原 FFMPEG结构体分析：AVStream

2013年11月10日 00:02:04　阅读数：34948

注：写了一系列的结构体的分析的文章，在这里列一个列表：

FFMPEG结构体分析：AVFrame

FFMPEG结构体分析：AVFormatContext

FFMPEG结构体分析：AVCodecContext

FFMPEG结构体分析：AVIOContext

FFMPEG结构体分析：AVCodec

FFMPEG结构体分析：AVStream

FFMPEG结构体分析：AVPacket

FFMPEG有几个最重要的结构体，包含了解协议，解封装，解码操作，此前已经进行过分析：

FFMPEG中最关键的结构体之间的关系

在此不再详述，其中AVStream是存储每一个视频/音频流信息的结构体。本文将会分析一下该结构体里重要变量的含义和作用。

首先看一下结构体的定义（位于avformat.h文件中）：

```cpp
/* 雷霄骅
 * 中国传媒大学/数字电视技术
 * leixiaohua1020@126.com
 *
 */
/**
 * Stream structure.
 * New fields can be added to the end with minor version bumps.
 * Removal, reordering and changes to existing fields require a major
 * version bump.
 * sizeof(AVStream) must not be used outside libav*.
 */
typedef struct AVStream {
    int index;    /**< stream index in AVFormatContext */
    /**
     * Format-specific stream ID.
     * decoding: set by libavformat
     * encoding: set by the user
     */
    int id;
    AVCodecContext *codec; /**< codec context */
    /**
     * Real base framerate of the stream.
     * This is the lowest framerate with which all timestamps can be
     * represented accurately (it is the least common multiple of all
     * framerates in the stream). Note, this value is just a guess!
     * For example, if the time base is 1/90000 and all frames have either
     * approximately 3600 or 1800 timer ticks, then r_frame_rate will be 50/1.
     */
    AVRational r_frame_rate;
    void *priv_data;

    /**
     * encoding: pts generation when outputting stream
     */
    struct AVFrac pts;

    /**
     * This is the fundamental unit of time (in seconds) in terms
     * of which frame timestamps are represented. For fixed-fps content,
     * time base should be 1/framerate and timestamp increments should be 1.
     * decoding: set by libavformat
     * encoding: set by libavformat in av_write_header
     */
    AVRational time_base;

    /**
     * Decoding: pts of the first frame of the stream in presentation order, in stream time base.
     * Only set this if you are absolutely 100% sure that the value you set
     * it to really is the pts of the first frame.
     * This may be undefined (AV_NOPTS_VALUE).
     * @note The ASF header does NOT contain a correct start_time the ASF
     * demuxer must NOT set this.
     */
    int64_t start_time;

```

```
    /**
     * Decoding: duration of the stream, in stream time base.
     * If a source file does not specify a duration, but does specify
     * a bitrate, this value will be estimated from bitrate and file size.
     */
    int64_t duration;

    int64_t nb_frames;                 ///< number of frames in this stream if known or 0

    int disposition; /**< AV_DISPOSITION_* bit field */

    enum AVDiscard discard; ///< Selects which packets can be discarded at will and do not need to be demuxed.

    /**
     * sample aspect ratio (0 if unknown)
     * - encoding: Set by user.
     * - decoding: Set by libavformat.
     */
    AVRational sample_aspect_ratio;

    AVDictionary *metadata;

    /**
     * Average framerate
     */
    AVRational avg_frame_rate;

    /**
     * For streams with AV_DISPOSITION_ATTACHED_PIC disposition, this packet
     * will contain the attached picture.
     *
     * decoding: set by libavformat, must not be modified by the caller.
     * encoding: unused
     */
    AVPacket attached_pic;

    /*****************************************************************
     * All fields below this line are not part of the public API. They
     * may not be used outside of libavformat and can be changed and
     * removed at will.
     * New public fields should be added right above.
     *****************************************************************
     */

    /**
     * Stream information used internally by av_find_stream_info()
     */
#define MAX_STD_TIMEBASES (60*12+5)
    struct {
        int64_t last_dts;
        int64_t duration_gcd;
        int duration_count;
        double duration_error[2][2][MAX_STD_TIMEBASES];
        int64_t codec_info_duration;
        int nb_decoded_frames;
        int found_decoder;
    } *info;

    int pts_wrap_bits; /**< number of bits in pts (used for wrapping control) */

    // Timestamp generation support:
    /**
     * Timestamp corresponding to the last dts sync point.
     *
     * Initialized when AVCodecParserContext.dts_sync_point >= 0 and
     * a DTS is received from the underlying container. Otherwise set to
     * AV_NOPTS_VALUE by default.
     */
    int64_t reference_dts;
    int64_t first_dts;
    int64_t cur_dts;
    int64_t last_IP_pts;
    int last_IP_duration;

    /**
     * Number of packets to buffer for codec probing
     */
#define MAX_PROBE_PACKETS 2500
    int probe_packets;

    /**
     * Number of frames that have been demuxed during av_find_stream_info()
     */
    int codec_info_nb_frames;

    /**
     * Stream Identifier
     * This is the MPEG-TS stream identifier +1
     * 0 means unknown
     */
    int stream_identifier;
```

```
148.
149.     int64_t interleaver_chunk_size;
150.     int64_t interleaver_chunk_duration;
151.
152.     /* av_read_frame() support */
153.     enum AVStreamParseType need_parsing;
154.     struct AVCodecParserContext *parser;
155.
156.     /**
157.      * last packet in packet_buffer for this stream when muxing.
158.      */
159.     struct AVPacketList *last_in_packet_buffer;
160.     AVProbeData probe_data;
161. #define MAX_REORDER_DELAY 16
162.     int64_t pts_buffer[MAX_REORDER_DELAY+1];
163.
164.     AVIndexEntry *index_entries; /**< Only used if the format does not
165.                                     support seeking natively. */
166.     int nb_index_entries;
167.     unsigned int index_entries_allocated_size;
168.
169.     /**
170.      * flag to indicate that probing is requested
171.      * NOT PART OF PUBLIC API
172.      */
173.     int request_probe;
174. } AVStream;
```

AVStream重要的变量如下所示：

int index：标识该视频/音频流

AVCodecContext *codec：指向该视频/音频流的AVCodecContext（它们是一一对应的关系）

AVRational　time_base：时基。通过该值可以把PTS，DTS转化为真正的时间。FFMPEG其他结构体中也有这个字段，但是根据我的经验，只有AVStream中的 time_base是可用的 。PTS*time_base=真正的时间

int64_t duration：该视频/音频流长度

AVDictionary *metadata：元数据信息

AVRational avg_frame_rate：帧率（注：对视频来说，这个挺重要的）

AVPacket attached_pic：附带的图片。比如说一些MP3，AAC音频文件附带的专辑封面。

该结构体其他字段的作用目前还有待于探索。

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/leixiaohua1020/article/details/14215821

文章标签： ffmpeg avstream 视频 解码 结构体

个人分类： FFMPEG

所属专栏： FFmpeg

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com