

FFmpeg的HEVC解码器源代码简单分析：概述

2015年06月08日 14:24:57 阅读数：9921

HEVC源代码分析文章列表：

【解码 -libavcodec HEVC 解码器】

FFmpeg的HEVC解码器源代码简单分析：概述

FFmpeg的HEVC解码器源代码简单分析：解析器（Parser）部分

FFmpeg的HEVC解码器源代码简单分析：解码器主干部分

FFmpeg的HEVC解码器源代码简单分析：CTU解码（CTU Decode）部分-PU

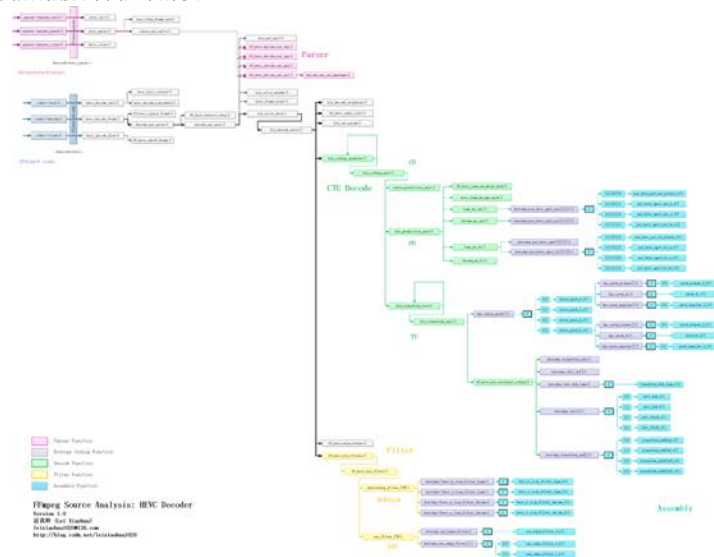
FFmpeg的HEVC解码器源代码简单分析：CTU解码（CTU Decode）部分-TU

FFmpeg的HEVC解码器源代码简单分析：环路滤波（LoopFilter）

从这篇文章开始，简单分析记录FFmpeg中libavcodec的HEVC (H.265) 解码器 (HEVC Decoder) 的源代码。本文综述整个解码器的框架，后续几篇文章再对解码器的内部模块进行分析。

函数调用关系图

FFmpeg的HEVC (H.265) 解码器的函数调用关系图如下所示。



单击查看更清晰的大图

下面解释一下图中关键标记的含义。

作为接口的结构体

FFmpeg和HEVC解码器之间作为接口的结构体有2个：

ff_hevc_parser：用于解析HEVC码流的AVCodecParser结构体。

ff_hevc_decoder：用于解码HEVC码流的AVCodec结构体。

函数背景色

函数在图中以方框的形式表现出来。不同的背景色标志了该函数不同的作用：

白色背景的函数：普通内部函数。

粉红色背景函数：解析函数（Parser）。这些函数用于解析SPS、PPS等信息。

绿色背景的函数：解码函数（Decode）。这些函数通过帧内预测、帧间预测、DCT反变换等方法解码压缩数据。

黄色背景的函数：滤波函数（Filter）。这些函数对解码后的数据进行滤波，去除方块效应。

蓝色背景函数：汇编函数（Assembly）。这些函数是做过汇编优化的函数。图中主要画出了这些函数的C语言版本，此外这些函数还包含MMX版本、SSE版本、NEON版本等。

箭头线

箭头线标志了函数的调用关系：

- 黑色箭头线：不加区别的调用关系。
- 粉红色的箭头线：解析函数（Parser）之间的调用关系。
- 绿色箭头线：解码函数（Decode）之间的调用关系。
- 黄色箭头线：环路滤波函数（Loop Filter）之间的调用关系。

函数所在的文件

每个函数标识了它所在的文件路径。

几个关键部分

下文简单记录几个关键的部分。

FFmpeg和HEVC解码器之间作为接口的结构体

FFmpeg和HEVC解码器之间作为接口的结构体有2个：ff_hevc_parser和ff_hevc_decoder。

ff_hevc_parser

ff_hevc_parser是用于解析HEVC码流的AVCodecParser结构体。AVCodecParser中包含了几个重要的函数指针：

- parser_init()：初始化解析器。
- parser_parse()：解析。
- parser_close()：关闭解析器。

在ff_hevc_parser结构体中，上述几个函数指针分别指向下面几个实现函数：

- hevc_init()：初始化HEVC解析器。
- hevc_parse()：解析HEVC码流。
- hevc_close()：关闭HEVC解析器。

ff_hevc_decoder

ff_hevc_decoder是用于解码HEVC（H.265）码流的AVCodec结构体。AVCodec中包含了几个重要的函数指针：

- init()：初始化解码器。
- decode()：解码。
- close()：关闭解码器。

在ff_hevc_decoder结构体中，上述几个函数指针分别指向下面几个实现函数：

- hevc_decode_init()：初始化HEVC解码器。
- hevc_decode_frame()：解码HEVC码流。
- hevc_decode_free()：关闭HEVC解码器。

解析函数（Parser）

解析函数（Parser）用于解析HEVC码流中的一些信息（例如SPS、PPS、Slice Header等）。在parse_nal_units()和decode_nal_units()中都调用这些解析函数完成了解析。下面举几个解析函数的例子：

- ff_hevc_decode_nal_vps()：解析VPS。
- ff_hevc_decode_nal_sps()：解析SPS。
- ff_hevc_decode_nal_pps()：解析PPS。
- ff_hevc_decode_nal_sei()：解析SEI。

普通内部函数

普通内部函数指的是HEVC解码器中还没有进行分类的函数。下面举几个例子。

ff_hevc_decoder中hevc_decode_init()调用的初始化函数：

- hevc_init_context()：初始化HEVC解码器上下文结构体。
- hevc_decode_extradata()：解析AVCodecContext中的extradata。

ff_hevc_decoder中hevc_decode_frame()逐层调用的和解码Slice相关的函数：

decode_nal_units(), decode_nal_unit(), hls_slice_data(), hls_decode_entry()等。
ff_hevc_decoder中hevc_parse()逐层调用的和解析Slice相关的函数：
hevc_find_frame_end(): 查找NALU的结尾。
parse_nal_units(): 解析一个NALU。

hls_decode_entry()

hls_decode_entry()是FFmpeg的HEVC解码器真正的解码函数。其中调了解码函数和滤波函数HEVC中的CTU进行处理。在HEVC中CTU（Coding tree unit，编码树单元）即对应H.264中的MB（Macroblock，宏块）。

解码函数（Decode）

解码函数（Decode）通过帧内预测、帧间预测等方法解码CTU压缩数据。CTU解码模块对应的函数是hls_coding_quadtree()。hls_coding_quadtree()用于解析HEVC码流的四叉树结构的句法，是一个递归调用的函数。当解析到单个CU的时候，会调用CU的解码函数hls_coding_unit()。

hls_coding_unit()会调用hls_prediction_unit()和hls_transform_tree()分别对CU中的PU和TU进行处理。hls_prediction_unit()会调用luma_mc_uni()或者调用luma_mc_bi()进行预测。hls_transform_tree()用于解析TU的四叉树结构的句法，是一个递归调用的函数。当解析到单个TU的时候，会调用hls_transform_unit()对TU进行处理。

环路滤波函数（Loop Filter）

环路滤波函数（Loop Filter）对解码后的数据进行滤波，去除方块效应和振铃效应。滤波模块对应的环路滤波函数是ff_hevc_hls_filters()。ff_hevc_hls_filters()调用了ff_hevc_hls_filter()。而ff_hevc_hls_filter()调用去块效应滤波器函数deblocking_filter_CTB()去除解码过程中的块效应；调用SAO滤波器函数sao_filter_CTB()去除解码过程中的振铃效应。

汇编函数（Assembly）

汇编函数（Assembly）是做过汇编优化的函数。为了提高效率，整个HEVC解码器中包含了大量的汇编函数。实际解码的过程中，FFmpeg会根据系统的特性调用相应的汇编函数（而不是C语言函数）以提高解码的效率。如果系统不支持汇编优化的话，FFmpeg才会调用C语言版本的函数。

至此FFmpeg的HEVC解码器的结构就大致梳理完毕了。

雷霄骅

leixiaohua1020@126.com

<http://blog.csdn.net/leixiaohua1020>

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/46412023>

文章标签： [FFmpeg](#) [HEVC](#) [视频编码](#) [CTU](#) [SAO](#)

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