SMPTE ST 2064-1:2015

SMPTE STANDARD

Audio to Video Synchronization Measurement — Fingerprint Generation

Page 1 of 23 pages

Table of Contents

Page

Foreword 2

Intellectual Property 2

Introduction 2

1 Scope 4

2 Conformance Notation 4

3 Normative References 4

4 Definitions and Terminology 5

5 Fingerprint Generation 5

5.1 Audio and Video Fingerprint Generation 5

5.2 Video Fingerprint Generation 6

5.3 Audio Fingerprint Generation 10

6 Encapsulation of Fingerprints 14

6.1 Container Structure 15

6.2 ID Sub-container Structure 16

6.3 Video Fingerprint Sub-container Structure 17

6.4 Audio Fingerprint Sub-container Structure 17

7 Sample Fingerprint Transport Packets (Informative) 21

Annex A Bibliography (Informative) 23

3 Barker Avenue, White Plains, NY 10601
(914) 761-1100

Approved
October 9, 2015

Foreword

The Society of Motion Picture and Television Engineers (SMPTE) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices, and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in the Standards Operations Manual.

SMPTE ST 2064-1 was prepared by Technology Committee 24TB.

Intellectual Property

SMPTE draws attention to the fact that it is claimed that compliance with this Standard may involve the use of one or more patents or other intellectual property rights (collectively, "IPR"). The Society takes no position concerning the evidence, validity, or scope of this IPR.

Each holder of claimed IPR has assured the Society that it is willing to License all IPR it owns, and any third party IPR it has the right to sublicense, that is essential to the implementation of this Standard to those (Members and non-Members alike) desiring to implement this Standard under reasonable terms and conditions, demonstrably free of discrimination. Each holder of claimed IPR has filed a statement to such effect with SMPTE. Information may be obtained from the Director, Standards & Engineering at SMPTE Headquarters.

Attention is also drawn to the possibility that elements of this Standard may be subject to IPR other than those identified above. The Society shall not be responsible for identifying any or all such IPR.

Introduction

This section is entirely informative and does not form an integral part of this Engineering Document.

Errors in audio to video timing relationships have become commonplace in the industry. Such errors are known as “lip-sync” errors because of a typical viewer’s sensitivity to inaccuracy in the synchronization of lip movement with the sound of speech. This situation arises because the audio and video portions of a program follow different processing paths, each with its own inherent timing factors. For various reasons, the audio and video delays through the program distribution chain may change dynamically. In order to correct such timing errors as they occur, a method is required to measure the change in audio to video synchronization without relying on out-of-service test signals.

The SMPTE 2064 suite of documents for Audio to Video Synchronization Measurements defines a process for extracting, packetizing, and transporting compact representations of audio and video essence, known as video and audio fingerprints, which change constantly as a function of changing picture and sound content. These fingerprints are extracted by non-intrusive analysis of the audio and video essence, and therefore can be used in a live on-air situation as well as in non-real-time systems. The fingerprints generated at a reference point where synchronization is known to be correct are intended to be transported through the program distribution chain, either bound to or separate from the associated essence. Where measurement and correction of synchronization errors is needed, a new set of fingerprints is extracted from the essence at a downstream location and compared with the fingerprints from the reference point. This comparison provides a dynamic measurement of the audio to video timing changes that have occurred, which may be used by other processes to display and/or correct any synchronization errors.

The timing relationship of the audio and video signals within a program is described from the perspective of the video signal. When the audio signal precedes, or arrives earlier in time than the video signal the measured value is stated as a negative value. When the audio signal arrives later in time than the video signal it is stated as a positive value. This measurement and comparison is outside the scope of this document.

The SMPTE ST 2064-1 standard specifies the method for generating the audio and video fingerprints, and the SMPTE ST 2064-2 standard specifies the carriage of fingerprints using various transport methods.

1 Scope

This standard defines algorithms and procedures for generating audio and video fingerprints from audio and video essence used for audio to video timing measurements. It also specifies a method for combining the audio and video fingerprints and associated metadata into a container suitable for transport. Composite video formats are not supported by this standard.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:”

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 274:2008) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 274M-2008). Documents with the same root number (e.g. 274) and publication year (e.g. 2008) are functionally identical.

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 125:2013, SDTV Component Video Signal Coding 4:4:4 and 4:2:2 for 13.5 MHz and 18 MHz Systems

SMPTE ST 274:2008, Television — 1920 × 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates

SMPTE ST 296:2012, 1280 × 720 Progressive Image 4:2:2 and 4:4:4 Sample Structure — Analog and Digital Representation and Analog Interface

SMPTE ST 352:2013, Payload Identification Codes for Serial Digital Interfaces

SMPTE ST 2036-1:2014, Ultra High Definition Television – Image Parameter Values for Program Production

SMPTE ST 2048-1:2011, 2048 × 1080 and 4096 × 2160 Digital Cinematography Production Image Formats FS/709

Recommendation ITU-R-BS.775-3 (08/2012), Multichannel Stereophonic Sound System with and without Accompanying Picture

4 Definitions and Terminology

4.1 Fingerprints

The term "fingerprint", as applied to audio and video signals, refers generally to a computed representation of key features of the image and sound being transported in the signals, with the computation specified so that the representation uniquely identifies the contents. To be practical, the fingerprint must be many orders of magnitude smaller than the signal it represents, while at the same time maintaining a very high probability of unique identification. Being content oriented, it must be resistant to many of the processes that the content may undergo, such as scaling and format conversion.

For the purposes of this document, the term "Video Fingerprint" refers to the values computed according to the processes described in Section 5.2, Video Fingerprint Generation.

For the purposes of this document, the term "Audio Fingerprint" refers to the values computed according to the processes described in Section 5.3, Audio Fingerprint Generation.

4.2 Reserved Bits

5 Fingerprint Generation

5.1 Audio and Video Fingerprint Generation

This section defines the method of generating audio and video fingerprints of an audio and/or video essence or signal.

These fingerprints can be generated at different points in the signal chain in order to enable measurement of the Audio to Video (A/V) lip-sync error at one or more points. An example of the fingerprint generation and analysis is shown in Figure 1.

General signal processing for fingerprint generation comprises multiple functions appropriate for video and audio, respectively. A reference function for each is described in the next sections. Other methods that achieve identical results at the points of interchange may be used.

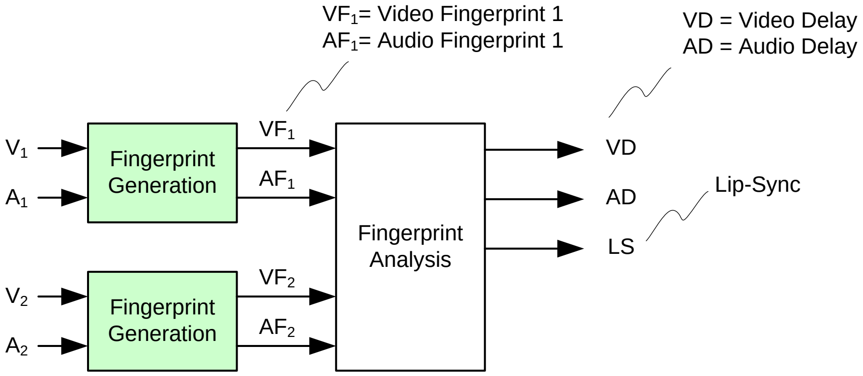


Figure 1 – Example of Lip-Sync Error Measurement System

Note: The method for fingerprint analysis is not defined in this document and can vary depending on the implementation.

5.2 Video Fingerprint Generation

Video fingerprint generation shall be based on evaluation of the amount of video content change (typically due to motion) during the time interval between two video frames or fields.

Fields (for interlaced video) or frames (for progressive video) shall be processed sequentially.

The current field/frame shall be compared with the second preceding field/frame to calculate a difference used for further processing. Only the 8 most significant bits of the luminance samples shall be used in the calculation.

General video signal processing for fingerprint generation shall comprise three major functions — prefiltering, windowing and sub-sampling, and motion detection — as illustrated in Figure 2. These functions are described in the following subsections.

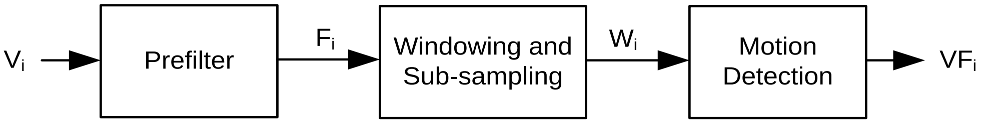


Figure 2 – Video Fingerprint Generation

5.2.1 Prefilter

Prior to sampling, the video shall be filtered to reduce its bandwidth and to facilitate consistent results with different video formats. To reduce the implementation resources required, a simple mean calculation shall be used only on the horizontal axis and shall be applied only to the luminance samples. Table 1 shows the filter used for different video formats.

Table 1 – Video Format Prefilter

Video Format	Video Standard	Prefilter Used
4096 X 2160p	ST 2048-1	[1 1 1 1 1 1] / 6
3840 X 2160p	ST 2036-1	[1 1 1 1 1 1] / 6
2048 X 1080p	ST 2048-1	[1 1 1] / 3
1080i, 1080p	ST 274	[1 1 1] / 3
720p	ST 296	[1 1 0] / 2
SD 525, SD 625	ST 125	[0 1 0] / 1

For 2160-line formats, the filter shall use the three previous, current and two next pixels. For 1080-line formats, the filter shall use the previous, current and next pixels. For 720p formats the previous and current pixels shall be used. For SD formats, no filtering shall be done.

Note: This filtering method results in an image that is not optimized for viewing but is suitable for fingerprint generation. The defined reduction in spatial resolution does not reduce the number of pixels in the image; such a reduction occurs in the windowing process.

5.2.2 Windowing and Sub-Sampling

When comparing fingerprints derived from two video signals, originating from the same picture content, it is possible that one signal has been altered compared to the other. Possible alterations could include such things as branding, graphic overlays and aspect ratio change. These alterations will decrease the effectiveness of signal matching. Accordingly, a window is defined to focus the fingerprint generation on the central area of the image and reduce the impact of such possible alterations to the video. The number of pixels in the window also is reduced by a sub-sampling process. It is recognized that the effectiveness of the window measurement may vary depending on the nature of the signals to be compared.

A windowing block shall be used to select a part of the image from which the video fingerprint is extracted. The pixel coordinates of the window depend on the video format of the signal in use at the time of fingerprint generation and shall be as shown in Table 2. A subset of pixels inside the defined window that is derived by sub-sampling shall be used for fingerprint generation, and pixels outside the window shall be ignored. The subset of pixels shall consist of 16 sample rows of 60 pixel samples, evenly spaced horizontally and vertically across the selected window. This yields a total of 960 pixels that are used in the motion detection process for fingerprint generation.

The process of determining the pixels to be used is illustrated in the following example for 720p, with a window sampled from a picture of 1280 pixels by 720 lines. Figure 3 shows an example of how to determine the lines and selected pixels to be compared:

> According to Table 2, the first horizontal pixel position to use is pixel 256. Since the pixel step is 13, the next pixels will be 269, 282, … up to pixel 1023, which is the 60th selected pixel on the line. Vertically, the first row of pixels to compare will be on line 117. Since the line step is 32, the next lines will be 149, 181, … up to line 597, which is the 16th selected lines. This is depicted in Figure 3.

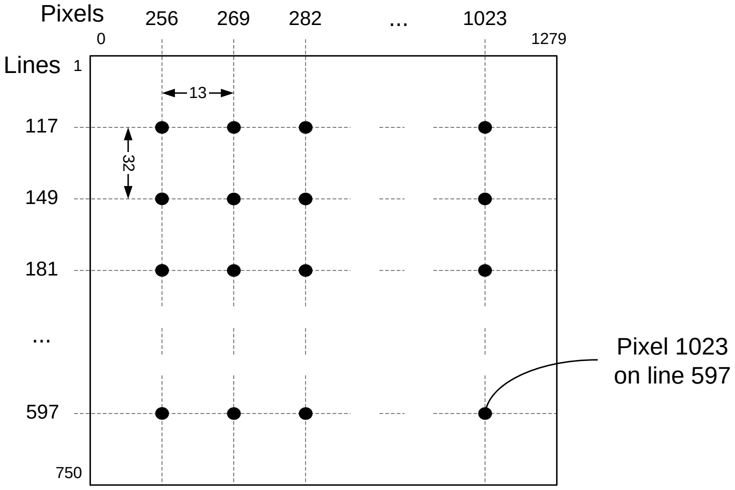


Figure 3 – Pixels used for comparing video frames in 720p

Table 2 – **Window Coordinates per Video Format**

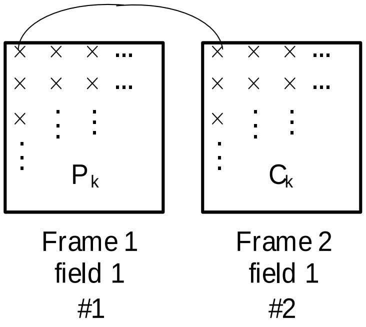
Video Format	Window							
	HStart	HStep	HStop	VStart (f1)	VStart (f2)	VStep	VStop (f1)	VStop (f2)
720 X 485i	123	8	595	60	323	10	210	473
720 X 576i	123	8	595	68	381	12	248	561
1280 X 720p	256	13	1023	117	–	32	597	–
1920 X 1080i	399	19	1520	89	652	24	449	1012
1920 X 1080p	399	19	1520	178	–	48	898	–
3840 X 2160p	798	38	3040	412	–	92	1792	–
2048 X 1080p	463	19	1584	206	–	46	896	–
4096 X 2160p	926	38	3168	412	–	92	1792	–

Note: Values for field 1 (f1) and field 2 (f2) are shown separately.

5.2.3 Motion Detection

The motion detection block shall calculate the amount of change in the video content between the current field/frame and the content of a prior field/frame. The difference in the video content between the current and a prior video field/frame shall be used to calculate a video fingerprint. The motion detection process, illustrated in Figure 4, shall compare pixels within the current field/frame C_k with pixels within a prior field/frame P_k to determine the amount of change between them.

Only the 8 most significant bits of the luminance samples shall be used in the calculation.



$$VS_i(f) = \frac{\sum_{k=1}^{N} \left(\mathrm{abs}(P_k - C_k) \;\middle|\; \begin{matrix} 1 \text{ (if >= 32)} \\ 0 \text{ otherwise} \end{matrix} \right)}{4}$$

Figure 4 – Formula showing how the pixels are compared and counted

5.2.3.1 Pixel Compare

In order to make fingerprints compatible between interlaced and progressive forms of the same content, the following comparisons are made:

For progressive video, comparison shall be made between the current frame and the frame two frames before the current frame.

> For example: Assuming a sequence of frame F1 F2 F3 F4 F5. If F4 is the current frame, the pixel comparison would be with frame F2. If F5 is the current frame, the pixel comparison would be with frame F3.

For interlaced video, comparison shall be made between the current field and the same field from the immediately preceding frame.

> For example: Assuming a sequence of fields: f1 f2 f3 f4 f5. If f4 is the current field, the pixel comparison would be with field f2. If f5 is the current field, the pixel comparison would be with field f3.

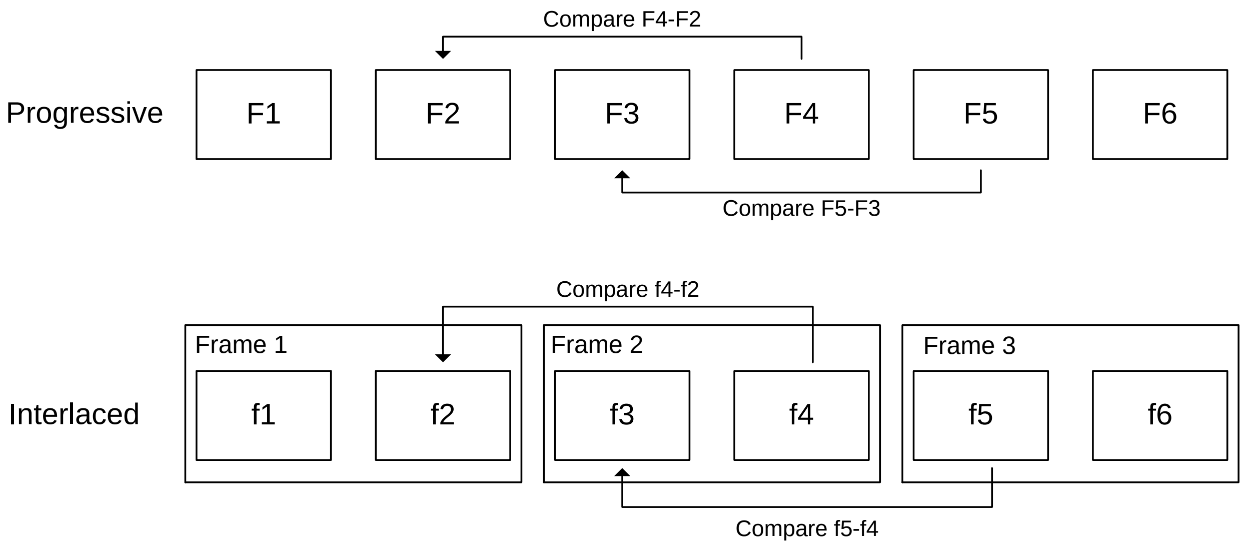


Figure 5 – Inter-Frame/Field Comparisons in Progressive and Interlaced Video

5.2.3.2 Pixel Counting

Pixel counting shall be done to establish the number of pixels, within the 960 pixels being monitored, that have changed between fields for interlaced video or between frames for progressive video. As shown in Figure 4, a pixel shall be considered changed if the difference between the current pixel and the same pixel in the previous field/frame is equal to or greater than 32 when using 8-bit video samples. For 10-bit video, the last two bits shall be ignored and the comparison shall be performed on the 8 most significant bits.

After counting how many pixels (out of 960) have changed compared to the previous field/frame, the count of changed pixels shall be divided by 4 to obtain a result varying from 0 to 240. This result represented as a byte value shall be the Video Fingerprint Data for this field/frame and is the point of interchange for video fingerprint generation methods not conforming to this documents methodology.

5.3 Audio Fingerprint Generation

An audio fingerprint is generated from a decimated comparison between the envelope and the mean of the absolute value of an audio signal over time as shown in Figure 6. The input to the audio fingerprint generation sub-system shall be Pulse Code Modulation (PCM) audio with a sample rate of 48 kHz. The audio processing for fingerprint generation shall be done on 16-bit audio samples. When processing higher bit depth audio only the 16 most significant bits shall be used.

General audio signal processing for fingerprint generation shall comprise five major functions — downmixing (when required), absolute value, two forms of detection (mean and envelope), comparison between detected data forms, and decimation — as illustrated in Figure 6. These functions are described in the following subsections.

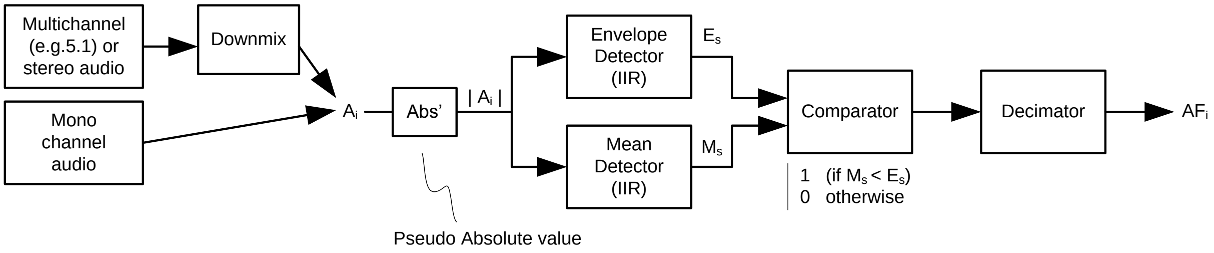


Figure 6 – Audio Fingerprint Generation

5.3.1 Downmix

Audio fingerprinting shall be performed on a single, monaural audio channel. In a multichannel (e.g. 5.1) or stereo audio program environment, a downmix to mono shall be used to generate the single, monaural audio channel to fingerprint. If fingerprints are required for individual audio source or other channels, they may also be generated.

The following formulae shall be used to downmix from multichannel to a single channel for fingerprint generation as documented in Recommendation ITU-R-BS.775-3.

From stereo (2/0):
Mono = (0.7071*L + 0.7071*R)/2

From 5.1 (3/2):
Mono = (0.7071*L + 0.7071*R + 1.0000*C + 0.5000*Ls + 0.5000*Rs)/4

5.3.2 Pseudo Absolute Value

The first step of the processing shall consist of determining the pseudo absolute value of the audio. A real absolute value | Ai | returns Ai for positive values of Ai and (-1 * Ai) for negative values of Ai.

To simplify hardware implementation, the 16 bits of the audio sample shall be inverted (one’s complement) if the most significant bit is ‘1’ and shall not be inverted if the most significant bit is ’0'.

Note: Mathematically, this will create an error of one sample value for negative values; this is considered to be insignificant for this application. For example, if Ai is 1000, the function will return 1000. If Ai is negative 1000, the function will return 999.

5.3.3 Envelope Detector

The audio envelope detection shall be performed using a single tap Infinite Impulse Response (IIR) filter as shown in Figure 7:

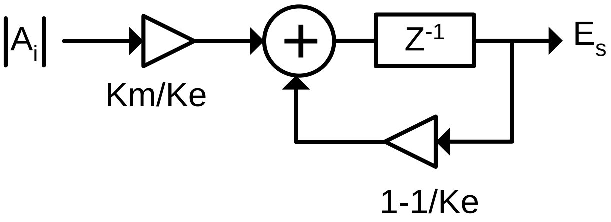


Figure 7 – IIR Filter Used for Audio Envelope Detection

The envelope detection function can be expressed in C code as follows:

```
max_sample  = Total number of samples to be fingerprinted
a_wav =  Pseudo absolute value of original unfiltered audio (per Section 5.3.2)
Km = Local mean IIR filter coefficient
Ke = Envelope detector IIR filter coefficient
Es = Envelope signal

// Envelope detector IIR filter
Km = 8192;             // local mean detector IIR filter coefficient
Ke = 1024;             // envelope detector IIR filter coefficient
Es [0] = 0;            // initialize first value to a known state
for ( i = 1; i < max_sample;  i++)
{
   Es [i] = (a_wav[i] * Km / Ke) + Es [i-1] - floor(Es [i -1] / Ke);
}
```

Ke shall be set to 1024 which is small enough to reproduce the audio envelope. This type of filter contains a single “memory” element (Z) which should be initialized to zero.

5.3.4 Local Mean Detector

The local mean detector shall be performed using a single tap IIR filter as described in Figure 8:

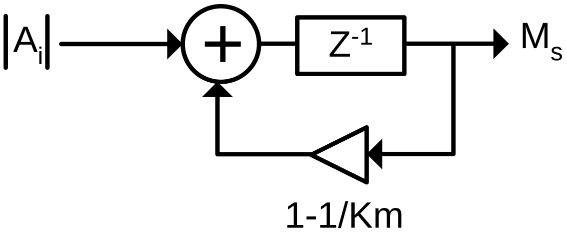


Figure 8 – IIR filter used for audio mean detection

The local mean detection function can be expressed in C code as follows:

```
max_sample  = Total number of samples to be fingerprinted
a_wav = Pseudo absolute value of original unfiltered audio (per Section 5.3.2)
Km = Local mean IIR filter coefficient
Ms = Mean signal
```

```
// Local mean IIR filter
Km = 8192;              // local mean detector IIR filter coefficient
Ms[0] = 0;              // initialize first value to a known state

for ( i = 1; i < max_sample;  i++)
{
    Ms[i] = a_wav[i] + Ms[i-1] - floor(Ms[i-1] / Km);
}
```

Km shall be set to 8192, which is large enough to emulate a local mean function. This type of filter contains a single “memory” element (Z), which should be initialized to zero.

5.3.5 Envelope/Mean Comparator

The envelope shall be compared with the mean. If the mean is lower in value than the envelope, the fingerprint bit shall be a ‘1’, otherwise it shall be a ‘0’. The result shall be a stream of ‘0’ and ‘1’ bits forming the audio fingerprint.

Operation of the envelope/mean comparator can be expressed in C code as follows:

```
// extract fingerprint by comparing envelope with local mean
for ( i=0;  i < max_sample; i++)
{
   if (Ms[i] <  Es[i])
      comp_bit[i]=1;
   else
       comp_bit[i]=0;
}
```

5.3.6 Decimator

The decimator is used to reduce the amount of data while preserving a lip-sync error detection resolution of approximately 1 msec.

The following algorithm expressed in C code shall be used to decimate the envelope and the mean.

```
// decimate envelope/mean comparison
For ( i = 0; i < max_sample; i += decimator_factor)
{
        result[i / decimator_factor] = comp_bit[i];
}
```

This algorithm produces one result bit for each decimator loop cycle. Due to the tolerance of the algorithm, which bits are kept or dropped will not significantly change the fingerprint.

For transport or storage of the result bits to form an audio fingerprint, the bits shall be accumulated in 8-bit bytes beginning with bit 0 and progressing to bit 7.

The decimator factors shown in Table 3 shall be used to ensure the proper number of bits for each video format.

The decimator factor, establishes the number of audio fingerprint bits/bytes for each group of video frames. Table 3 specifies the decimator factors for different video formats. The distribution of audio fingerprint bytes is specified in Table 13. These bytes represent the Audio Fingerprint Data and this is the point of interchange for audio fingerprint generation methods not conforming to this documents methodology.

Table 3 – Decimator Factors for Each Video Frame Rate

Video Frame Rate	Decimator factor	Bytes per x frames	Bitrate per Second
24/1.001 fps	52	77 per 16 frames	~923 b/s
30/1.001 fps	52	77 per 20 frames	~923 b/s
48/1.001	52	77 per 32 frames	~923 b/s
60/1.001 fps	52	77 per 40 frames	~923 b/s
24 fps	50	80 per 16 frames	960 b/s
25 fps	50	96 per 20 frames	960 b/s
30 fps	50	80 per 20 frames	960 b/s
48 fps	50	80 per 32 frames	960 b/s
50 fps	50	96 per 40 frames	960 b/s
60 fps	50	80 per 40 frames	960 b/s

The Bitrate per Second column in Table 3 shows the resulting bitrates of the fingerprint signal.

For example, audio related to a video frame rate of 30/1.001 produces 77 bytes of audio fingerprint data every 20 video frames with a resulting bitrate of approximately 923 bits per second.

6 Encapsulation of Fingerprints

As described in the following subsections audio and video fingerprints and associated metadata shall be combined into a container suitable for transport as defined in other documents. Fingerprint containers shall be created continuously (at the video frame rate) even when fingerprint information is not available.

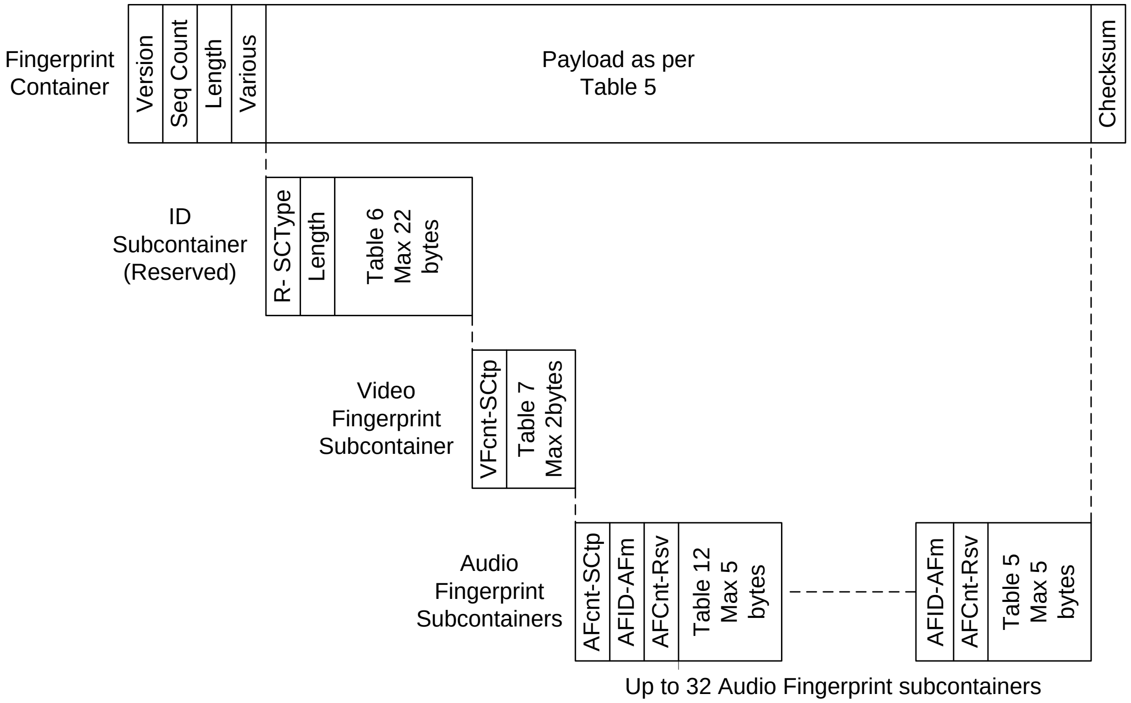


Figure 9 – Example of Encapsulation of Fingerprints

6.1 Container Structure

The container shall be as shown in Table 5. The following sub-containers may be present.

- Identification (ID) sub-container.
- The video fingerprint sub-container.
- The audio fingerprint sub-container.

Each sub-container contains a 3-bit field called SCType which indicates the type of sub-container data as shown in Table 4.

Table 4 – SCType

Value	SCType
0h	ID Sub-container
1h	Video Fingerprint Sub-container
2h	Audio Fingerprint Sub-container
3h - 7h	Reserved for future use by SMPTE

Table 5 – Container Structure

	Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Transport Header	1	FP_protocol_version							
	2	Sequence_Counter: 8 bit wrapping binary counter							
	3	Length							
	4	Picture_Rate				Reserved	IDPresent Flag	VFpPresentFlag	AFpPresentFlag
ID	n	If (IDPresentFlag == ‘1h’) ID Sub-container (See Section 6.2)							
Video Fingerprint	n	If (VFpPresentFlag == ‘1h’) Video Fingerprint Sub-container (See Section 6.3)							
Audio Fingerprint	n	If (AFpPresentFlag == ‘1h’) Audio Fingerprint Sub-container (See Section 6.4)							
Checksum	Last Byte	8-bit Checksum							

Transport Header Byte 1:

FP_protocol_version – This 8-bit field identifies the protocol version used in forming the container. It enables future versions of this protocol to carry parameters that may be structured differently from those defined in the current protocol. For containers conforming to this version of the standard, the value for the FP_protocol_version field shall be 0h.

Transport Header Byte 2:

Sequence_Counter: The sequence counter is an 8-bit zero based binary counter incrementing by one count with each fingerprint container. The sequence counter wraps around to zero (0) after its maximum value.

Transport Header Byte 3:

Length: This 1-byte field contains the length of the audio and video fingerprint container from the start of the FP_protocol_version field to the end of the Checksum field (inclusive).

Transport Header Byte 4:

Picture_Rate – This 4-bit field shall indicate the frame rate using the values as specified in the SMPTE ST 352 Picture Rate table. If the Picture Rate data from the incoming signal is not available, the value shall be derived from the current picture rate.

IDPresentFlag – This 1-bit field, when set to '1', shall indicate that the ID Sub-container is present. When set to '0', it shall indicate that no ID sub-container is present. In this version of the standard the IDPresentFlag shall be set to '0'.

VFpPresentFlag – This 1-bit field, when set to '1', shall indicate the presence of the video sub-container. When set to '0', it shall indicate that no video sub-container is present

AFpPresentFlag – This 1-bit field, when set to '1', shall indicate the presence of the audio sub-container. When set to '0', it shall indicate that no audio sub-container is present.

Container Last Byte:

Checksum – An 8-bit checksum shall be determined by calculating the arithmetic sum of all bytes from Byte 1 of the Transport Header up to the byte before the Last Byte (checksum byte) and placing the two's complement of the result (complement of the result plus one) in the checksum byte. To check for errors, the arithmetic sum can be computed over the same set of bytes, including the Checksum byte. If all bits of the result (modulo 256) are zero, the check succeeds.

6.2 ID Sub-container Structure

The ID Sub-container is divided into two sections. The first section is the header, which shall indicate the sub-container type and ID Data Length. The second section shall contain the ID data payload, which is reserved for future definition by SMPTE.

Table 6 – ID Sub-container Transport Structure

	Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sub-Container Header	1	Reserved					SCType		
	2	Reserved			Length of ID Data				
Sub-Container Data	n	ID Data							

Sub-Container Header Byte 1: Sub-container header, which defines the sub-container type in SCType. This defines the type of data in the current sub-container, such as optional ID, audio or video.

SCType – The value shall be 0h for the ID sub-container.

Sub-Container Header Byte 2: Length of ID Data payload.

Sub-Container Data n Bytes: ID data payload (reserved for future definition by SMPTE). The maximum length of this **Sub-Container Data** shall be 20 bytes.

6.3 Video Fingerprint Sub-container Structure

The first section is the header which shall indicate the sub-container type and video fingerprint byte count. The second section shall contain the video fingerprint data payload.

Table 7 – Video Fingerprint Sub-container Transport Structure

	Byte Number	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Sub-Container Header	1	Reserved			VFDataCount		SCType		
Video Fingerprint Data	n	Video Fingerprint Data							

Sub-Container Header Byte 1:

VFDataCount – This 2-bit field shall indicate the byte count for the video fingerprint data present in this sub-container. It shall be set to 1h for progressive video formats or 2h for interlaced video formats.

SCType – The value shall be 1h for the video fingerprint sub-container.

Video Fingerprint Data, n Bytes:

Progressive Formats – This video fingerprint is a 1-byte field (VFDataCount = n = 1) that shall contain the video fingerprint data for the current frame.

Interlaced Formats – This video fingerprint is a 2-byte field (VFDataCount = n = 2) that shall contain the video fingerprint data for field 1 in the first byte and for field 2 in the second byte.

6.4 Audio Fingerprint Sub-container Structure

The Audio Fingerprint Sub-container is divided into multiple sections. The first section is the Sub-container header, comprising one byte that shall indicate the sub-container type and audio fingerprint count. This byte shall be present only once per audio fingerprint sub-container. The second section is the audio fingerprint header, comprising 2 bytes that shall contain the audio fingerprint ID, the audio mix type, and the audio fingerprint count. The third section shall contain the audio fingerprint data as specified by Table 13 and associated text. The second and third sections may be repeated, depending on the number of audio fingerprints present, up to a maximum of 32 audio fingerprints. The first sub-container shall contain the fingerprint from the mandatory audio downmix (Section 5.3.1).

Table 8 – Audio Fingerprint Sub-Container Structure

	Number of Bytes	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sub-container Header	1	AudioFingerprintCount					SCType		
Audio Fingerprint Header (1st)	2	AudioFingerprintID					AudioMixType		
		AFDataCount					Reserved		
Audio Fingerprint Data (1st)	Note 1	Audio Fingerprint Data							
Audio Fingerprint Header (nth)	2	Additional Audio Fingerprint Headers and Audio Fingerprint Data as specified by the AudioFingerprintCount. (Repeat as required).							
Audio Fingerprint Data (nth)	Note 1								

Note 1: The number of bytes is dependent on frame rate in use according to Table 13.

Sub-Container Header Byte 1:

AudioFingerprintCount – This 5-bit field shall indicate the number of audio fingerprints present, as shown in Table 9.

Table 9 – AudioFingerprintCount definition

Value	AudioFingerprintCount
0h	1 fingerprint
1h-1Fh	2 to 32 fingerprints

SCType – The value shall be 2h for the audio fingerprint sub-container.

Audio Fingerprint Header Byte 2:

AudioFingerprintID – This 5-bit field shall label the audio fingerprint and associated audio fingerprint data as shown in Table 10.

Table 10 – AudioFingerprintID

Value	AudioFingerprintID
0h-1Fh	Audio Fingerprint ID: 0 to 31

Note: While audio fingerprints are enumerated, it is beyond the scope of this document to define the corresponding audio source channels from which particular audio fingerprints are derived. Consequently, users of this standard are advised to assure correspondence between fingerprint IDs and their associated audio source channels at the various locations of fingerprint origination and utilization.

AudioMixType – This 3-bit field shall indicate the type of audio downmix for the current audio fingerprint ID, as specified in Table 11.

Table 11 – **AudioMixType definition**

Value	**AudioMixType**
0h	Reserved for future use by SMPTE
1h	Mono (no downmix)
2h	Downmix from 2.0-channel audio
3h	Reserved for future use by SMPTE
4h	Reserved for future use by SMPTE
5h	Downmix from 5.1-channel audio
6h	Reserved for future use by SMPTE
7h	Reserved for future use by SMPTE

Audio Fingerprint Header Byte 3:

AFDataCount – This 5-bit field shall indicate the number of Audio Fingerprint Data bytes that follow as shown in Table 12. The AFDataCount byte count shall be derived from Table 13.

Table 12 – **AFDataCount**

Value	AFDataCount
0h	Reserved
1h~5h	Byte count: 1 to 5

Table 13 – Audio Fingerprint Cadence (AFDataCount values)

Frame Number within Cadence	p60	i60/ p30	p59.94	i59.94/ p29.97	p50	i50/ p25	p48	p24	p47.95	p23.98
1	2	4	1	3	2	4	2	5	2	4
2			2	4	2	5	3		2	5
3			2	4	3	5			3	5
4			2	4	2	5			2	5
5			2	4	3	5			3	5
6			2	4					2	4
7			2	3					2	5
8			2	4					3	5
9			2	4					2	5
10			2	4					3	5
11			2	4					2	4
12			2	4					2	5
13			2	4					3	5
14			1	3					2	5
15			2	4					3	5
16			2	4					2	5
17			2	4					2	
18			2	4					3	
19			2	4					2	
20			2	4					3	
21			2						2	
22			2						2	
23			2						3	
24			2						2	
25			2						3	
26			2						2	
27			1						2	
28			2						3	
29			2						2	
30			2						3	
31			2						2	
32			2						3	
33			2							
34			2							
35			2							
36			2							
37			2							
38			2							
39			2							
40			2							

Note: In Table 13 the cadence is cyclic and restarts at Frame Number 1.

The number of audio fingerprint data bytes per each video frame will vary according to the video frame rate. The varying cadence of audio fingerprint data bytes per each video frame is defined for each video frame rate. The start of the cadence is not aligned to any particular video frame. Table 13 shows the distribution of the number of audio fingerprints per each video frame that shall be used within each frame rate cadence. The cadences shown in Table 13 ensure uniform distribution of audio fingerprints over time for each video frame rate, and shall be the same for all audio fingerprints associated with that video frame cadence.

7 Sample Fingerprint Transport Packets (Informative)

Table 14 and Table 15 show examples of fingerprint transport packets generated under the conditions stated.

> Example 1: Table 14 illustrates a fingerprint transport packet containing one video fingerprint and three audio fingerprints. The video fingerprint is derived from a 1080i59.94 source. The audio fingerprint labeled Audio A is derived from a downmix of the 5.1-channel audio program in audio source channels 1 through 6 and has been assigned the fingerprint ID of 0h; the audio fingerprint labeled Audio B is derived from a downmix of the 2.0-channel audio program in audio source channels 7 and 8 and has been assigned the fingerprint ID of 1h; the audio fingerprint labeled Audio C is derived from the mono audio source channel 9 and has been assigned the fingerprint ID of 2h. The FP_protocol_version value is zero (0); the Sequence_Counter value for this packet is 43; the Length value is 24 bytes; and an ID Sub-Container is not present. The Checksum value is calculated as specified in Section 6.1.

Table 14 – Example 1 Fingerprint Transport Packet

	Byte Number	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Transport Header (Table 5)	1	00h							
	2	2Bh							
	3	18h							
	4	6h (0110b)				0b	0b	1b	1b
Video Fingerprint Sub-Container (Table 7)	5	0h (000b)			2h (10b)		1h (001b)		
Video Fingerprint Data	6	Video Fingerprint Data for field 1							
	7	Video Fingerprint Data for field 2							
Audio Fingerprint Sub-Container (Table 8)	8	2h (00010b)					2h (010b)		
Audio Fingerprint Header For Audio A	9	0h (00000b)					5h (101b)		
	10	3h (00011b)					0h (000b)		
Fingerprint Data For Audio A	11	Audio A Fingerprint Data(0)							
	12	Audio A Fingerprint Data(1)							
	13	Audio A Fingerprint Data(2)							
Audio Fingerprint Header For Audio B	14	1h (00001b)					2h (010b)		
	15	3h (00011b)					0h (000b)		
Fingerprint Data For Audio B	16	Audio B Fingerprint Data(0)							
	17	Audio B Fingerprint Data(1)							
	18	Audio B Fingerprint Data(2)							
Audio Fingerprint Header For Audio C	19	2h (00010b)					1h (001b)		
	20	3h (00011b)					0h (000b)		
Fingerprint Data For Audio C	21	Audio C Fingerprint Data(0)							
	22	Audio C Fingerprint Data(1)							
	23	Audio C Fingerprint Data(2)							
Checksum	24	xxh (To Be Calculated)							

Example 2: Table 15 illustrates a fingerprint transport packet containing one video fingerprint and two audio fingerprints: The video fingerprint is derived from a 720p50 source. The audio fingerprint labeled Audio A is derived from a downmix of the 5.1-channel audio program in audio source channels 1 through 6 and has been assigned the fingerprint ID of 0h; the audio fingerprint labeled Audio B is derived from a downmix of the 2.0-channel audio program in audio source channels 7 and 8 and has been assigned the fingerprint ID of 1h. The FP_protocol_version value is zero (0); the Sequence_Counter value for this packet is 212; the Length value is 16 bytes; and an ID Sub-Container is not present. The Checksum value is calculated as specified in Section 6.1.

Table 15 – Example 2 Fingerprint Transport Packet

	Number of Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Transport Header (Table 5)	1	00h							
	2	D4h							
	3	10h							
	4	9h (1001b)				0b	0b	1b	1b
Video Fingerprint Sub-Container (Table 7)	5	0h (000b)			1h (01b)		1h (001b)		
Video Fingerprint Data	6	Video Fingerprint Data							
Audio Fingerprint Sub-Container (Table 8)	7	1h (00001b)					2h (010b)		
Audio Fingerprint Header For Audio A	8	0h (00000b)					5h (101b)		
	9	2h (00010b)					0h (000b)		
Fingerprint Data For Audio A	10	Audio A Fingerprint Data(0)							
	11	Audio A Fingerprint Data(1)							
Audio Fingerprint Header For Audio B	12	1h (00001b)					2h (010b)		
	13	2h (00010b)					0h (000b)		
Fingerprint Data For Audio B	14	Audio B Fingerprint Data(0)							
	15	Audio B Fingerprint Data(1)							
Checksum	16	xxh (To Be Calculated)							

Annex A Bibliography (Informative)

Recommendation ITU-R BT.1359-1 (1998), “Relative Timing of Sound and Vision for Broadcasting”
International Telecommunications Union, Geneva,

ATSC IS-191 (2003), “Relative Timing of Sound and Vision for Broadcast Operations”
Advanced Television Systems Committee, Washington, D.C.,

EBU Recommendation R37-2007, “The Relative Timing of the Sound and Vision Components of a Television Signal” European Broadcasting Union, Geneva,

EBU Technical Guidelines Report 3311 (2006), “EBU Guidelines for Multichannel Audio in DVB”
European Broadcasting Union, Geneva,

“Detection and Correction of Lip-Sync Errors Using Audio and Video Fingerprints”
Kent Terry and Regunathan Radhakrishnan, SMPTE Motion Imaging Journal, April 2010

“Fingerprinting for Solving A/V Synchronization Issues within Broadcast Environments”
Sara Kudrle, Michel Proulx, Pascal Carrières, and Marco Lopez, SMPTE Motion Imaging Journal, July-August 2011

“Interoperable AV Sync Systems in the SMPTE 22TV Lip Sync AHG: Content-Fingerprinting-Based Audio-Video Synchronization”
Mihailo Stojancic and Daniel Eakins, SMPTE Motion Imaging Journal, July-August 2011

“Monitoring and Control of Audio-to-Video Delay in Broadcast Systems”
Tom Tucker and Dan Baker, SMPTE Motion Imaging Journal, October 2002

“Video-to-Audio Synchrony Monitoring and Correction”
J. Carl Cooper, SMPTE Journal, September 1988

“Factors affecting perception of audio-video synchronisation in television”
Andrew Mason and Richard Salmon, Audio Engineering Society Convention Paper, October 2008