MPEG-4 Audio Synchronization

Masayuki Nishiguchi, Shusuke Takahashi, Akira Inoue

Oct 22, 2014
Sony Corporation
Agenda

- Use case
- Synchronization Scheme
- Audio Feature Extraction tool (Normative)
- Audio Feature Similarity Calculation Tool (Informative)
- Performance evaluation
- Conclusion
Audio Synchronization

Use case of "Second Screen" Application

- foreign language audio tracks
- audio commentary
- closed caption information
- audio/visual contents recorded from various angles
- high quality audio/visual contents
- advertisement
Synchronization Scheme

Transmitter
- Main Media Stream
- Sub Media Stream
- Audio Feature Extraction
- Audio Feature of Main Media Stream
- Audio Signal
- MUX

Receiver (Sub device)
- Main Device
- DeMUX & Playback
- Audio Feature of Main Media Stream
- Audio Playback device (such as speaker)
- Main Media Presentation
- Noise
- Audio Recording Device (such as microphone)
- Audio Signal of Main Media Stream
- Sub Device
- Audio Feature Extraction
- Audio Feature of Main Media Stream (Transmitted)
- Audio Feature of Main Media Stream (Extracted)
- Timing Adjustment & Playback
- Synchronization Information
- Sub Media Presentation
- Sub Media Stream
- DeMUX
- Audio Feature Similarity Calculation

Multiplexed Data Stream
- Sub Media Stream
- Audio Feature of Main Media Stream

(Synchronous in time)
Audio Feature Extraction tool (Normative)
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz) → Framing → Pre Emphasis

Filter bank → Auto-correlation → Integration → Peak Detection → Audio Feature Frame Rate Conversion → Audio Feature
Overall Signal Flow

split the audio signals into 5 equally spaced frequency bands in log frequency domain

Input Signal (After Pre emphasis filter)
\[ H(z) = 1 - 0.97 \cdot z^{-1} \]

Confidence Measure for this band is less than threshold

Integrate together into single Auto Correlation

Band split signal
Auto Correlation

Integrated Auto Correlation
converted into a 128-bit length feature vector \( f(k) \)
Prominent peak : 1
Otherwise : 0
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz)

Framing

Pre Emphasis

Filter bank

Auto-correlation

Integration

Peak Detection

Audio Feature Frame Rate Conversion

Audio Feature
Framing

Input frame interval: 8msec (64sample)

Input frame length: 32msec (256samples), Hamming Window

Feature Extraction

Audio Feature Frame Rate Conversion

Output frame interval: 8msec or 32msec (audio_sync_feature_time_resolution)
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz) → Framing → Pre Emphasis

Filter bank → Auto-correlation → Integration → Peak Detection → Audio Feature Frame Rate Conversion → Audio Feature
Filter Bank

For each audio frame, a pre-emphasis filter is applied to emphasize the high frequency, then band pass filtering is applied in order to split the audio signals into 5 equally spaced frequency bands in log frequency domain.
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz) → Framing → Pre Emphasis

Filter bank → Auto-correlation → Integration → Peak Detection → Audio Feature Frame Rate Conversion → Audio Feature
Auto-correlation

For each band, Auto-correlation is calculated using:

\[ ACF_m(k) = \sum_{n=0}^{N-1} x_m(n) \cdot x_m(n + k), \quad 0 \leq k < K, \quad 0 \leq m < M \]

The Auto-correlation is normalized using:

\[ NACF_m(k) = \frac{ACF_m(k)}{ACF_m(0)}, \quad 0 \leq k < K, \quad 0 \leq m < M \]

For each frequency band \( m \), confidence measure \( CM_m \) is calculated based on the auto-correlation value.

\[ CM_m = \max_{10 \leq k \leq K - 1} NACF_m(k), \quad 0 \leq m < M \]

\( N \): input frame length,
\( m \): index of frequency band
\( k \): index of lag for autocorrelation
\( K \): order of auto-correlation and is set to 128,
\( n \): index of the input audio signal,
\( M \): number of frequency bands and is set 5
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz) → Framing → Pre Emphasis

Filter bank → Auto-correlation → Integration

Integration → Peak Detection → Audio Feature Frame Rate Conversion → Audio Feature
Integration

The normalized auto-correlation function $NACF_m(k)$ values derived from each sub-band are summed together into a single integrated auto-correlation function.

$$ACF_{integrated}(k) = \frac{\sum_{m=0}^{N_b-1} NACF_m(k) \cdot W_m}{\sum_{m=0}^{N_b-1} W_m}, \quad 0 \leq k < K$$

where $W_m$ is defined as following

$$W_m = \begin{cases} 0, & CM_m < 0.3 \\ 1, & CM_m \geq 0.3 \end{cases}$$
Block Diagram of Audio Feature Extraction tool

Audio Signal (fs=8kHz) → Framing → Pre Emphasis

Filter bank → Auto-correlation → Integration → Peak Detection → Audio Feature Frame Rate Conversion → Audio Feature
Peak Detection

The integrated auto-correlation function is converted into a 128-bit length feature vector $f(k)$ and each bit position corresponds to the lag of the auto-correlation function.
Audio Feature Similarity Calculation Tool
(Informative)
Block Diagram Audio Feature Similarity Calculation Tool (Informative)

Audio Feature Sequence #1

Audio Feature Frame Rate Conversion → Audio Feature Block Extraction

Audit Feature Sequence #2

Audio Feature Frame Rate Conversion → Audio Feature Block Extraction

Block Similarity Calculation

Time difference between audio signals
Block Diagram Audio Feature Similarity Calculation Tool (Informative)

Audio Feature Sequence #1
- Audio Feature Frame Rate Conversion
- Audio Feature Block Extraction

Audio Feature Sequence #2
- Audio Feature Frame Rate Conversion
- Audio Feature Block Extraction

Block Similarity Calculation

Time difference between audio signals
Block Extraction

The blocks are generated by concatenating the consecutive audio features.

\[
A_i = (f_i, f_{i+1}, f_{i+2}, \ldots, f_{i+N_b-1}), \quad 0 \leq i < N_f - N_b
\]

\[
B_j = (g_j, g_{j+1}, g_{j+2}, \ldots, g_{j+N_b-1}), \quad 0 \leq j < N_g - N_b
\]

Block Similarity Calculation is performed between two blocks of audio features.
Block Diagram Audio Feature Similarity Calculation Tool (Informative)

Audio Feature Sequence #1
- Audio Feature Frame Rate Conversion
- Audio Feature Block Extraction

Audio Feature Sequence #2
- Audio Feature Frame Rate Conversion
- Audio Feature Block Extraction

Block Similarity Calculation

Time difference between audio signals
Block Similarity Calculation

Block Similarity between $A_i$ and $B_j$ is calculated as follows:

$$J(A_i, B_j) = \frac{|A_i \cap B_j|}{|A_i \cup B_j|}$$

Example

Block Similarity between $A_i$ and $B_j$ is calculated as follows:

$$\frac{|A_i \cap B_j|}{|A_i \cup B_j|} = \frac{4}{10} = 0.4$$
Time Difference Estimation

For each time difference $\tau$, a score is calculated by using the block similarity as follows:

$$\text{Score}(\tau) = \frac{1}{\min(N_{g'} - \tau, N_{f'}) - \max(-\tau, 0)} \sum_{i = \max(-\tau, 0)}^{\min(N_{g'} - \tau, N_{f'})} J(A_i, B_{i+\tau})$$

The time difference which has the largest score is regarded as the time difference between two audio feature sequences:

$$\text{delay} = \arg\max_{-N_{f'} \leq \tau < N_{g'}} \text{Score}(\tau)$$

The summations for each time difference is performed along the arrows.
Time Difference Estimation

Example
Performance evaluation

- Capturing the 1st screen content and additive noise sound at the 2nd screen. → The noise contaminated 1st screen content files

- Line-out of the 1st screen and the 2nd screen are captured as a single stereo wave file. → Time difference between the L-ch and the R-ch in the file is measured
### 1st Screen content and additive noise files

#### The 1st Screen Content Files

<table>
<thead>
<tr>
<th>Filename</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st_betty</td>
<td>5.1 down mix (according to ARIB STD-B32) version of CO_11_Betty3b_output</td>
</tr>
<tr>
<td>1st_speech</td>
<td>2 Speech (German Male, SQAM track 54)</td>
</tr>
<tr>
<td>1st_music</td>
<td>2 Music (Wind ensemble, SQAM track 67)</td>
</tr>
</tbody>
</table>

#### Additive Noise Sound Files

<table>
<thead>
<tr>
<th>Filename</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File4</td>
<td>noise_pinknoise</td>
</tr>
<tr>
<td>File5</td>
<td>noise_speech Speech (English Female, SQAM track 49)</td>
</tr>
<tr>
<td>File6</td>
<td>noise_music Music (Eddie Rabbitt, SQAM track 70)</td>
</tr>
<tr>
<td>Filename of 1st screen content files</td>
<td>Filename of additive noise sound files</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1st_betty</td>
<td>noise_pinknoise</td>
</tr>
<tr>
<td></td>
<td>noise_speech</td>
</tr>
<tr>
<td></td>
<td>noise_music</td>
</tr>
<tr>
<td>1st_speech</td>
<td>noise_pinknoise</td>
</tr>
<tr>
<td></td>
<td>noise_speech</td>
</tr>
<tr>
<td></td>
<td>noise_music</td>
</tr>
<tr>
<td>1st_music</td>
<td>noise_pinknoise</td>
</tr>
<tr>
<td></td>
<td>noise_speech</td>
</tr>
<tr>
<td></td>
<td>noise_music</td>
</tr>
</tbody>
</table>

The figures with orange background is approximately within 1 frame length (32ms). → Synchronization is successful!
Result (cont.)

Synchronization robustness against interference noise

Allowable SNR for synchronization measured at 2nd screen (dB)

-24 -21 -18 -15 -12 -9 -6 -3 0

Additive noise  |  pinknoise  speech  music  pinknoise  speech  music  pinknoise  speech  music
1st screen content  |  betty       |speech       |music       |speech       |music       |speech       |music       |
MPEG-4 Audio Object Type (ISO/IEC 14496-3:2009)

<table>
<thead>
<tr>
<th>Object Type ID</th>
<th>Audio Object Type</th>
<th>Gain control</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Null</td>
<td>[...]</td>
<td></td>
</tr>
<tr>
<td>[...]</td>
<td>[...]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>SAOC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>LD MPEG Surround</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>SAOC-DE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Audio Sync</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47 - 95</td>
<td>(reserved)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Demonstration

- 1st screen (blue walkman): Instrument only
- 2nd screen (my note PC): Vocal only
- Noise (white walkman): Female speech
Conclusion

• MPEG-4 Audio Synchronization standard defines:
  ✓ Audio Feature Extraction tool and syntax of the feature stream (Normative)
  ✓ Feature Similarity Calculation Tool (Informative)
  ✓ The Audio Object Type (AOT=46) “Audio Sync”

  to allow transmission of audio feature for synchronization as elementary stream

• The MPEG-4 synchronization mechanism works with highly noisy environment and proven that the scheme is useful under practical conditions.
End