

JPEG Encoder Overview

© 2011 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

1 Library Overview.....	3
Scope of This Document.....	3
Purpose and Features.....	3
Main Functions	3
Embedding in Program	3
Sample Programs.....	3
References	4
2 Usage	5
Basic Procedure	5
3 Precautions	7
Buffer Restrictions for Positioning I/O Data.....	7
Image Size Supported for Color Space Conversion	7

1 Library Overview

Scope of This Document

This document describes the JPEG Encoder library (scejpegenc), which provides functions for JPEG data encoding. It describes the basic procedure for encoding image data in YCbCr422/420 and the buffer restrictions for positioning I/O data that results from processing on a Codec Engine.

Purpose and Features

The JPEG Encoder library provides functions for encoding image data in YCbCr422/420 to JPEG format. To encode RGBA images, you can use the color space conversion function provided by this library to convert from RGBA to YCbCr422/420, and then you can encode the image data.

Main Functions

The JPEG Encoder library provides the following main functions.

- Encoding of YCbCr422/420 input images to JPEG data
(The color space sampling of the input and output formats are the same.)
- The following encoding settings can be changed:
 - Frame buffer size
 - Maximum image size
 - Encoding quality
 - Header format (Motion JPEG or normal JPEG)
- Color space conversion function for converting from RGBA to YCbCr422/420
(Special hardware can be used for high-speed processing.)

Embedding in Program

The files required for using JPEG Encoder library are as follows.

Filename	Description
scejpegenc.h	Header file
libSceJpegEnc_stub.a	Stub library file

Include scejpegenc.h in source program (some other header files will also be included automatically). When building the program, link libSceJpegEnc_stub.a.

Sample Programs

Refer to the following sample program for the JPEG Encoder library.

samples/sample_code/audio_video/api_scejpegenc/simple

This sample shows the basic usage of the JPEG Encoder library.

References

For the JPEG format, refer to the following standard specifications as necessary.

- ITU-T Recommendation T.81 "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines"
<http://www.itu.int/rec/T-REC-T.81-199209-I/en>
ITU, updated 1992/09
- JPEG File Interchange Format Version 1.02
<http://www.jpeg.org/public/jfif.pdf>
Eric Hamilton, C-Cube Microsystems, updated 1992/09/01

(The above reference destination has been confirmed as of March 8, 2011. Note that pages may have been subsequently moved or its contents modified.)

000004892117

2 Usage

Basic Procedure

Basic procedures to process encoding of image data to JPEG files are explained here. The following is an overview of the process flow.

- (1) Provide the images to be encoded and the buffer for storing the encoding results.
- (2) Allocate work memory for encoding
- (3) Initialize the encoder
- (4) Configure the encoding settings
- (5) If the input image is RGBA, convert its color space to YCbCr422/420
- (6) Encode the data
- (7) End processing

(1) Provide the images to be encoded and the buffer for storing the encoding results.

If the input image is RGBA, prepare the required buffer to perform conversion to YCbCr422/420 using the color space conversion function `sceJpegEncoderCsc()`. (Note that a pitch difference is added to the width for YCbCr data output by `sceJpegEncoderCsc()`. Pitch difference refers to the added padding area when the number of bytes of one line per component (Y, Cb, and Cr) is not a multiple of 16, and is required when using the dedicated hardware for color space conversion.)

It is recommended that a sufficient output buffer size be allocated (roughly the number of bytes for the number of pixels of the image to be encoded).

The buffer where this I/O data is positioned has restrictions, and the buffer must be allocated on the video memory or physical continuous area of the main memory. Refer to "Buffer Restrictions for Positioning I/O Data" in "3 Precautions" for details.

(2) Allocate work memory for encoding

When `sceJpegEncoderGetContextSize()` is called, the required work memory size is returned, so allocate the memory of that size (bytes). Use a multiple of four for the starting address of the memory.

(3) Initialize the encoder

Call the encoder initialization function `sceJpegEncoderInitWithParam()` to perform initialization. Specify the pointer to the memory allocated in step (2), the input image format, vertical and horizontal pixel size of the frame buffer, pointer to the output buffer, output buffer size, and initialization option to the structure specified in the argument. When the input image is YCbCr420, specify both the width and height of the pixel size of the frame buffer in multiples of 16. When the input image is YCbCr422, specify the width in multiples of 16 and the height in multiples of 8. Appropriate value must be specified to the initialization option by determining whether video memory or main memory is utilized as a buffer region to pass data to the encoder. If you plan to use video memory only, specify `SCE_JPEGENC_INIT_OPTION_NONE`. If only the main memory is to be used, or both video and main memory are used, specify `SCE_JPEGENC_INIT_OPTION_LPDDR2_MEMORY` instead.

(4) Configure the encoding settings

By default, all data within the frame buffer is compressed at standard quality. To compress only part of the frame or to change the quality (reduce the output data size or increase the quality), this must be set separately.

Set the valid region with `sceJpegEncoderSetValidRegion()` and set the quality with `sceJpegEncoderSetCompressionRatio()`.

SCE CONFIDENTIAL

In addition, to compile the data into an AVI file and record it as a movie, the Motion JPEG header must be output. Do this with `sceJpegEncoderSetHeaderMode()`.

(5) If the input image is RGBA, convert its color space to YCbCr422/420

If the input image is RGBA, call the color space conversion function `sceJpegEncoderCsc()` and convert the input image to YCbCr422/420. (This is not required if the input image is YCbCr422/420.)

(6) Encode the data

Call `sceJpegEncoderEncode()` to perform encoding.

The size of the output JPEG data is returned as the return value, so record this to the appropriate file.

(7) End processing

Call `sceJpegEncoderEnd()` to end processing.

000004892117

3 Precautions

Buffer Restrictions for Positioning I/O Data

Because this library is processed on the Codec Engine, the following restrictions apply to buffers that position I/O data.

(1) Input data buffer passed to encoding functions

This must be a physical address in a continuous area and fulfill 16-byte alignment. However, conditions are even stricter when simultaneously calling encoding functions from several threads. They must have a physical address in a continuous area and fulfill 256-byte alignment.

(2) Output data buffer passed to encoding functions, and I/O data buffers passed to color space conversion functions

This must have a physical address in a continuous area and fulfill 256-byte alignment.

To allocate a physical address in a continuous area, the buffer must be positioned within the video memory or physical continuous area of the main memory. For details on memory allocation, refer to the "Kernel Reference" document.

Image Size Supported for Color Space Conversion

Supported image size when performing color space conversion with `sceJpegEncoderCsc()` is limited to the range between 64 x 64 pixels and 2032 x 1088 pixels. For images outside this range, use the user application to perform color space conversion and then perform encoding.