Variable Length Segmentation Technique for Dynamic Video Streaming

Rana Fareed Al-Tuma[*] and Rasha Ismaiel Ahmed[**]

University of Technology - Computer science Department – Baghdad - IRAQ

[*]110016@uotechnology.edu.iq
[**]110108@uotechnology.edu.iq

Received: 20 October 2017 **Accepted: 8 August 2018**

Abstract

To transfer videos using dynamic streaming approach, they must be divided into segments; these segments would be compressed to different quality levels to be delivered from the source to the destination, with fewer amounts of delay and packet loss according to the network's statues at the sending time. The almost used segmentation techniques depend on the time that the segment time length is constant within the same video file. This paper presents a suggestion to take into consideration the contents of the video to split the video file into segments. The length of the segments depends on the length of video scenes, which made the scene length varies according to the content of the video file itself.

Keywords: video streaming, segment length, video quality, dynamic streaming, frame dropping.

تقنية تجزئة الطول المتغير للتدفق الديناميكي للفيديو

رنا فريد غني و رشا إسماعيل احمد

الجامعة التكنولوجية - قسم علوم الحاسوب

الخلاصة

لنقل الفيديو باستخدام نهج تدفق الديناميكي، يجب تقسيم الفيديو إلى أجزاء. ثم يتم ضغط هذه الأجزاء على مستويات جودة مختلفة ليتم إرسالها من المصدر إلى المستلم بمقادير أقل من التأخير والفقدان وفقا لخصائص الشبكة في وقت الإرسال. إن تقنيات التقسيم المستخدمة حاليا تعتمد على كون أطوال الأجزاء في ملف الفيديو ثابتة. يقدم هذا البحث اقتراحا يأخذ في الاعتبار محتويات الفيديو لتقسيمه إلى أجزاء. حيث يعتمد طول الجزء على طول المشهد ضمن الفيديو.

الكلمات المفتاحية: بث الفيديو، طول المشهد، جودة الفيديو، التدفق الديناميكي، حذف صورة.

Introduction

Videos are the most used multimedia data types over the internet nowadays since videos usually are big file size then transferring them over the internet would require too many resources and high network throughput, to solve this problem many solutions were proposed and one of them is the dynamic video streaming. Dynamic streaming is the streaming solution which transfers videos from one quality level to lower one over the internet according to the network's parameters statues like the available bandwidth and buffer size.

Since a video file consists of a sequence of frames; then dropping some frames for any reason would affect the continuity of motion. In [1] researchers characterize the effect of frame dropping at a constant rate either alone or combined with a burst of discarded pictures for different durations on perceived quality.

The quality of the videos delivered to the client is an important aspect in video streaming, many researches study the quality of video files and the effect of reducing the video data on the quality itself in [2, 3] the researchers study many video quality measurement techniques and their correlation with the subjective video quality measurement.

Video files to be transmitted from the server to the client using a dynamic streaming approach, first they have to be divided into segments. Video segments differ in length according to the providers, for Silver Light Smooth streaming (Microsoft Smooth Streaming MSS) the segment length is 2 seconds, for HTTP live Streaming the segment length is 10 seconds and in HTTP Dynamic streaming the segment length is 2-5 second [4, 5, 6, 7, 8]. Also, compression techniques are applied to these segments in order to reduce their size to suite the network's available bandwidth. The reduction of segment size in proportion to network resources availability allows smooth segment delivery and play at the client side. The most common compression techniques that are used with video streaming are H26X, VP8 and VP9.

In this work, a new segmentation technique of video file is suggested. This technique differs from the almost adopted dynamic adaptive streaming over HTTP protocol (DASH) technique in the inconstancy of segment length. The suggested segmentation technique divides the video into segments of variable size according to the frame belonging to the video different scenes. Therefore, the suggested segmentation technique can be considered as contents-based segmentation technique.

The rest of this paper is organized as follows, section two explains the transmission model, section three explains the suggested video segmentation technique, section four will present the experimental results of the proposed techniques, and section five will present the conclusions.

The Transmission Model

There are two approaches for transmitting the video files from the server to the client; the video files in the first approach are transferred as a sequence of frames one by one from the server to the client. In the other approach, the video file is transmitted as a sequence of chunks, the chunk is a video segment, usually, the video file is divided to segments with fixed length (from 3 to 12 seconds length) and stored in many quality levels to be transmitted according to the network's available bandwidth [4].

When the video segments with specific quality sent according to the available bandwidth is called Dynamic Adaptive Streaming over HTTP Protocol (DASH)as shown in figure 1.

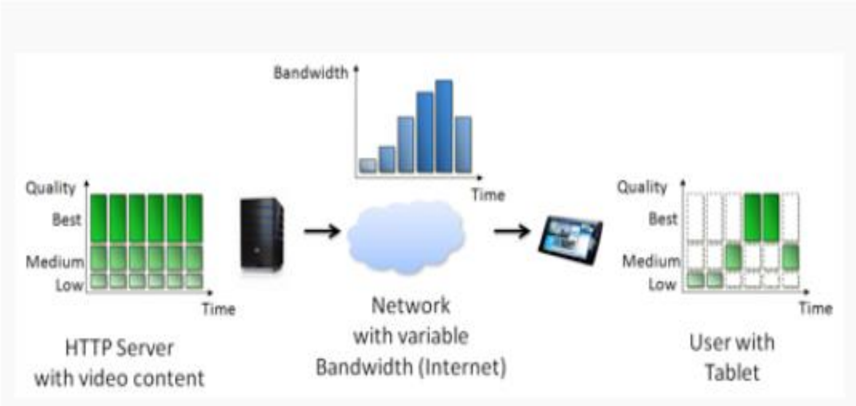


Figure 1: DASH Technology [10]

The suggested video segmentation technique

The suggested contents based video segmentation for video streaming technique is performed depending on the scenes in the video file itself. Each scene will be saved as a segment at the server side. Then, the segments of the video are compressed with a compression technique such as H.26x or VPx with 3 to 12 quality levels, after reducing the number of frames depending on the motion in the video scene. The process of segmenting the video file is implemented on stages: at first the video file is divided into N number of segments, the segments are varied in length according to the content of the video itself, in other words, the video file will be segmented depending on the scenes in the video file, each segment will contain a single scene of the video as shown in figure 2. If the scene is too long then this scene will be divided into sub-scenes, these scenes also with variable length depending on the difference between the consecutive frames in the scene itself.

The algorithm below describes the steps of splitting the video file

Algorithm 1: for video splitting

Input:
V: video file
F: video frame
Thr : threshold for splitting
Output:
SN: scene number
S: video scene
For each F ∈ V
    diff = $F_{n+1} – F_n$
    If diff > Thr **then**
    SN =SN+1
    Create (S_{SN})
    End if

The threshold value suggested is 0.95, this value can be decreased, this will result to increase the number of dropped frames and decrease the quality of the constructed video file.

Figure 2: Variable Length Segments Video Streaming

Experimental results

Three videos [9] have been used with different properties in the paper as shown in table 1.

Table 1: video samples properties

Sample name	duration	Video size	Frame dimensions	Video type
Rotation	10 s	1.18 MB	720 x 576	YUV Color
bs1	8.6 s	102 MB	768 x 432	YUV Color
st1	10 s	118 MB	768 x 432	YUV Color

In order to divide a video file into segments depending on the content, first compute the amount of motion difference between the consecutive video frames. If the difference between two consecutive frames is greater than a defined threshold; which is equal to 0.95 %; then the first scene has ended and the next scene starts as shown in the following example:

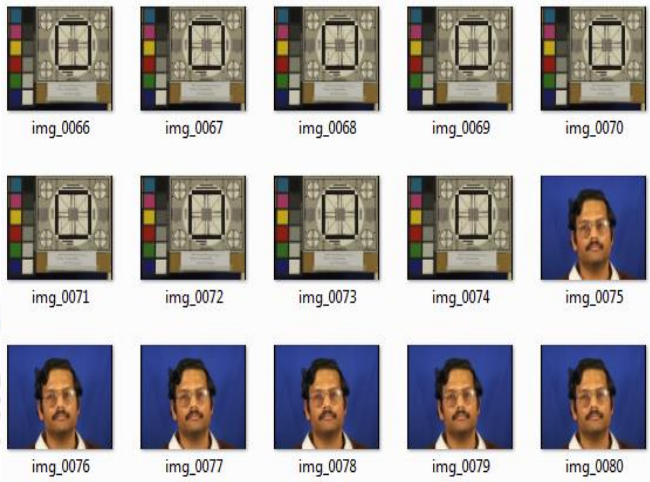


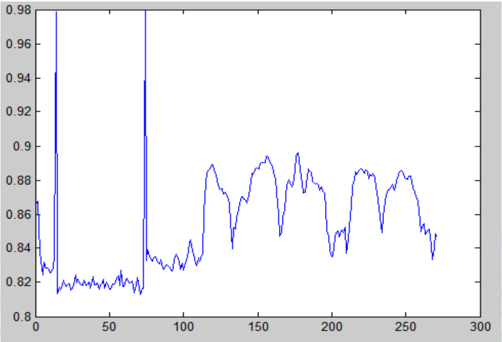
Figure 3: Sequence of frames from Rotation Video Sample

Table 2 shows the differences between every two consecutive frames, and depending on this result the video is divided into scenes.

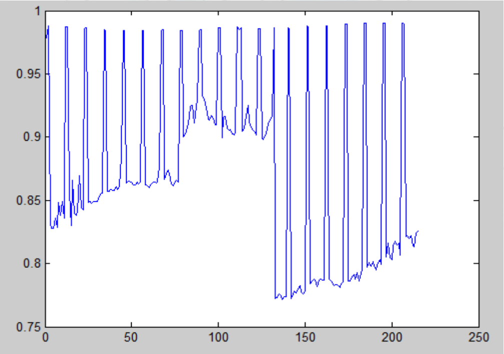
Table 2: Difference Between Each Two Consecutive Frames

Frame N	Frame N+1	Motion difference
Img_0066	Img_0067	0.09
Img_0067	Img_0068	0.07
Img_0068	Img_0069	0.08
Img_0069	Img_0070	0.1
Img_0070	Img_0071	0.08
Img_0071	Img_0072	0.07
Img_0072	Img_0073	0.08
Img_0073	Img_0074	0.08
Img_0074	Img_0075	0.9
Img_0075	Img_0076	0.2
Img_0076	Img_0077	0.3
Img_0077	Img_0078	0.3
Img_0078	Img_0079	0.2
Img_0079	Img_0080	0.2

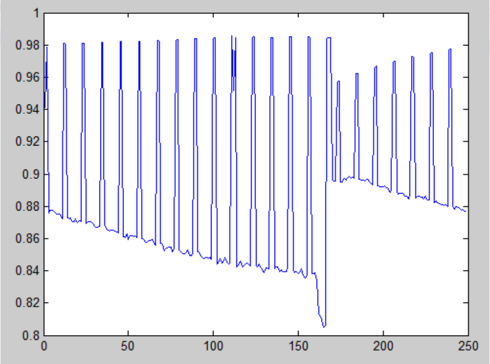
The second stage of the suggested video segmentation is that when a scene is longer than three seconds (or 90 frames), then this scene would be divided into sub-scenes, to avoid long segments to be transmitted from the server to the client. Figure 4 shows the scenes in the three sample videos.



a) scenes splitting for Video Sample Rotation



b) scenes splitting for Video Sample bs1



c) scenes splitting for Video Sample bs2

Figure 4: Scenes splitting

Each video segment will also be reduced temporally; this is achieved by deleting frames with the least difference value from their preceding frame. Reduction mechanism is performed by computing the ratio of the frames to be deleted from the segment, equation (1).

$$RTD = \frac{(N \times Difference)}{100} \qquad \text{-----------(1)}$$

N= frame No. within a segment

Difference = diff (1st frame, Nth frame)

Where RTD is the Ratio of frames to be deleted.

The less amount of difference represents less amount of motion in the frames, so visually there will be no motion stalls during the displayed video at the client side. Table 3 shows the number of frames that would be deleted from a video scene.

Table 3: Number of frames to be deleted from a video scene

Video	Scene	No. of Frames	Difference	No. of deleted frames
Rotation	Scene1	14	0.7	0.8 ≈ 1
	Scene2	59	0.4	2.3 ≈ 3
	Scene3.1	90	0.8	7.2 ≈ 7
	Scene3.2	106	0.9	9.5 ≈ 10
bs1	Scene1	14	0.9	1.2 ≈ 1
	Scene2	10	0.9	0.9 ≈ 1
	Scene3	10	0.9	0.9 ≈ 1
	Scene4	10	0.9	0.9 ≈ 1
	Scene5	10	0.9	0.9 ≈ 1
	Scene6	10	0.9	0.9 ≈ 1
	Scene7	10	0.9	0.9 ≈ 1
	Scene8	10	0.9	0.9 ≈ 1
	Scene9	10	0.9	0.9 ≈ 1
	Scene10	10	0.9	0.9 ≈ 1
	Scene11	10	0.9	0.9 ≈ 1
	Scene12	10	0.9	0.9 ≈ 1
	Scene13	10	0.9	0.9 ≈ 1
	Scene14	10	0.9	0.9 ≈ 1
	Scene15	10	0.9	0.9 ≈ 1
	Scene16	15	0.9	1.3 ≈ 1
	Scene17	10	0.9	0.9≈ 1
	Scene18	19	0.9	1.7 ≈ 2
St1	Scene1	14	0.9	1.2 ≈ 1
	Scene2	10	0.9	0.9 ≈ 1
	Scene3	10	0.9	0.9 ≈ 1
	Scene4	10	0.9	0.9 ≈ 1
	Scene5	10	0.9	0.9 ≈ 1
	Scene6	10	0.9	0.9 ≈ 1

	Scene7	10	0.9	0.9 ≈ 1
	Scene8	10	0.9	0.9 ≈ 1
	Scene9	10	0.9	0.9 ≈ 1
	Scene10	11	0.9	0.9 ≈ 1
	Scene11	10	0.9	0.9 ≈ 1
	Scene12	10	0.9	0.9 ≈ 1
	Scene13	10	0.9	0.9 ≈ 1
	Scene14	10	0.9	0.9 ≈ 1
	Scene15	11	0.9	0.9 ≈ 1
	Scene16	15	0.9	1.3 ≈ 1
	Scene17	10	0.9	0.9 ≈ 1
	Scene18	10	0.9	0.9 ≈ 1
	Scene19	10	0.9	0.9 ≈ 1
	Scene20	10	0.9	0.9 ≈ 1
	Scene21	10	0.9	0.9 ≈ 1
	Scene22	8	0.9	0.7 ≈ 1

The frames that will be deleted are the frames that have the less amount of motion difference with their former frames.

After deleting the most redundant frames from the video file, the server rebuilds the video scenes again, for the deleted frames; a sign will be added to the former frame to indicate that there is a deleted frame after this frame that holds the sign.

The client-side application will reconstruct the video segment, when a sign appears during the reconstruction phase the client will automatically repeat the frame that holds the sign, to compensate the deleted frames.

The Structural similarity (SSIM) index quality measurement has been used, it is an important measurement which has been shown to be more effective in the human vision system (HVS) [11], table 4 describes the SSIM results of the original and the distorted videos.

Table 4: SSIM Results of The Original and The Distorted Videos

Video	Scene	No. of deleted frames	SSIM result
Rotation	Scene1	1	0.999
	Scene2	3	0.999
	Scene3.1	7	0.999
	Scene3.2	10	0.998
bs1	Scene1	1	0.978
	Scene2	1	0.973
	Scene3	1	0.972
	Scene4	1	0.971
	Scene5	1	0.970
	Scene6	1	0.968

	Scene7	1	0.966
	Scene8	1	0.964
	Scene9	1	0.960
	Scene10	1	0.963
	Scene11	1	0.958
	Scene12	1	0.956
	Scene13	1	0.955
	Scene14	1	0.956
	Scene15	1	0.960
	Scene16	1	0.969
	Scene17	1	0.950
	Scene18	1	0.538
St1	Scene1	1	0.978
	Scene2	1	0.973
	Scene3	1	0.977
	Scene4	1	0.979
	Scene5	1	0.978
	Scene6	1	0.982
	Scene7	1	0.982
	Scene8	1	0.982
	Scene9	1	0.983
	Scene10	1	0.984
	Scene11	1	0.984
	Scene12	1	0.985
	Scene13	1	0.986
	Scene14	1	0.987
	Scene15	1	0.989
	Scene16	1	0.992
	Scene17	1	0.990
	Scene18	1	0.990
	Scene19	1	0.991
	Scene20	1	0.991
	Scene21	1	0.991
	Scene22	1	0.98

Conclusions

This work suggested variable length segmentation of video files as an improvement to DASH streaming technique. Each segment represents a scene within the video file.

In order to reduce the video file size, which is usually large in size, in the proposed method, the number of deleted images represents the number of frames to be deleted.

without affecting the motion within the video file. the selected threshold value (0.95) can be increased, which will lead to a larger number for deleted frames, which will affect the temporal discontinuity of the motion. In this case, freeze can be noticed by the viewer. Freeze appears when displaying the received video with a large number of deleted frames.

Depending on the suggested scene segmentation, the number of possible frame dropping is computed according to the difference between the consecutive frames as shown in table 2. Also, the location of the dropped frames is taken into consideration, no Two consecutive frames deleted from the scene even if they have the lowest motion difference to avoid the staling problem in case there will be more than two frames deleted from the same scene.

The quality of the video file before and after frame dropping has been measured, using SSIM quality metric, showing the change of quality between the original video file version and the constructed version after dropping a specific number of frames according to the threshold value (0.95) as shown in table 4.

After the frame reduction phase the server will be able to apply any compression technique that is suitable with the video streaming like H264 or VP8, then sends the video file to the requesting client.

References

1. Y. Wang, “Survey of Objective Video Quality Measurements “, EMC Corporation Hopkinton, MA 01748, USA R. R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi, “Framedropping effects on user quality perception,” in Proc. 5th Int. WIAMIS,Lisbon, Portugal, 2004.
2. R. R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi, “Framedropping effects on user quality perception,” in Proc. 5th Int. WIAMIS,Lisbon, Portugal, 2004.
3. ZHU Tao, WANG Jin-ming, ZHANG Xiong-wei, ZHANG Peng, "Distortion Estimation Method for Coefficient Dropping of H.264 Bit-stream",Institute of Communications Engineering, PLA Univ. of Sci. & Tech., Nanjing, China, jdcta.vol4.issue5.12,2015.
4. M. Seufert, S.Egger, M.Slanina,”A Survey on Quality of Experience of HTTP Adaptive Streaming”, IEEE, March 13,2015.
5. A. Zambelli, Smooth streaming technical overview, Microsoft Corp.,Redmond,WA, USA, Tech. Rep. [Online]. Available: http://www.iis.net/learn/media/on-demand-smooth-streaming/smooth-streamingtechnicaloverview

6. Apple Inc., HTTP Live Streaming Overview 2013. https://developer.apple.com/library/ios/documentation/networkinginternet/conceptual/streamingmediaguide/Introduction/Introduction.html

7. Adobe Systems Inc., HTTP Dynamic Streaming on the AdobeFlash Platform 2010. [Online]. Available:https://bugbase.adobe.com/index.cfm?event=file.view&id=2943064&seqNum=6&name=httpdynamicstreaming_wp_ue.pdf

8. K. Seshadrinathan, R. Soundararajan, A. C. Bovik and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video", IEEE Transactions on Image Processing , vol.19, no.6, pp.1427-1441, June 2010

9. Laboratory for Image and Video Engineering, www.live.ece.utexas.edu

10. Thomas Stockhammer , "Dynamic adaptive streaming over HTTP: standards and design principles", ACM Degital Library,2011 .[Online]. Available: https://dl.acm.org/citation.cfm?id=1943572.

11. Jianji Wang, Nanning Zheng, Badong Chen, Jose C. Principe, " Associations among Image Assessments as Cost Functions in Linear Decomposition: MSE, SSIM, and Correlation Coefficient ", Institute of Artificial Intelligence and Robotics, arXiv ,4 august 2017.