Seamless Audio Splicing for ISO/IEC 13818 Transport Streams

A New Framework for Audio Elementary Stream Tailoring and Modeling

Seyfullah Halit Oguz, Ph.D. and Sorin Faibish
EMC Corporation
Media Solutions Group Engineering
EMC Media Solutions Group Profile

The EMC Media Solutions Group is part of EMC Engineering

— EMC Engineering will spend well over $1 Billion Dollars on Research and Development in FY 2001.

— Over 300 engineers concentrating on Celerra Server development and Rich Media products.

— The Media Solutions Group Lab has in excess of $300 Million dollars of hardware for development of Rich Media solutions.
Mission Statement

The EMC Media Solutions Group is tasked with:

— Deploying EMC Products into the Rich Media market.

— Develop products and solutions which meet the requirements of Rich Media customers.

— Developing partnerships with key companies to develop and deploy customer solutions.

— Provide Professional Services in the Rich Media environment.
Outline

- Splicing in brief
- Objective
- Problem description
- Basic algorithm
- A model for the audio elementary streams
- Enhanced algorithm
- Additional implementation details
- Conclusions
Splicing

Splicing is the act of switching from one MPEG-2 program (embedded in a transport stream) to another MPEG-2 program (again embedded in a transport stream).

Commercial insertion, camera or content switching and content editing all require splicing to be performed on compressed bit-streams.

The structure of the compressed data makes a seamless splicing algorithm be far from trivial.
Objective

A generic method to process the audio elementary streams during the splicing of ITU-T Rec. H.222.0 | ISO/IEC 13818-1 transport streams (TS) to achieve a seamless audio splice.

* “generic”: No constraining assumptions are made about signal formats (e.g. the video frame rate (PAL, NTSC), the audio sampling frequency), or various encoding parameters (e.g. the audio bit rate, the layer of audio encoding algorithm employed).

* “audio elementary streams”. Current focus will be on the audio elementary streams. Ultimately audio and video splicing should be considered jointly.

* “transport streams”. Achieve audio elementary stream splicing directly on transport streams with lowest possible complexity.
Definitions and Notation

Encoded Data Domain

Audio Access Unit
AAU (Audio Frame)
(576 bytes)*

Audio Presentation Unit,
APU, (a block of contiguous audio samples)
(24 ms)*

Video Access Unit
VAU (variable size)

Video Presentation Unit,
VPU, (a video frame)
(1/29.97 s)**

(*ISO/IEC 11172-3 Layer-II Audio coding with sampling frequency 48kHz and audio bitrate 192kbits/s assumed only for illustrative purposes)

(**NTSC frame rate assumed only for illustrative purposes)
The start of a VPU will be aligned with the start of an APU possibly at the beginning of a stream and then only at multiples of 5 minutes increments in time. This implies that later they will not be aligned again for all practical purposes.
The setting for splicing

Splicing point is naturally defined with respect to VPUs.

Ending stream

Beginning stream

VPU (k-2)  VPU (k-1)  VPU k  VPU (k+1)  VPU (k+2)

APU (j-2)  APU (j-1)  APU j  APU (j+1)  APU (j+2)  APU (j+3)

VPU (n-2)  VPU (n-1)  VPU n  VPU (n+1)  VPU (n+2)

APU (m-2)  APU (m-1)  APU m  APU (m+1)  APU (m+2)  APU (m+3)

time base #1
time base #2
Audio processing at splicing

APUs are available only through the decoding of their corresponding AAUs. Fractional (i.e. truncated) AAUs in the encoded data domain are useless.

Time base of the beginning stream is shifted to achieve video presentation continuity.
So far...

- Decoding, time domain editing and re-encoding. High computational complexity.
- Gaps in the audio stream. Audio mutes, uncontrolled audio-visual skew.
- Overlaps in the scopes of APUs. Uncontrolled audio-visual skew, inconsistent ES structure.
Observations

- Audio truncation should always be done at AAU boundaries i.e. no fractional AAUs!
- Audio truncation for the ending stream should be done with respect to the end of its last VPU’s presentation interval.
- Audio truncation for the beginning stream should be done relative to the beginning of its first VPU’s presentation interval.

“BEST ALIGNED APUs”
Best aligned APUs

The APU of the ending stream whose presentation interval ends within the identified 24 ms interval is called the “best aligned final APU”.

The APU of the beginning stream whose presentation interval starts within the identified 24 ms interval is called the “best aligned initial APU”.

There is a comprehensive list of 8 possible cases that can be identified regarding the alignment of ending and beginning audio streams based on the above definitions.
How to make use of best aligned APUs

**ACTION**

- Truncate the ending audio stream at the end of the best aligned final APU.

- Start the beginning audio stream at the beginning of the best aligned initial APU.

- Re-stamp the audio PTSs of the beginning stream to generate an immediate continuation of the ending audio stream.

**REQUIRED PROCESSING AT ELEMENTARY STREAM LEVEL**

- In the audio PES packet carrying the best aligned final APU
  - truncate after the AAU associated with the best aligned final APU,
  - modify the PES packet size information accordingly.

- In the audio PES packet carrying the best aligned initial APU
  - delete the AAU data preceding the AAU associated with the best aligned initial APU,
  - modify the PES packet size information accordingly.

- Modify the PTS values associated with the first and all consequent audio PES packets accordingly.
Case 6) b. a. final APU long, b. a. initial APU short and 0 msec. < audio overlap < 12 msec.

Best aligned initial APU

SOLUTION:

A/V skew of at most 12 msec.
Minimal Achievable Skew Algorithm

- Immediately applicable to 6 out of the 8 possible best aligned APU relative position classes.

- In the remaining 2 classes of relative position, a slight modification to the proposed algorithm is needed to achieve an A/V skew bounded by half APU duration.
Case 1) Both best aligned APUs are short and 12 msec. < audio gap < 24 msec.

Best aligned initial APU

A/V skew of at most 12 msec.

SOLUTION (a):

SOLUTION (b):
An audio elementary stream construction with no holes and no audio PTS discontinuity is possible.

As a consequence, an A/V skew of magnitude at most half APU duration will be induced in the beginning stream. This is below the sensitivity limits of human perception.

The proposed algorithm can be repeatedly applied an arbitrary number of times with neither a failure to meet its structural assumptions nor a degradation in its promised A/V skew performance.
Facts - II

- The A/V skews induced as a result of the proposed processing do not accumulate, i.e. irrespective of the number of consecutive splices the worst A/V skew at any point in time will be half of the APU duration.

- At each splice point, at the termination of the PUs of the ending stream, we always have total audio and video presentation durations up to that point almost matching each other, i.e.

\[|\text{video\_duration} - \text{audio\_duration}| \leq (1/2) \text{ APU\_duration}\]

i.e. always correct amount of audio data provided.
Facts - III

- The resulting audio stream is error-free and totally ISO/IEC 11172-3 and ISO/IEC 13818-3 compliant.
Implementation at TS level

- TS level implementation necessitates further considerations:
  1) Editing of transport packets
  2) Transport packet buffering and re-multiplexing
  3) Audio buffer management
  4) Meta-data generation
Audio buffer management

- We have to control the dynamic behavior of the audio buffer during the transient induced by the splicing as well as consequent to splicing.

- We need a simple model to characterize the audio elementary stream within a TS, such that
  1) model parameters should be easy to estimate,
  2) model should accurately provide the following desired information
     a) the mean audio buffer fullness,
     b) the extent of buffer fullness variation around its mean value.
Audio elementary stream model

Audio elementary stream will be modeled on the basis of AAUs which determines its granularity with respect to audio decoder actions.

The proposed framework for the model is an arrival process to a FIFO queue with a deterministic service time.

\[ a_j : \text{arrival time of } j^{th} \text{ AAU} \]
\[ p_j : \text{presentation time of } j^{th} \text{ AAU} \]

service time 0.024 sec.
Audio elementary stream model

- The presentation times $p_j$ can be easily computed for each AAU.
- The arrival times $a_j$ however are not uniquely defined since each AAU arrives in a distributed fashion owing to transport packet encapsulation.

Solution:

Use “weighted-average arrival times”.
Audio elementary stream model

Weighted-average arrival time is defined for each AAU as follows:

\[ a_j = \sum_{\text{all TS packets whose } \text{payloads have some of the } j^{th} \text{ AAU data}} (\text{TS packet arrival time}) * (\text{fraction of the } j^{th} \text{ AAU in the payload data}) \]

- Note: TS packet arrival time is defined with respect to packet’s 94th byte through a PCR extrapolation process.
Audio elementary stream model

Based on these definitions we let

\[ w_j = p_j - a_j \]

where \( w_j \) is the waiting time for \( j^{th} \) AAU in the audio buffer.

The mean and variance of \( w_j \) as a random variable \( w \), provide very important information about the audio elementary stream within the TS multiplex and hence the audio buffer.
Audio elementary stream model

Specifically,

$$E[w].(\text{audio bitrate}) = \text{mean audio buffer fullness}$$

$$\sigma_w.(\text{audio bitrate}) = \text{a measure of the variation in the audio buffer fullness around its mean value}$$

$E[.]$: expectation operation.
$\sigma$: standard deviation.
Audio elementary stream model

Example 1.
Encoder B characteristics:
- Long mean waiting time
- Highly irregular, bursty arrival regime
Example 1

**TS PACKET TYPE VISUALIZATION PLOT**

- Interpolated play-out times for individual TS packets, [seconds]

**INTERLEAVED AUDIO-VIDEO ALIGNMENT IN THE BITSTREAM WRT PTS VALUES**

- TS packet indices
- TS packet types (color coded)
Example 1

Predicted mean audio buffer fullness: 15585.8 bits = 54.36%.
$\pm$2 STD interval around the mean: $\pm$ 4504.2 bits = $\pm$15.71%.
An important observation

- PTS re-stamping in the *beginning* audio stream
  - The waiting times of AAUs will be modified in the *beginning* stream.

- The mean waiting time of AAUs in the *beginning* stream will change (decrease or increase) by at most half APU duration.

- A corresponding change in the mean audio buffer fullness level for the *beginning* stream will be induced.
Improvement to the proposed processing

- For audio elementary streams structured with mean audio buffer fullness level bounded away from both underflows and overflows, the already proposed methodology with the minimal achievable A/V skew is the best to choose.

- For audio elementary streams with mean audio buffer fullness level close to either underflow or overflow states, the requirement of minimal resultant A/V skew should be relaxed and instead of “best aligned APUs” those APUs introducing the minimal safe A/V skew should be employed.
Outline of the improvement - I

- Let the type of A/V skew in which the audio signal component is delayed with respect to its associated video signal, be referred to as the **forward (in time) skew**.

- Forward skew is a result of a uniform increment in the **beginning stream’s** AAU presentation time stamps and is therefore associated with an increase in the mean audio buffer fullness level.

- Hence, for **beginning** audio elementary streams with mean audio buffer fullness levels close to underflow, the minimal achievable forward skew is the **minimal safe skew** and should be the one to be employed.
Outline of the improvement - II

Let the type of A/V skew in which the audio signal component moves to earlier in time with respect to its associated video signal, be referred to as the **backward (in time) skew**.

Backward skew is a result of a uniform decrement in the **beginning stream**’s AAU presentation time stamps and is therefore associated with a decrease in the mean audio buffer fullness level.

Hence, for **beginning** audio elementary streams with mean audio buffer fullness levels close to overflow, the minimal achievable backward skew is the minimal safe skew and should be the one to be employed.
Minimal Safe Skew Algorithm Summary

For all 8 possible relative position classes of best aligned APUs, three solutions are defined:

1. the minimal achievable skew solution
2. the minimal safe forward skew solution
3. the minimal safe backward skew solution

- Minimal achievable skew is upper-bounded by half APU duration.
- Minimal safe skew (forward or backward) is upper bounded by one APU duration. (Still below the sensitivity limits of human perception.)
Case 1) Both best aligned APUs are short and 12 msec. < audio gap < 24 msec.

Best aligned initial APU

Best aligned final APU
VPU (k-1) VPU k VPU (k+1) VPU (k+2) 

APU (j+1) APU (j+2) APU (j+3) 

A/V skew of at most 12 msec.

SOLUTION I: 

Minimal achievable skew implementation for clips with originally moderate audio buffer levels. 
Note: An alternate implementation by omitting APU (j+2) and including APU (m-1) is also possible and this approach achieves exactly the same resultant A/V skew. This latter solution is preferable in terms of ease of implementation.

SOLUTION II: 

VPU (k-1) VPU k VPU (k+1) VPU (k+2) 

APU (j+1) APU (j+2) APU (j+3) 

A/V skew of at most 12 msec.

SOLUTION II: 

VPU (k-1) VPU k VPU (k+1) VPU (k+2) 

APU (j+1) APU (j+2) APU (j+3) 

A/V skew of at most 24 msec.

SOLUTION III: 

VPU (k-1) VPU k VPU (k+1) VPU (k+2) 

Minimal achievable safe (backward) skew implementation for clips with originally high audio buffer levels.

Note: An alternate implementation by omitting APU (j+2) and including APU (m-1) is also possible and this approach achieves exactly the same resultant A/V skew. This latter solution is preferable in terms of ease of implementation.
Case 6) b. a. final APU long, b. a. initial APU short and 0 msec. < audio overlap < 12 msec.
SOLUTION I:

VPU (k-1) VPU k VPU (k+1) VPU (k+2)

A/V skew of at most 12 msec.

Minimal achievable skew implementation for clips with originally moderate audio buffer levels.

SOLUTION II:

VPU (k-1) VPU k VPU (k+1) VPU (k+2)

A/V skew of at most 12 msec.

Minimal achievable safe (forward) skew implementation for clips with originally low audio buffer levels.

SOLUTION III:

VPU (k-1) VPU k VPU (k+1) VPU (k+2)

A/V skew of at most 24 msec.

Minimal achievable safe (backward) skew implementation for clips with originally high audio buffer levels.

Note: An alternate implementation by omitting APU (j+1) and including APU m is also possible and this approach achieves exactly the same resultant A/V skew. This latter solution is preferable in terms of ease of implementation.
Implementation at TS level

TS level implementation necessitates further considerations:

1) Editing of transport packets
2) Transport packet buffering and re-multiplexing
3) Audio buffer management
4) Metadata generation
Editing of transport packets

- The truncation of the final PES packet of the ending audio stream will typically necessitate the insertion of some adaptation field padding into its last transport packet.

- The deletion of some AAU data from the front end of the beginning audio stream’s first PES packet will typically necessitate the editing of at most two audio transport packets.

- Possible use of a causal bit-reservoir in Layer III encoding, typically dictates a structural constraint on the beginning audio stream.
Transport packet buffering and re-multiplexing

In the transport stream
- the audio bit rate is a constant
- the VAUs are of varying sizes
therefore

the relative positions of VAUs and AAUs associated with VPU and APU almost aligned in time cannot be maintained constant.

Almost always it is the case that the AAUs are significantly delayed with respect to the VAUs for which the decoded representations are almost synchronous.
Legend

- **Black**: TS video packets initiating I type pictures.
- **Blue**: TS video packets initiating P or B type pictures.
- **Cyan**: TS video packets.
- **Red**: TS audio packets initiating PES packets.
- **Magenta**: TS audio packets,
- **Yellow**: Null TS packets.
- **Green**: TS packets carrying various system information.
Example 1. Encoder B.
Transport packet buffering and re-multiplexing

This TS multiplex structure necessitates

1) locating and temporarily storing (buffering) the delayed audio packets when the ending stream is truncated based on the last VAU,

2) TS re-multiplexing in the form of
   
   a) deletion of some obsolete green audio packets in the beginning stream,
   
   b) insertion of the above blue audio packets into the beginning stream.
During the ingest of MPEG-2 transport streams to storage systems,

1. estimate the parameters of the proposed audio elementary stream model,

2. record this descriptive information within the metadata associated with the asset.

In splicing scenarios with live streams, known characteristics and/or settings of the encoder employed can be used.
Conclusions

- Audio elementary stream tailoring based on **minimal achievable** or **minimal safe** A/V skew concepts.
- Audio splicing **without any artifacts** made possible.
- A simple and efficient model to characterize the audio elementary stream structure embedded in the transport stream. (Prediction and possible control of audio buffer behavior within reach.)
- Audio splicing **cannot** be considered independently from the video splicing.