January 2013

AUDIO VIDEO SYNCHRONIZATION USING BITSTREAM SYNTAX

KHYAMADITA NAYAK

ECE Dept. The Oxford College of Engineering, Bangalore, khyamadita.nayak@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijess>

Part of the [Electrical and Electronics Commons](#)

Recommended Citation

NAYAK, KHYAMADITA (2013) "AUDIO VIDEO SYNCHRONIZATION USING BITSTREAM SYNTAX,"
International Journal of Electronics Signals and Systems: Vol. 2 : Iss. 3 , Article 18.

DOI: 10.47893/IJESS.2013.1110

Available at: <https://www.interscience.in/ijess/vol2/iss3/18>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Electronics Signals and Systems by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

AUDIO VIDEO SYNCHRONIZATION USING BITSTREAM SYNTAX

¹KHYAMADITA NAYAK & ²MEENA K V

^{1&2}ECE Dept. The Oxford College of Engineering, Bangalore

Abstract - This paper has proposal to develop audio-video synchronization in the bit-stream syntax using the Slice type and Macro-Block usage. The Media file like MP4, MOV and M4V container formats consists of both Audio and Video packed as tracks. The Video is encoded in H.264 format while Audio is encoded in AAC formats. There is a common way to achieve synchronization by looking into the type of Slice and the type of MB. This can be achieved by scanning across all the atoms present within the Media data. Time synchronization can be achieved by taking input from the Sample to Time atoms and feeding the data into Auxiliary units. The Slice type plays important role for synchronization as the decode order and display order for Video are different and sufficient time is taken to decode the video based on Profile, Level, type and number of Macroblocks for High profile. So the synchronization can be achieved for a Program stream or a Digital storage medium playback by synchronizing the formats based on type of slice information.

Keywords: H.264 video, AAC audio, Multiplexing/de-multiplexing.

I. INTRODUCTION

With better quality and less bandwidth requirement, digital television transmission has already replaced analog television transmission in a big way. With the advent of HDTV, transmission schemes are aiming at transmitting superior quality video with provision to view both standard format and wide screen (16:9) format along with one or more audio streams per channel. Digital video broadcasting (DVB) in Europe and advanced television systems committee (ATSC) in North America are working in parallel to achieve high quality video and audio transmission. Choosing the right video codec and audio codec plays a very important role in achieving the bandwidth and quality requirements. H.264, MPEG-4 part-10 or AVC, achieves about 50% bit rate savings as compared to earlier standards. H.264 provides the tools necessary to deal with packet losses in packet networks and bit errors in error-prone wireless networks. These features make this codec the right candidate for using in transmission.

Advanced audio coding (AAC) is a standardized lossy compression scheme for audio. This codec showed higher coding efficiency and superior performance at both low and high bit rates, as compared to MP3 and AC3. The H.264 video and AAC audio coded bit streams need to be multiplexed in order to construct a single stream. The multiplexing process mainly focuses on splitting the individual streams into small packets, embedding information to easily realign the packets, achieving lip sync between the individual streams and providing provision to detect and correct bit errors and packet losses.

Factors to be considered for multiplexing and transmission :

- Split the video and audio coded bit streams into smaller data packets

- Maintain buffer fullness at de-multiplexer without buffer overflow or underflow
- Detect packet losses and errors
- Send additional information to help synchronize audio and video.

Packetization

There are two layers of packetization process carried out. First layer of packetization yields packetized elementary stream (PES) and the second layer yields transport stream (TS). This second layer is what is used for transmission. Multiplexing takes place after the second layer of packetization, just before the transmission.

Packetized elementary stream

The packetized elementary stream (PES) packets are obtained by encapsulating coded video, coded audio, and data elementary stream. This forms the first layer of packetization. The encapsulation on video and audio data is done by sequentially separating the elementary streams into access units. Each PES packet contains data from one and only one elementary stream. PES packets may have a variable length since the frame size in both audio and video bit streams is variable. The PES packet consists of the PES packet header followed by the PES packet payload. The header information distinguishes different elementary streams, carries the synchronization information in the form of timestamps and other useful information.

Fig. 1: PES from elementary stream

Frame number as timestamp

The proposed method uses frame number as timestamps. Both H.264 and AAC bit streams are composed of data blocks sorted into frames. A particular video bit stream has a constant frame rate during playback specified by frames per second (fps). So, given the frame number, one can calculate the time of occurrence of this frame in the video sequence during playback as follows:

$$\text{Time of playback} = \text{Frame number} / \text{fps} \quad (1)$$

AAC compression standard defines each audio frame to contain 1024 samples. The audio data in the AAC bit stream can have any discrete sampling frequency between 8 – 96 kHz. The frame duration increases from 96 kHz to 8 kHz. However, the sampling frequency and hence the frame duration remains constant throughout a particular audio stream. So, the time of occurrence of the frame during playback is as follows:

$$\text{Playback time} = \frac{1024 * \text{frame number}}{\text{Sampling rate}} \quad (2)$$

Thus from (1) and (2) we can find the time of playback by encoding the frame numbers as the time stamps. In other words, given the frame number of one stream, we can find the frame number of the other streams that will be played at the same time as the frame of the first stream. This will help us synchronize the streams during playback.

This method has the following advantages over using clock samples as timestamps.

- Less complex and more suitable for software implementation.
- Saves the extra PES header bytes used for sending the program clock reference (PCR) information periodically.
- No synchronization problem due to clock jitters or inaccurate clock samples.
- No propagation of delay between audio and video due to drift between the master clocks at the transmitter and receiver.

Transport stream

The second layer of packetization forms a series of packets called as transport stream (TS). These are fixed length sub divisions of the PES packets with additional header information. These packets are multiplexed together to form a transport stream carrying more than one elementary stream. A TS packet is 188 bytes in length and always begins with a synchronization byte of 0x47. This structure of the packet was originally chosen for compatibility with ATM systems. However there are some applications

where more bytes are added at the end to accommodate error correction data like Reed-Solomon or CRC error check data. There are a few constraints to be met while forming the transport packets:

- Total packet size should be of fixed size (188 bytes).
- Each packet can have data from only one PES.
- PES header should be the first byte of the transport packet payload.
- PES packet is split or stuffing bytes are added if the above constraints are not met.

The encapsulation of PES packets to form TS packets is shown in Fig 2

Fig 2: TS packet formation from PES packet

TS packet header

The TS packet normally consists of a 3 byte header. However, if adaptation field is present, then an additional byte is added to the header and 184 bytes are available for payload including the adaptation field. The header structure of the TS packet with the description of the syntax is as follows.

- *Payload unit start indicator :*
Single bit flag set to indicate presence of PES header in the payload.
- *Adaptation field control :*
Single bit flag to indicate presence of any data other than PES data in payload.
- *PID (Packet identifier) :*
Each elementary stream has a unique 10 bit PID. Some are reserved for NULL packets or PSI (Program Specific Information). Sequence parameter set and picture parameter set are sent as PSI at frequent intervals. Null packets are used to maintain constant bit rate.

- *Continuity counter :*

4 bit rolling counter which is incremented by 1 for each consecutive TS packet of the same PID. To detect packet loss.

- *Payload byte offset :*

If adaptation field control bit is '1', byte offset value of the start of the payload or the length of adaptation field is mentioned here.

Proposed Bit-stream Synchronization method:

Synchronization during transmission plays an important part in avoiding buffer overflow or underflow at the Decoder. The Video buffer verified should neither overflow nor underflow.

If lot of information is sent by the Stream the decoder should not be overloaded as the video should be displayed in 25 frames or 30 frames per second. In order to prevent this scenario we have proposed a method to overcome by looking into the type of the Slice and initiate the audio decoding so that the video and audio data is fed to the Auxiliary units and follows the display order for the video thus rendering minimal delay for synchronization.

The MP4, MOV and M4V organizes information of media as tracks and each track consists of either Video or Audio or Meta data only. So synchronization between the tracks play important role. The Video tracks consists of video samples and each sample is of different length which consists of different types of slices like I, P, B. The audio samples are equally distributed and each sample is of same length. The size of time to decode and the time taken to display depends on various factors.

I slice has maximum residues and size of each I slice is indicated by start of IDR within NAL header.

P slice has fewer residues and it is inter predicted slice.

B slice has minimal residues and it is bi-directionally inter predicted

$$T(I \text{ slice}) \approx \sum MB(I) * T(IDCT)$$

$$T(P \text{ slice}) \approx \sum MB(I) * T(IDCT') + \sum MB(P)$$

$$T(B \text{ slice}) \approx \sum MB(I) * T(IDCT'') + \sum MB(P) + \sum MB(B)$$

$$T(IDCT) > T(IDCT') > T(IDCT'')$$

T(Slice) is proportional to size of I slice.

Audio samples can be decoded faster than video but cannot be ∞ . So The proposed solution is mentioned in below figure

When there is a I slice header in NAL the number of MB's present within the Slice should be

counted and add delay based on (Number of MB's / Frame rate) * Jittery time .

For B and P frames this includes the Motion vectors also which is significantly lesser than I frames.

Results:

The compression achieved, using H.264 video codec and AAC audio codec, in the proposed method is more than that of the MPEG-2 movie files (.mpg) or AVI files.

Some random TS packets are chosen to begin the de-multiplexing process and the results of the synchronization process are shown in Table 1. The delay between audio and video is less than 6 milliseconds in the observed cases. This delay remains constant throughout the playback time once synchronized

Start TS packet number	Synchronized frame numbers chosen		Video frame playback time (sec)	Audio frame playback time (sec)	Delay (msec)	Visual delay
	Video	Audio				
120	18	34	0.72	0.723	3.22	Not perceptible
2000	107	191	4.08	4.074	5.97	Not perceptible
6000	282	529	11.28	11.284	3.57	Not perceptible
23689	1303	2453	52.32	52.322	2.49	Not perceptible

Fig. 3: Calculation of playback time using slice type

CONCLUSIONS

This paper has developed an effective transport stream that would carry multiple elementary streams and deliver a synchronized multimedia program to the receiver. This is achieved by adopting 2 layers of

packetization of the elementary stream followed by a multiplexing procedure. The proposed multiplexing procedure enables the user to start de-multiplexing from any TS packet and achieve synchronized playback for any program. There is also provision for error detection and correction after receiving the packets. These are absolutely essential for video broadcasting applications. Usage of H.264 video bit stream and AAC audio bit stream helps transmit high quality audio-video at less bit rates. This meets all the basic requirements to transmit a high quality multimedia program.

REFERENCES

- [1] MPEG-2 advanced audio coding, AAC. International Standard IS 13818-7, ISO/IEC JTC1/SC29 WG11, 1997.
- [2] MPEG. Information technology — generic coding of moving pictures and associated audio information, part 3: Audio. International Standard IS 13818-3, ISO/IEC JTC1/SC29 WG11, 1994.
- [3] MPEG. Information technology — generic coding of moving pictures and associated audio information, part 4: Conformance testing. International Standard IS 13818-4, ISO/IEC JTC1/SC29 WG11, 1998.
- [4] MPEG-4: ISO/IEC JTC1/SC29 14496-10: Information technology – Coding of audio-visual objects - Part 10: Advanced Video Coding, ISO/IEC, 2005.
- [5] P. V. Rangan, S. S. Kumar, and S. Rajan, "Continuity and Synchronization in MPEG," IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 52-60, Jan. 1996.
- [6] "FAAC and FAAD AAC software, website : www.audiocoding.com