. In the 2.5-second delay scenario, uniform frame skipping performs better than the other methods. The proposed method with Scheme B has the second highest MOS in this delay scenario. In the 4.5-second delay scenario, the proposed method with Scheme AB outperforms the other methods by a more significant margin.
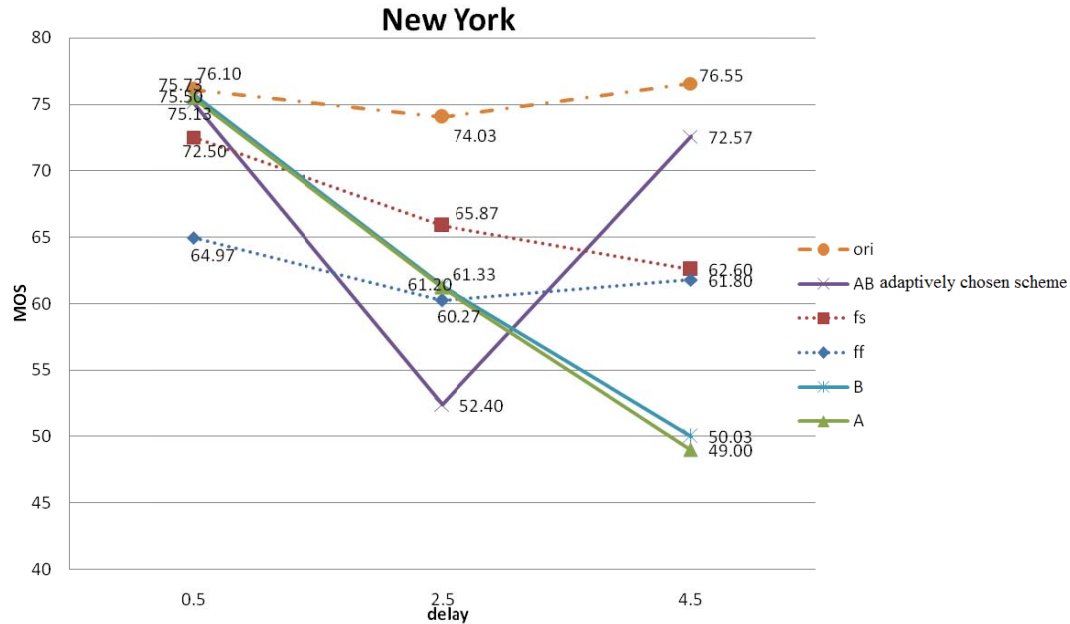

Figure 11. MOS of subjective assessment for New York sequence.

Again, the proposed method with Scheme AB gives the best performance for the 4.5-second delay scenario. A possible reason is that the middle part of the sequence has a higher spatial frequency and contains almost the same content for each frame. The proposed method with Scheme A got the worst MOS at the 4.5-second delay scenario because the front part of Chunk $C(1)$ has an important scene that becomes noticeable by the viewer. In the 2.5-second delay scenario, the proposed method with Scheme AB does not perform well because the frames dropped affect the

continuous scene of the video.

Uniform frame skipping (fs) performs better than fast forwarding for low motion video sequence. Fast forwarding (ff) is easily noticed by a viewer even in the low delay (0.5-second delay) scenario.

D.3.3 Soccer Sequence

Figure 12 shows the MOS result for the Soccer test sequence. The proposed method with Scheme AB still has the best performer in the 4.5-second delay scenario. The proposed method performs only slightly better than uniform frame skipping for the 0.5-second delay scenario. The reason is that we performed uniform skipping on the front part of Chunk 1, where it only has little impact on the assessment of video quality for the viewer. Fast forwarding is better than uniform frame skipping in the case of 4.5-second delay. In this particular case, the viewers prefer to see faster playback scene rather than jerky scene on the sport video.
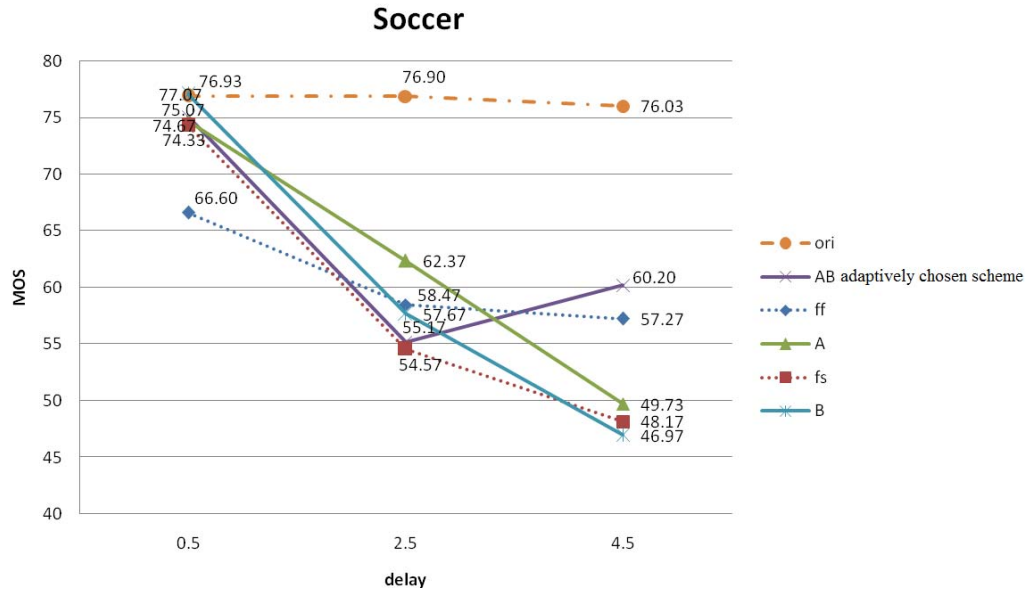

Figure 12. MOS of subjective assessment for Soccer sequence.

D.4 Performance of Adaptive Scheme

In Table 2, we summarize the performance of the proposed method based on the result of our adaptive scheme.

Table 2. Performance of Proposed method based on Result of Adaptive Scheme

Test Sequence	Proposed Method with Chosen Scheme and the MOS performance	Average MOS gain over Uniform Frame Skipping Method	Average MOS gain over Fast Forwarding Method
Formula	B, 61.79	+3.79	+3.06
New York	AB, 66.70	-0.29	+4.36
Soccer	AB, 63.48	+4.46	+2.70

D.5 Discussion

The delay problem in P2P actually is not as simple as that in our experiment. It could have a huge number of variations. For the low delay (0.5-second) case, MOS values of all three methods are almost the same because viewers cannot identify the delay effect very well. For the 2.5-second delay scenario, our proposed method does not perform well because the viewers notice the artifact that is caused by constantly skip 2 frame dropping pattern rather than consecutive frame dropping [19]. For the longer delay (4.5-second) case, the proposed method almost always achieves the best MOS performance except for the New York sequence that has medium shot in the whole video that is easily noticed by viewer. Due to length limitation of the test sequence, it affects our method in finding low visibility frames on the limited area of chunk. Fort the New York sequence which contains the scenes captured by a continuously moving camera, the performance of uniform frame skipping method is slightly better than that of our adaptive scheme, where the Scheme AB was selected automatically.

For the slow speed motion scene containing static objects, uniform frame skipping is quite acceptable by viewers. On the other hand, fast forwarding is acceptable by viewers for a sport sequence, where the objects of interest contained in the frames of the sequence are moving constantly. Some comments from the viewers even show that the side effect of using fast forwarding method can make the sport sequence looks lively.

Conclusions and Future Works

Perceptual discontinuity in P2P video streaming occurs when the chunk of video from a peer has a delay. The perceptual continuity between adjacent chunks needs to be maintained for better visual quality. Otherwise, the viewer who watches the video may feel some temporal information loss.

In this study, we proposed a frame dropping scheme based on frame loss visibility information. We successfully evaluate the performance of the proposed method and two other methods in 3 delay scenarios by subjective experiments. The MOS obtained for all methods decreases as the delay time increases because more frames are sacrificed. Fast forwarding method is more suitable for fast motion scenes, while the uniform frame skipping is more suitable for low motion scenes. The proposed method is suitable for both types of scenes. The adaptive frame dropping scheme improves the performance of the proposed method. It can select the area of the chunk with low frame visibility information and do the frame dropping process.

For the future work, more test sequences with different types of image characteristic can be used to evaluate the proposed method. Furthermore, we shall study how to implement this playback control design with the proposed method and embed the resulting design into a real P2P live video streaming system.

References

[1] T. Z. J. Fu, W. –T. Leung, P. –Y. Lam, D. M. Chiu, and Z. Lei, "Perceptual Quality Assessment of P2P Assisted Streaming Video for Chunk-level Playback Controller Design," *18th Int. Packet Video Workshop*, pp. 102-109, Dec. 2010.

[2] T. Z. J. Fu, D. M. Chiu, and Z. Lei. Designing QoE Experiments to Evaluate Peer-to-peer Streaming Applications. [Online]. Available:
      personal.ie.cuhk.edu.hk/~zjfu6/publication/QoE_vcip_final.pdf

[3] Setton and B. Girod, Peer-to-peer Video Streaming, Springer, New York, 2007.

[4] J. F. Bufford, H. Yu, E. K. Lua, P2P Networking and Applications, Elsevier, Burlington, 2009.

[5] X. Shen, H. Yu, J. Bufford, and M. Akon, Handbook of Peer-to-Peer Networking, Springer, New York, 2010.

[6] S. Androutsellis-Theotokis and D. Spinelliss. A Survey of Peer-to-Peer Content Distribution Technologies. [Online]. Available:
      http://www.spinellis.gr/pubs/jrnl/2004-ACMCS-p2p/html/AS04.pdf

[7] Z. lu, W. Lin, B. C. Sheng, S. Kato, and S. Yao, "Perceptual Quality Evaluation on Periodic Frame-Dropping Video," *IEEE Int. Conf. on Image Processing*, vol. 3, pp. III – 433, Oct. 2007.

[8]   K. –C. Yang, C. C. Guest, K. El-Maleh, and P. K. Das, "Perceptual Temporal Quality Metric for Compressed Video," *IEEE Trans. on Multimedia*, vol. 9, no. 7, pp. 1528, Nov. 2007.

[9]   R. R. P. Vidal, J. –C. Gicquel, C. Colomes, and H. Cherifi, "Frame Dropping Effects on User Quality Perception," *Int. Workshop on Image Analysis for Multimedia Interactive Services*, April 2004.

[10] C. Kiraly, L. Abeni, and R. Lo. Cigno, "Effects of P2P Streaming on Video Quality ," *IEEE Int. Conf. on Comunications*, pp. 1, May 2010.

[11] S. Menasche, A. A. A. Rocha, E. S. Silva, R. M. M. Leao, D. Towsley, and A. Vemkataramani, "Modeling Chunk Availability in P2P Swarming Systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, Oct. 2009.

[12] Y. Liu, "Delay Bounds of Chunk-Based Peer-to-Peer," IEEE/ACM Trans. on Networking, vol. 18, no. 4, Aug. 2010.

[13] Russo and R. L. Cigno. Delay-Aware Push/Pull Protocols for Live Video Streaming in P2P Systems. [Online]. Available:
http://napa-wine.eu/twiki/pub/Public/DocumentsOld/PID1016418.pdf

[14] PPTV website. [Online]. Available: http://www.pptv.com/

[15] X. Hei, C. Liang, J. Liang, Y. Liu and K. W. Ross. A Measurement Study of a Large-Scale P2P IPTV System. [Online]. Available:
http://cis.poly.edu/~ross/papers/P2PliveStreamingMeasurement.pdf

[16] Y. Huang, T. Z. J Fu, D. –M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P-VOD system," *ACM SIGCOMM Conf. on Data Communication*, vol. 38, no. 4, Oct. 2008.

[17] N. Hegde, F. Mathieu, and D. Perino. Size Does Matter (in P2P Live Streaming). [Online]. Available: http://hal.inria.fr/docs/00/41/46/74/PDF/RR-7032.pdf

[18] Y. Liu. On the Minimum Delay Peer-to-peer Video Streaming: how Realtime can it be? [Online]. Available: http://eeweb.poly.edu/faculty/yongliu/docs/mm07.pdf

[19] Y. Qi and M. Dai, "The Effect of Frame Freezing and Frame Skipping on Video Quality," *Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, pp. 423, Dec. 2006. W. J. Heng and K. N. Ngan, Digital Video Transition Analysis and Detection, World Scientific*, Singapore, 2002.

[20] S. Kanumuri, S. G. Subramanian, P. C. Cosman, A. R. Reibman, and V. Vaishampayan, "Predicting H.264 Packet Loss Visibility using a Generalized Linear Model," in *ICIP. IEEE*, 2006, pp. 2245–2248.

[21] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. Vaishampayan, "Modeling Packet-Loss Visibility in MPEG-2 Video," *IEEE Trans. Multimedia*, vol. 8, pp. 341–355, April 2006.

[22] T. –L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, "A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization," *IEEE Trans. on Image Processing*, vol. 19, pp. 722, March. 2010. http://en.wikipedia.org/wiki/GLM

[23] T. –L. Lin, Y. –L. Chang, and P. C. Cosman, "Subjective Experiment and Modeling of Whole Frame Packet Loss Visibility for H.264," *18th Int. Packet Video Workshop*, spp. 186, Dec. 2010.

[24] R. Christanto, S. –G. Miaou, and T. –L. Lin, "Reducing Perceptual Discontinuity in Peer-to-peer Live Video Streaming Based on Frame Loss Visibility Information," *3rd Int. Conf. on Electrical Engineering and Informatics*, Jul. 2011.

[25] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, Digital Image Processing Using Matlab 2nd Edition, McGrawHill Educations, Gatesmark, 2011.

[26] R. C. Gonzalez and R. E. Woods (Adapted by S. –G. Miaou), *Digital image Processing, 3rd Edition, an Adapted Version, Pearson Education Taiwan, Taipei,* 2008.

[27] K. Seshadrinathan, R. Soundararajan, A.C. Bovik, and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video," *IEEE Trans. on Image Processing*, vol. 19, pp. 1427, June 2010.

[28] SAMVIQ – Subjective Assessment Methodology for Video Quality, EBU Project Group B/VIM Video in Multimedia, BPN 056 report, May 2003.

[29] "ITU-R Recommendation BT.500V11: Methodology for the Subjective Assessment of the Quality of Television Pictures," ITU-R 211/11, 2004.

[30] J. L. Blin, "New Quality Evaluation Method Suited to Multimedia Context SAMVIQ,"[Online].Available:enpub.fulton.asu.edu/resp/vpqm2006/papers06/313.pdf

[31] MSU Perceptual Video Quality Tool. [Online]. Available: http://compression.ru/video/quality_measure/perceptual_video_quality_tool_en.html

[32] V. Kadilaya, S. Pinelli, E. Larson, and D. M. Chandler. Quantifying the Perceived Interest of Objects in Images: Effects of Size, Location, Blur, and Contrast. [Online]. Available: http://vision.okstate.edu/pubs/Vamsi_HVEI.pdf

Ricky Christanto obtained his B. Eng in Electronics and Computer Engineering Degree from Satya Wacana Christian University, Indonesia in 2005 and his M. Sc in Electronics and Computer Engineering Degree from Chung Yuan Christian University, Taiwan in 2011. From 2007-2009, he worked as lecturer at Department of Information Technology of Satya Wacana Christian University, Indonesia. During 2009 to 2011, he worked as researcher in Multimedia Computing and Telecommunication (Mu-CaT) Laboratory, Taiwan under the supervision of Professor Shaou-Gang Miaou. His research interests include but not limited to image processing, perceptual quality of video streaming, routing, and peer-to-peer network.

Shaou-Gang Miaou obtained the B.S. degree in electronic engineering from the Chung-Yuan Christian University (CYCU), Chung-Li, Taiwan, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the University of Florida (UF), Gainesville, FL, USA in 1990 and 1993, respectively. From 1983 to 1988, he was a Research Assistant at the Chung-Shan Institute of Science and Technology, Lung-Tan, Taiwan. From 1989 to 1993, he was a TA in the Dept. of Electrical Eng. or an RA in the Center for Information Research, UF. In 1993, he joined the Dept. of Electronic Eng., CYCU, as an associate prof. He was promoted to a full prof. in 2002 and served as the Chairman of that department from 2004 to 2008. His current research interests include data compression, pattern recognition, and wireless multimedia communications. He is an Editor for the *Int. J. of Electrical Eng.* and the *J. of Advanced Eng.* He has authored or coauthored over 120 technical papers in referred journals and conference proceedings. He is the author or coauthor of several books on digital image processing, communication systems, probability and statistics, and engineering mathematics. Prof. Miaou is a permanent member of the Image Processing and Pattern Recognition Society of R.O.C., a member of the Biomedical Eng. Society of R.O.C. and a senior member of IEEE.

Ting-Lan Lin obtained his B.S. and M.S. degrees in Electronic Engineering from Chung Yuan Christian University, Chung Li, Taiwan in 2001 and 2003, respectively. He obtained his Ph.D. degree in Electrical and Computer Engineering at the University of California, San Diego, La Jolla, CA, USA in 2010. During the summer of 2008, he interned in the Display System group at Qualcomm, San Diego, CA. In 2011, he joined the faculty of the department of Electronics Engineering at Chung Yuan Christian University, Taiwan, where he is currently an Assistant Professor. His research interests include video compression, video streaming in lossy networks, optimization of packet prioritization and perceptual video quality.