Objective Verification of Audio-Video Synchronization

Veli-Tapani Peltoketo
Sofica Ltd, Tiedekatu 2, 60320 Seinajoki, Finland
veli-tapani.peltoketo@sofica.fi

Abstract — The digital multimedia and especially video presentations are very widely used nowadays. Quality issues, like audio-video synchronization problems, are very common due to varying hardware and software environment. Several digital video encoding standards and different hardware and software implementations create very complicated combinations and the verification of each combination is demanding but essential. There are several real time algorithms that detect the synchronization problems and correct them in real time. Still, the real time solutions have some weaknesses and they concentrate more to fix the possible problems than to verify and measure the quality of the whole video pipeline. There is a demand on objective and general testing system, which could offer comparable test results of audio-video synchronization between digital video systems. In this research, the most promising algorithms are compared and the most suitable algorithm combination is selected and tested. According to the results, a general method can be found even if more work and comprehensive testing is required before robust solution can be released.

Index Terms — audio codecs, video codecs, synchronization, automatic testing

I. INTRODUCTION

The modern world is full of digitalized multimedia and the role of this presentation type is increasing all the time. There are a lot of different multimedia components, for example, in the mobile phones, computer applications, home pages and advertising. The definition of the multimedia is quite wide, it may include texture, still images, audio, video, animation, interactive features and combinations of all these. However, the role of the digital video presentation is more and more popular. This trend has highlighted the quality of the digital video broadcasting.

The quality of the video can be measured using several different parameters like frame count, noise, resolution and color accuracy. An essential quality parameter is also the audio-video synchronization which defines the timing difference of the sound and vision components in the video representation. Humans are very sensitive to detect the difference between visual representation and corresponding audio. Less than 100 ms difference can be detected especially when it is question of a lip synchronization problem, i.e., the voice is not synchronized to the lip movements.

There are several reasons why the audio-video synchronization is a quite common quality problem. The fundamental reason of the problem is that the digital video pipeline is very long and contains several different components, like recording devices, encoding, encoder buffering, multiplexing, transmission, demultiplexing, decoder buffering decoding and presentation devices [1]. Moreover, in most of these components the audio and video data are handled separately and the adjusting between audio and visual data is very dependent on the implementation and hardware and software environment.

Onwards, there are several encoding and decoding methods, for example, MPEG2 (Moving Picture Experts Group), AVI (Audio Video Interleave) and Quicktime. Especially the newest codecs are quite heavy to execute. Also the modern audio systems with several audio channels, like Dolby surround voice with six audio channels, increase the complexity of the video handling. Moreover, there are several hardware and software techniques to implement encoding and decoding; the functionality can be done using DSPs (Digital Signal Processor), FPGA-chips (Field Programmable Gate Array), unique ASICs (Application-Specific Integrated Circuit) or by software. Especially, when the codecs are made using software, the performance is dependent on the used processor, the load of the processor and, the most importantly, the software implementation skills.

Why the performance is so crucial in case of audio-video synchronization? The amount of the visual data is much bigger than audio data and the delays which are generated to the audio and video signals are typically unequal [1]. The adjustment of the audio and video data inside the heavy encoding and decoding process is very demanding and the requirements of the adjusting operation are strict. Moreover, the current market trend forces to raise the frame count of the video which also increases the load of the codecs. The result can be noticed quite often when there is a timing difference between audio and video signals.

There are several real-time algorithms which are verifying and adjusting the audio-video synchronization during video recording and playback. Still, the real-time solutions have some weaknesses and they concentrate to fix the possible problems rather than to verify and evaluate the quality of the whole stream pipeline. There is a strong demand on objective and general testing system, which could offer comparable test results of the audio-video synchronization between digital video systems. Fortunately, different mathematical algorithms and statistical methods give powerful tools, which can be used to measure and validate this phenomenon.

This paper describes the first steps towards the general audio-video synchronization measurement algorithm. The paper concentrates mainly on evaluating and comparing the most promising methods of the audio-video synchronization detection. The final optimization and implementation as well as the comprehensive testing with large reference data will be done in the next phase.
II. CURRENT SITUATION

There are two main facts which are affecting to the evaluation of the audio-video synchronization. Firstly, the audio-video synchronization requirements are defined in several standards, which give a good base to verify and judge the quality of this feature. Furthermore, the current real time algorithms offer several solution proposals even thought they are not suitable as such.

A. Standardization

The audio-video synchronization requirements are standardized very accurately because the same problems are valid both in the television broadcasting and the modern digital video broadcasting, because the acceptable delay is dependent on the end user experience, not on the used technology.

The standardized limits help significantly the evaluation of the video signal and give strict limits to the synchronization. Three different organizations have given recommendations of the audio-video synchronization as described in Table I. The delayed audio represents situation, where the audio data of the video stream is played too late and correspondingly the advanced audio is the artifact where the audio data is played too early.

<table>
<thead>
<tr>
<th>Standard</th>
<th>Year</th>
<th>Delayed audio</th>
<th>Advanced audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSC IS-191, [1]</td>
<td>2003</td>
<td>45</td>
<td>15</td>
</tr>
<tr>
<td>EBU R37, [2]</td>
<td>2007</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>Subjective evaluation by ITU-R, undetectable limits</td>
<td>1998</td>
<td>100</td>
<td>20</td>
</tr>
</tbody>
</table>

Even if the standards give certain limits of the acceptable difference between the sound and visual component in the video handling, the verification of the video quality has to be much more accurate. The evaluation of the video has to detect also such kind of problems which are acceptable, but near to the limits. Also, a jitter, a small variance of the audio-video synchronization, has to be detected.

B. Available Methods

In general, modern codecs have already functionalities, which are ensuring the audio-video synchronization. For example, the MPEG standard contains timestamp based approach, where the MPEG encoding calculates the difference between audio and video signals and stores the difference to the stream. Onwards, when the video stream is decoded, the difference data is used to adjust the audio and visual signal correctly. Still, the synchronization result depends on the implementation of the standard and the quality of the final video stream is not always the best one.

As mentioned before, there are several real time algorithms which are handling audio-video synchronization. For example, the synchronization can be recovered using correlation analysis [7]. The study proposes a method which is based on face detection and adjusts the synchronization when human faces are recorded. Also a bimodal linear prediction model is used to handle a speech synchronization detection [6].

Furthermore, there are methods, which are detecting the audio-video synchronization using hidden watermarks of the visual data and also algorithms which are inserting the audio data to the corresponding video frames [5], [9].

There are obvious advantages when real time methods are used. The audio-video synchronization issues are not only evaluated but also fixed as well as possible. The real time methods measure the video stream and have possibility to adjust to the current situation.

However, there are also many restrictions in the real time methods or, actually, the real time methods are not suitable to the strict evaluation and verification. They do not test the video quality by giving results to the tester but try to ensure as good synchronization as possible. Furthermore, the characteristics of the real time methods bind the method implementation tightly to the used codec implementation and the outcome is not universal but suitable only to corresponding codec or even suitable to the specific video content when face detection based algorithms are used.

Maybe the most obvious restriction is that the real time methods are executed in the target environment, which is normally an embedded system like a mobile phone. The high quality evaluation of the audio-video synchronization requires heavy mathematical algorithms, a lot of time and a powerful processor. In case of an embedded real time system, such resources are unusual. One solution is to evaluate the existing video and audio data in a separate environment which have enough capacity. This solution offers also possibility to create a codec independent evaluation algorithm.

In the beginning of this study, the most promising research was done by Radhakrishnan, Bauer, Cheng and Terry [8]. Based on the signature calculations of the audio and video data, the correlations were calculated between the reference and processed data. According to the correlation and Hamming distance calculations, the quality of the audio-video synchronization can be measured [8]. This research was selected to be the first algorithm, which was implemented and tested. However, the hash results of the audio stream were not as unique as expected but clear trends were noticed which caused faulty hits during the stream comparison. Therefore, another study by Haitsma and Kalker [3] was also evaluated.

III. METHODS

The used algorithms [8] and [3] are based on the usage of two streams, the reference and processed ones. Furthermore, the [3] is intended only to audio fingerprinting. Figure 1 describes the main calculations steps of both algorithms. The figure does not contain the signature and robust hash calculation blocks of the reference data because the reference data is static and the calculations can be done beforehand. The
feedback loop from the best hit selection block to the processed hash selection is an addition to the original structure and done to both optimize and extend the current algorithms.

![Diagram of signature extraction process]

Fig. 1. Steps of the difference calculation

The calculation of the video signatures is very similar than the audio one, only the spectrogram calculation differs; in case of the video, spectrograms are not used, but the differences of the consecutive frames are used. The coarse presentation, hash calculations and hamming distance are same than in the audio measurements when method [8] is used.

The feature extraction and robust hash parts include the differences between algorithms [8] and [3]. Where the former method uses coarse representation, the latter calculates energies of certain frequency ranges. Also the hash calculation differs; [8] uses random matrices and the [3] detects the changes between consecutive frequency values. Following chapters defines the details and differences of the methods.

A. Signature Extraction

The goal of the signature extraction is to create a signature which is robust against different changes that are done to the stream, such as compression and time scale modifications [8]. Despite this kind of changes, the signature of a certain clip has to be unique and identifiable against the reference data. The same calculations are made to the reference and processed streams.

During the signature extraction, the audio data is divided to parts and one signature is calculated from each part. Moreover, the parts are overlapped so that the consecutive parts are almost equal [8], [3]. In this study, the T origin which defines the difference between consecutive parts, is 50 audio samples and defines the accuracy of the synchronization verification. The size of the part T is 5120 which ensures the uniqueness of the signatures. Furthermore, the audio samples are transformed to the frequency domain, which is defined as S in the (1) and (2).

Equation (1) defines the coarse representation Q of [8], where W and W represent the time and frequency blocks of frequency-time representation S. The coarse representation averages the magnitude of frequency coefficients in time-frequency blocks. This study uses the same constant values F=20 and T=10 as the reference study.

\[
Q_a(k,l) = \frac{1}{W_i \times W_j} \sum_{i=(k-1)W_j}^{i=W_j} \sum_{j=(l-1)W_i}^{j=W_i} S(i,j)
\]

Equation (2) defines the frequency energy \( E(n,m) \) calculation of [3]. The frequency area of the certain audio samples \( W_t \) are divided to the frequency blocks \( k \) and sum of those frequency amplitudes are calculated. The number of the frequency blocks F is 33 which defines later the size of the hash number. However, the size of the frequency block \( W_t \) is not a constant but the frequency has a logarithmic spacing.

\[
E(n,m) = \sum_{i=(k-1)W_t}^{i=W_t} \sum_{j=(l-1)W_t}^{j=W_t} S(i,j)
\]

In the first phase, when method [3] is used, the bandwidth of the audio signal is filtered so that only the frequencies between 300-2000 Hz were used. This bandwidth is the most relevant spectral range for the human auditory system [3]. Despite this filtering, the [3] generates more unique values than [8]. However, the full range of audio frequencies has to be also evaluated and study, if the filtering causes inaccuracy to the measurements.

B. Robust Hash Extraction

The purpose of the robust hash calculation is two-fold: it filters the small changes by signal processing and it reduces the size of the signature.

The method [8] uses random matrices \( P_k \) and the size of the \( Q_a \) can be reduced to the \( k \) bits, where the \( k \) is the number of the random matrices. Before the projection \( H_k \) is calculated in (3), the matrix \( P_k \) is changed by removing the mean of the matrix from the components of the matrix.

Each bit of the robust hash is gotten by calculating \( H_k \) using corresponding \( P_k \). If the value of the \( H_k \) is greater than the median of all projections \( H_k \), the bit gets value '1', otherwise '0'. Finally, the robust hash is the combination of these bits, and the size of the robust hash is \( k \) bits.

\[
H_k = \sum_{i=1}^{F} \sum_{j=1}^{T} Q_a(i,j) \ast P_k(i,j)
\]

The principle of method [3] is different. It is based on the difference of consecutive audio samples as well as the difference between consecutive frequency blocks. If previous values are greater, the hash bit gets value '1', otherwise '0' (4). The number of frequency blocks defines the size of the hash number which is 32 bits.

\[
F(n,m)=1, \text{if } E(n,m) - E(n,m+1) - E(n-1,m) + E(n-1,m+1) > 0
\]
\[
F(n,m)=0, \text{otherwise}
\]

C. Hamming Distance

The hamming distance is calculated between multiple consecutive signatures (signature block W) of reference and processed data. The main procedure contains a loop, which compares a signature block of the reference data R against the same size of signature block of the processed data A. The same reference block is compared against several blocks in the processed data using a certain window size 2xL. The
corresponding hamming distances of each signature pairs are stored. Onwards, the next signature block is selected from the reference data and again compared against processed data inside the correspondingly shifted window. Using this loop, the whole data has been measured and the best hit of each signature block can be selected. \[8\]

\[
D(m,i) = \sum_{j=0}^{j=W} \text{HammingDistance}(R(i+j), A(m+j)) \quad (5)
\]

Figure 2 illustrates the hamming distance calculation of the (3). The signature block 1-W is compared between reference and processed data. When the whole window 2xL is measured, the signature block of the reference data and the window of the processed data are stepped forward.

There are three crucial parameters, which affect to the hamming distance calculation: the size of the signature block \(W\), the size of the searching window against the processed data \(2xL\) and the step size i.e. how much the signature block is moved between the comparisons. The step size affects significantly to the performance of the measurement whereas the number of the signatures affects to the reliability of the measurement and also to the filtering characteristics of the comparison - big signature block filters quick changes. The larger size of the searching window reduces the accuracy requirements of the feedback loop in the Figure 1, but also reduces the performance of the calculation.

D. Selecting the Best Hit

The obvious solution to calculate the best hit is defined in (4), where the \(D(m,i)\) is the result of the Hamming distance calculation \[8\].

\[
\text{best hit} = \arg \min_m D(m,i) \quad (6)
\]

However, there are cases in which the best hit (i.e., the smallest hamming distance), is found from incorrect place and the second or third best hit is the correct one. Therefore, the minimum distance should not be the only rule when selecting the right hit. The best hit can be validated, for example, by calculating the probability of the placement of the next hit according to the previous ones. In case of stereo and surround audio, the data of the other audio channels can be also used to filter incorrect hits. Nevertheless, this logic is not yet used in this study.

E. Adjusting Window of Processed Data

Normally, the window of the processed data is stepped on the same rate than the signature block of the reference data. However, there are situations in which this kind of logic does not work. For example, if the processed audio contains cumulative delay, sooner or later the corresponding signatures of the processed data will locate outside the measured window. This issue can be avoided by dynamically adjusting the window location depending on the latest measurements of the hamming distance.

IV. SOLUTION

The implementation of the methods is based on a layered architecture, which isolates the evaluation algorithm from the codec specific implementation as described in Figure 3. This kind of implementation is essential, when a generally usable and codec independent algorithm implementation is required.

Fig. 3. Layered architecture

The role of the codec specific part is to transform the processed data of the codec to raw audio and video format. The raw audio contains the digitalized samples of the audio data and raw video contains the frame based images.

The evaluation part makes the real verification work. It receives the raw data of the tested codec and the corresponding reference data. Using the signature based algorithms, it generates the evaluation results which contain the difference values between reference and processed streams. Different statistical values, like mean difference, maximum difference and variance can be measured from these values.

There is two ways to use the reference data. It can be even raw stream or it can contain ready signatures which are calculated beforehand. In this case, the random matrices have to be same when the signatures of the processed stream are calculated, if method \[8\] is used.
V. THE PROGRESS OF THE STUDY AND RESULTS

The first evaluation of the signature based algorithm contains hash distribution investigation and verification of three different audio-video synchronization problems: clipped audio data with decoding delay, clipped video data and cumulative audio delay.

A. Hash Distribution Investigation

Even if the method [8] seemed very usable algorithm, it was noticed that it generated several wrong hits when reference and processed audio streams were compared. When the corresponding hash numbers were evaluated, clear trends was noticed as Figure 4 shows. If there are hash numbers which are very near to each other's or even same, the probability of wrong hits increases significantly.

As Figure 5 defines, the hash distribution of the method [3] is significantly more uniformly distributed than previous method. (The number of hash values is equal in both figures.) This was also noticed in the best hit selection; the number of wrong hits was reduced notably.

The corresponding problem was not noticed when video data was evaluated. It is noteworthy to notice that method [8] uses the difference between consecutive frames as an input data when the signatures of video part were calculated. Onwards, the method [3] uses the difference between consecutive audio frequency energies when signatures were calculated.

This study uses the combination of these methods; method [8] is used to calculate video signatures and method [3] is used to calculate audio signatures.

B. Clipped Audio Data With Decoding Delay

Figure 6 describes a problem, when 500 ms of audio data is removed from the processed video. Moreover, during the audio removing process, the audio part of the video was converted from AAC to MP3 and back to AAC -format. Lossy codecs, like AAC and MP3 have characteristic which add delay to the beginning of the audio stream. The constant delay of MP3 decoding is 528 samples and in case of AAC decoding it is 2112 samples. However, the decoding delay of AAC may vary depending on the implementation. Due to decoding delay, the result has two opposite asynchronous artifacts; in the beginning there is decoding delay and in the middle the removed audio data can be seen. The amplitude of both delays follow the decoding and clipped audio data delays.

C. Clipped Video Data

Figure 7 describes a problem, when 10 frames of video data is removed from the processed video. Ten frames represents 400 ms when fps is 25. The asynchronous can be clearly seen from the figure and it is opposite to the previous measurement. The amount of the clipped data follow the generated delay exactly.
D. Cumulative Audio Delay

The processed audio data is cumulative and uniformly delayed 1%, meaning 2 seconds during the video. Figure 8 defines the result where the asynchronous can be noticed and the result follows the generated artifact.

VI. CONCLUSIONS

The combination of two signature based algorithms is a promising method for the measurement of the audio-video synchronization. The tested problems like cumulative delay and clipped audio and video data were detected correctly.

Also the result of the algorithms reveal more than audio-video synchronization evaluation. It also points out, how the timings of the audio and video parts are changed separately. This evaluation method can detect such problems that cannot be verified from the basic audio-video synchronization results.

There are quite many steps in the signature based algorithm. However, the implementation of the method is quite straightforward and can be done, for example, using C++ language. Also the performance facts encourage to use efficient language, like C++. It was clearly noticed, that the implementation is a tradeoff between performance and accuracy. It is possible to achieve a great exactness using, for example, big has numbers, but the verification time of videos will increase correspondingly.

Even if the parametrization of the methods requires more study as well as the optimization of the implementation, the current results give very positive signals that signature based evaluation can be used to measure the delays in audio and video streams. The next steps will reveal the how general implementation can be done using signature based methods.

A. Next Steps

The goal of the research is demanding; to find a general and objective method which is suitable to all video formats. This study is definitely the very first step towards the goal and much more work is required.

Even if the first evaluated methods gave good results, more methods have to be evaluated and tested. As the study has revealed so far, the combination of several methods may be the best way to implement comprehensive audio-video synchronization detection.

Another improvement is to optimize the signature algorithm and to find the optimal parameter values to the audio and video part. Also different video formats require parameter optimization as well as the usage of several audio channels.

Finally, the implementation requires more testing. A comprehensive testing using numerous audio and video formats is needed as well as validation using known audio-video synchronization problems.

REFERENCES