Mp4Rec Library Overview

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

1 Library Overview

Scope of This Document

This document is for the Mp4Rec library, which records input video and audio as an MP4 format video file. Basic procedure and restrictions are explained.

Purpose and Features

The Mp4Rec library compresses input video to AVC codec and input audio to AAC codec, and outputs them as an MP4 file. Video and audio stream data can be added without taking special note of the MP4 file format or codec to easily create MP4 files.

Main Features

From an input video and audio, an MP4 file of one video and one audio track will be created.

Output File Specifications

The specifications of MP4 files output by the Mp4Rec library are as follows.

Format	Description	
Video	Codec	MPEG4, AVC baseline profile
	Frame size	368 x 208, 480 x 272, 640 x 368 pixels
	Frame rate	29.97 Hz progressive
	Bit rate	2 Mbps, 1 Mbps
Audio	Codec	AAC-LC 16 bit stereo
	Bit rate	128 Kbps
	Sample rate	48 KHz

Embedding into a Program

The files required to use the Mp4Rec library are as follows.

Filename	Description
mp4rec.h	Header file
libSceMp4Rec_stub.a	Stub library file
libSceMp4Rec_stub_weak.a	Weak import stub library file

Include mp4rec.h in the source program. Various header files will be automatically included as well. Upon building the program, link libSceMp4Rec_stub.a.

Before calling the library functions, load PRX with the following code.

```
sceSysmoduleLoadModule( SCE_SYSMODULE_MP4_RECORDER );
```

After using the library, unload PRX with the following code.

```
sceSysmoduleUnloadModule( SCE_SYSMODULE_MP4_RECORDER );
```

Sample Programs

Refer to the following sample program for the Mp4Rec library.

sample_code/audio_video/api_mp4recorder//simple/

This sample shows basic usage of the Mp4Rec library.

sample_code/audio_video/api_mp4recorder/recorder/

In addition to exemplifying the basic usage of the Mp4Rec library, this sample changes the MP4 recording mode (bit rate, for example), and attaches metadata (titles, thumbnails, etc.) to the MP4 file.

sample_code/audio_video/api_mp4recorder/transcode/

This sample transcodes MP4 video. After decoding input MP4 video, the Mp4Rec library is used to encode the data again. Image quality in each mode of the Mp4Rec library can be confirmed using the transcoded MP4 file.

2 Using the Library

Basic Procedure

Basic procedure for MP4 recording is explained below. An overview of the processing flow is as follows.

(1) Create an instance of the Mp4Rec library
(2) Query size of physical continuous memory required for recording
(3) Allocate buffers for recording
(4) Initialize the MP4 recorder feature
(5) Add a video sample
(6) Add an audio sample
(7) Repeat video sample and audio sample additions
(8) Terminate the MP4 recorder feature
(9) Delete the instance of the Mp4Rec library

(1) Create an instance of the Mp4Rec library

Call `sceMp4RecCreateRecorder()` to create an Mp4Rec library instance. To create an instance, the heap area to be used by the Mp4Rec library must be specified. A heap memory of at least `SCE_MP4REC_MIN_HEAP_SIZE` (4 MiB) is required.

(2) Query size of physical continuous memory required for recording

Call `sceMp4RecQueryPhysicalMemSize()` to obtain the physical continuous memory sizes to be used by the AVC encoder and AV recorder for recording. The required sizes will vary according to the frame size to record.

(3) Allocate buffers for recording

Allocate a buffer of the sizes obtained in step (2). Allocate an AVC encoder work memory and AV recorder memory. For details, refer to the Description of the `SceMp4RecInitParam` structure in the "Mp4Rec Library Reference" document.

(4) Initialize the MP4 recorder feature

Call `sceMp4RecInit()` to initialize MP4 recorder feature. Set the `SceMp4RecInitParam` structure with the MP4 recording mode constant, as well as pointers to - and sizes of - the AVC encoder work memory and AV recorder memory allocated in step (3). A thread for writing to a file is generated in the library; the CPU affinity mask and priority of this thread can be specified upon initialization.

(5) Add a video sample

Call `sceMp4RecAddVideoSample()` to add a video sample (one picture). The video sample must be in the `SCE_MP4REC_PIXEL_YUV420_PACKED_RASTER` format. For details on conversion, refer to the "sceMp4RecAddVideoSample " section of the "Mp4Rec Library Reference" document.

(6) Add an audio sample

Call `sceMp4RecAddAudioSample()` to add an audio sample (1024 samples).

(7) Repeat video sample and audio sample additions

Repeat steps (5) and (6) until recording ends. The maximum recording time is 30 minutes.

Take note that one second is not exceeded as the time axis differential when adding video and audio samples. When one second is exceeded, a buffer overflow will occur. For details, refer to the "Notes on Adding Video/Audio Samples" section.

(8) Terminate the MP4 recorder feature

Call `sceMp4RecTerm()` to terminate recording.

When `false` is specified to *`isCancel`* of the `SceMp4RecTermParam` structure, data recorded up to that point will be output as an MP4 file. This output Mp4 file can be played from the Videos application.

When `true` is specified to *`isCancel`* of the `SceMp4RecTermParam` structure, the recording can be stopped and the MP4 file being recorded can be deleted.

By setting the `SceMp4RecMetadata` structure, metadata (title, thumbnail, etc.) can be attached to the MP4 file.

(9) Delete the instance of the Mp4Rec library

Call `sceMp4RecDeleteRecorder()` to delete the Mp4Rec library instance. This will release resources allocated when the Mp4Rec library instance was created.

Main APIs Used in Basic Processing

API	Description
`sceMp4RecCreateRecorder()`	Creates an instance of the Mp4Rec library
`sceMp4RecDeleteRecorder()`	Deletes the instance of the Mp4Rec library
`sceMp4RecQueryPhysicalMemSize()`	Queries the sizes of physical continuous memory to be used by the AVC encoder and AV recorder
`sceMp4RecInit()`	Initializes the MP4 recorder feature
`sceMp4RecAddVideoSample()`	Adds a video sample
`sceMp4RecAddAudioSample()`	Adds an audio sample
`sceMp4RecTerm()`	Terminates the MP4 recorder feature

Frame Data of Pictures

The frame of pictures handled in the Mp4Rec library must be images in the RGBA (32 bit), BGRA (32 bit), or YCbCr format that have been raster scanned from the upper left to lower right direction.

There are two types to the YCbCr format. When `SCE_MP4REC_PIXEL_YUV420_RASTER` is specified, the raster image has each Y, Cb, and Cr component taken independently from separate pixels. When `SCE_MP4REC_PIXEL_YUV420_PACKED_RASTER` is specified, the raster image is one in which the Y component is separated from the Cb and Cr (taken from the same pixel) chroma components. The pixel type must be `SCE_MP4REC_PIXEL_YUV420_PACKED_RASTER` for `sceMp4RecAddVideoSample()`.

When a change in frame size or pixel type is required, call `sceMp4RecCsc()` and have conversion carried out in advance. *`frameWidth`* and *`frameHeight`* of the `SceMp4RecFrame` structure within the `SceMp4RecPicture` structure must be the same value as the frame size set to the MP4 recording mode of the `SceMp4RecInitParam` structure with `sceMp4RecInit()`. Make the value of *`framePitch`* the same as *`frameWidth`*.

Set the picture to the pointer address stored in *`pPicture`*`[0]` of the `SceMp4RecFrame` structure in the `SceMp4RecPicture` structure. Allocate a buffer of size *`framePitch`* x *`frameHeight`* x (coefficient according to *`pixelType`*) on an un-cached continuous physical address space (custom DRAM or physical continuous memory on main memory) that is aligned to a 256-byte boundary.

The coefficients according to *pixelType* are as follows.

pixelType	Coefficient
SCE_MP4REC_PIXEL_RGBA8888	4
SCE_MP4REC_PIXEL_BGRA8888	
SCE_MP4REC_PIXEL_YUV420_RASTER	1.5
SCE_MP4REC_PIXEL_YUV420_PACKED_RASTER	

For example, allocate a buffer of 557056 bytes (= 512 x 272 x 4) for the following picture.

- *framePitch* = 512
- *frameHeight* = 272
- *pixelType* = SCE_MP4REC_PIXEL_RGBA8888

frameWidth must be equal to or less than *framePitch*.

framePitch must be a multiple of 32 only for SCE_MP4REC_PIXEL_YUV420_RASTER.

Moreover, *framePitch* x *frameHeight* must be 640 x 480 or less.

Color Gamut Coefficients (Color Conversion Coefficients)

Color gamut coefficients are used to specify a color matrix when displaying a video clip. ITU-R BT.601 and ITU-R BT.709 are supported in this library. Color gamut coefficients are used with sceMp4RecInit() and sceMp4RecCsc().

When a color gamut coefficient is specified as an MP4 recording mode option for *recMode* in the SceMp4RecInitParam structure (an sceMp4RecInit() parameter), ITU-R BT.601 or ITU-R BT.709 will be stored in the VUI syntax in the AVC stream in the MP4 file, and it will be used as reference information during playback. If the wrong color gamut coefficient is specified, playback with the proper colors may not be possible. If the color gamut coefficient is omitted in the MP4 recording mode options, ITU-R BT.601 will be set.

Color gamut coefficients are also used when converting from RGB to YUV or from YUV to RGB with sceMp4RecCsc(). In both cases, specify the RGB color conversion coefficient for *cscCoefficient* in the SceMp4RecFrameOptionRGBA structure. Match the color conversion coefficient specified here with the color gamut coefficient for the aforementioned sceMp4RecInit().

Color conversion does not occur for YUV to YUV size conversion, therefore *cscCoefficient* will be invalid if specified.

When creating an MP4 file with this library using YUV format video as input, the color matrix of the original video must be determined and the proper color gamut coefficient must be specified for *recMode* as an MP4 recording mode option. If the proper color gamut coefficient is not specified, there is a possibility that the MP4 file will have different colors from the original video.

Notes on Adding Video/Audio Samples

Interleave Interval

Sample data is added with sceMp4RecAddVideoSample() and sceMp4RecAddAudioSample(). The call timing of these two functions must be noted. In this library, video and audio sample data of approximately one second are alternately interleaved and written to an MP4 file. Thus, when the addition of one sample data is continued for over one second, the other sample data cannot be written and the limited buffer will overflow. Carry out addition of the video sample and audio sample following the time axis.

When the video buffer overflows, insert the subsequently-described skip picture.

Skip Picture

For a video with an average bit rate that doesn't fit into the required value, the size of one sample becomes large and a buffer overflow may occur. When a video buffer overflows, this library generates a skip picture and continues with the recording. A skip picture refers to the process of using the previous picture as a current picture to reduce the sample size. When a skip picture is generated, the same picture is used twice and the picture will be played back as though the picture has temporarily stopped.

When a skip picture is generated, the return value of `sceMp4RecAddVideoSample()` is `SCE_MP4REC_ERROR_AVREC_VIDEO_BUFFER_FULL`. When the required number of video samples and audio samples is collected, the buffer contents are written. The buffer will subsequently be restored from the overflow state.

Insufficient Storage Space

When calling `sceMp4RecAddVideoSample()` and `sceMp4RecAddAudioSample()` and the `SCE_MP4REC_ERROR_NOSPACE` error returns, there is not enough recording space. Stop adding samples and call `sceMp4RecTerm()` to create an MP4 file of data recorded before the error occurrence.

Recordable Time

When calling `sceMp4RecAddVideoSample()` and `sceMp4RecAddAudioSample()` and the `SCE_MP4REC_ERROR_DURATION_LIMIT` error returns, the recording time has exceeded the maximum 30 minutes. Stop adding samples and call `sceMp4RecTerm()` to create an MP4 file before the error occurrence.

3 Notes

Restrictions on the Recording Time of the MP4 Recorder Feature

The minimum recording time of the MP4 recorder feature is 5 seconds. When `sceMp4RecTerm()` is called before the duration of 5 seconds, dummy data will be inserted to create a 5-second MP4 file.

The maximum recording time is 30 minutes.

Restriction on the Buffer for Placing Video Samples

The input frame buffer must be on an un-cached continuous physical address space (custom DRAM or physical continuous memory on main memory) that is aligned to a 256-byte boundary. An error will occur if this condition is not satisfied.

Restriction on the Buffers Specified for Initializing the MP4 Recorder Feature

The buffers for the AVC encoder work memory and the AV recorder memory specified to `sceMp4RecInit()`, which initializes the MP4 recorder feature, must be on an un-cached continuous physical address space (custom DRAM or physical continuous memory on main memory) that is aligned to a 256-byte boundary.

Processing Load

Because AAC encoding is carried out on the ARM processor, computation resources are required.

Moreover, AVC encoding carries out computation on the media processor Codec Engine. On an application that uses the Codec Engine such as the NGS module, note that execution is carried out in the FIFO order and processing will take longer because of this.

Real-time MP4 Recording

This library does not guarantee real-time recording.

There are cases where processing load becomes heavy due to encode processing or write to storage and real-time processing becomes difficult.

Required Storage Space for MP4 Recording

At least 10 MiB is required to start MP4 recording. When storage space becomes less than 10 MiB, `sceMp4RecAddVideoSample()` and `sceMp4RecAddAudioSample()` will return the `SCE_MP4REC_ERROR_NOSPACE` error.

Do Not Remove the Memory Card or Switch Off Power During MP4 Recording

If the memory card is removed or the power is switched off during MP4 recording, the memory card may become corrupted. Make sure not to remove the memory card or switch off power during MP4 recording.