

用X265对HEVC/H.265视频进行隐写steganography---一、基本工具

原创

dyaa44 于 2021-01-17 23:20:40 发布 337 收藏 1

分类专栏: [HEVC视频隐写 steganography](#) 文章标签: [机器学习](#) [hevc](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/dyaa44/article/details/112758861>

版权



[HEVC视频隐写 steganography](#) 专栏收录该内容

3 篇文章 0 订阅

订阅专栏

用X265对HEVC/H.265视频进行隐写steganography---一、基本工具

概述

1.X265的安装及测试

2.elecard HEVC analyzer的基本使用

2.1 CUIPUTU划分

2.2 帧内预测模式

2.3 运动向量

2.4 DCT/DST系数

概述

写个用X265对HEVC/H.265视频隐写的指南, 希望对大家有帮助。

基本工具如下(全部windows平台):

X265(编码器)

elecard HEVC analyzer (265文件的分析软件, 可以用来查看是否参数按你想要的方式修改成功了)

ffmpeg/HM (解码器, 任选)

测试序列

所有相关程序也可在以下网盘中下载:

链接: <https://pan.baidu.com/s/1qQUIGu3CRaKa8W4cmTvIGA>

提取码: jy83

HEVC标准测试序列也贴下网盘:

链接: <https://pan.baidu.com/s/1kyjCXumOYWmycdg-IXN4Qg>

提取码: af9b

1.X265的安装及测试

在 <http://x265.org> 官网Get Involved栏点击contribute to x265，进入后点download
(现在好像需要注册)

X265对应的文档在<https://x265.readthedocs.io/en/master/>,里面command line options可以查到所有的命令行代码

x265环境需要nasm和cmake，之前官网中有提示安装，现在好像没找到（越来越不友好了）

cmake:<https://cmake.org/download/>,下载对应版本，一路确定往下直到安装完成,重启电脑后，命令行输入camke出现提示即安装成功

```
C:\Users\dongy>cmake
Usage

cmake [options] <path-to-source>
cmake [options] <path-to-existing-build>

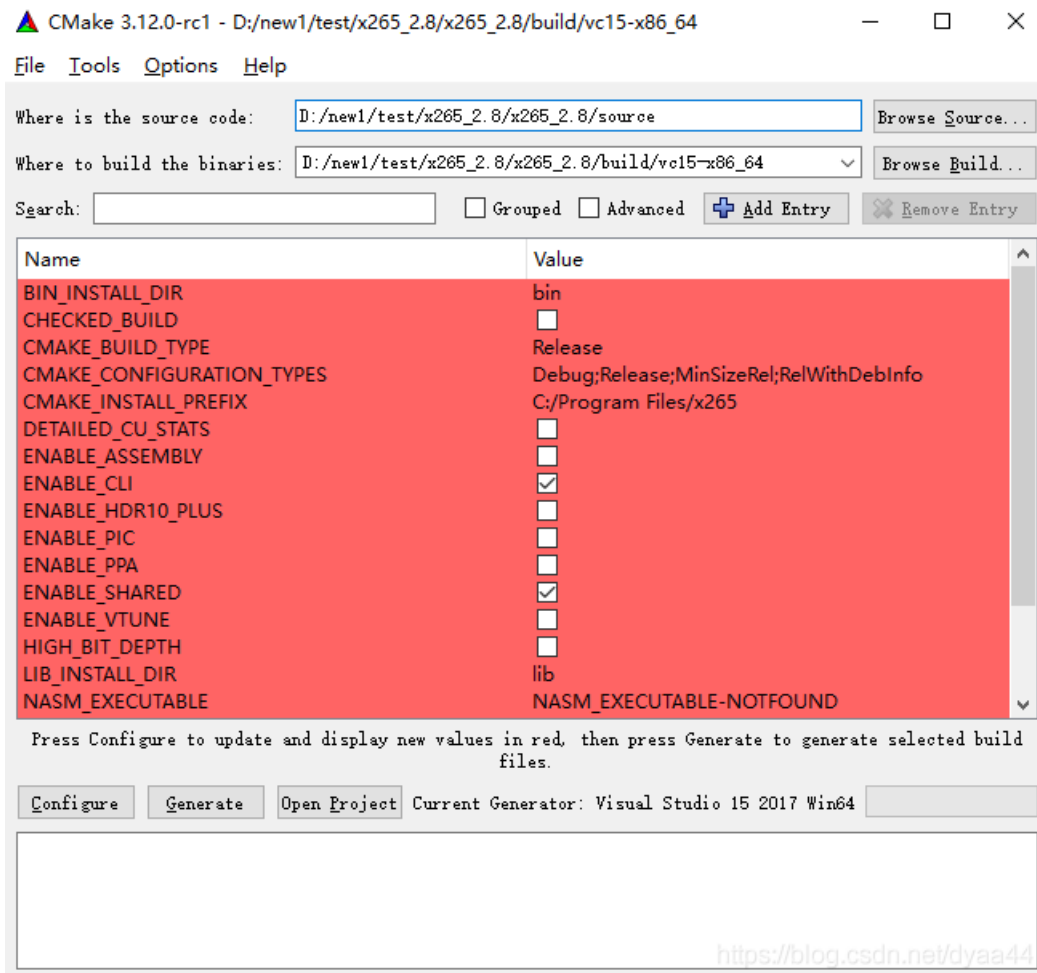
Specify a source directory to (re-)generate a build system for it in the
current working directory.  Specify an existing build directory to
re-generate its build system.

Run 'cmake --help' for more information.

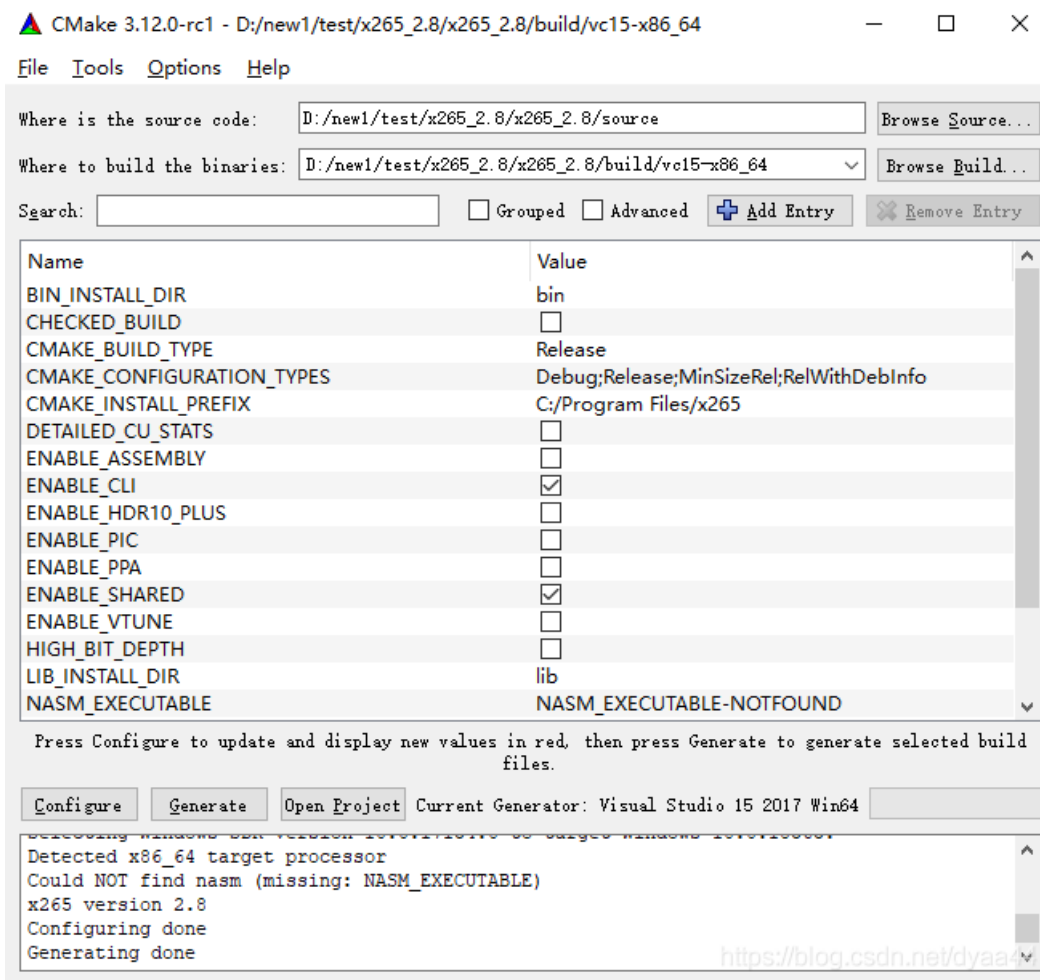
https://blog.csdn.net/dyaa44
```

nasm:<https://www.nasm.us/pub/nasm/releasebuilds/?C=M;O=D>,进入对应版本目录，根据电脑选win32/win64，下载exe，一路确定完成

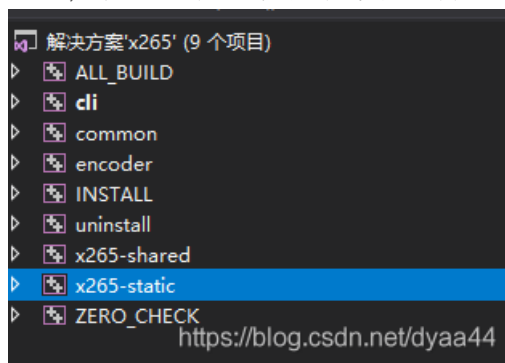
环境完成后，解压x265_2.8，进入build文件夹，选择你对应的VS版本号的进入，例如我安装的是VS2017，64bit则进入vc15-x86_64目录，点击make-solutions.bat.等一会出现以下界面



直接点击Configure，下面提示栏出现Configure done之后 点击Generate,出现Generating done之后直接右上角关闭即可（图中尴尬的出现NOT FIND NASM，不影响后面任何操作）



然后在同一目录下，你会发现他生成了一大堆文件，直接双击打开x265.sln，进入vs2017界面。右键点击CLI，设为启动项目，如下图所示（即CLI变粗即可，默认是allbuild），右键cli->生成，即可生成项目，形成exe。



为了以后在VS里面进行单步调试，我们需在VS中用调试模式进行运行。右键cli->属性，出现的界面中配置属性->调试，工作目录那栏看到工作目录，默认应该是x265_2.8/build/vc15-x86_64这个文件夹，即以后所有的相对路径的根目录是这个目录。

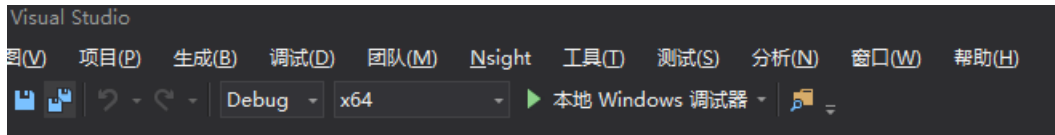
我们将SlideEditing_1280x720_30.yuv拷入这个目录，当然你放在其他地方给绝对路径也可以。

在右键cli->属性，出现的界面中配置属性->调试命令参数那行，输入以下参数

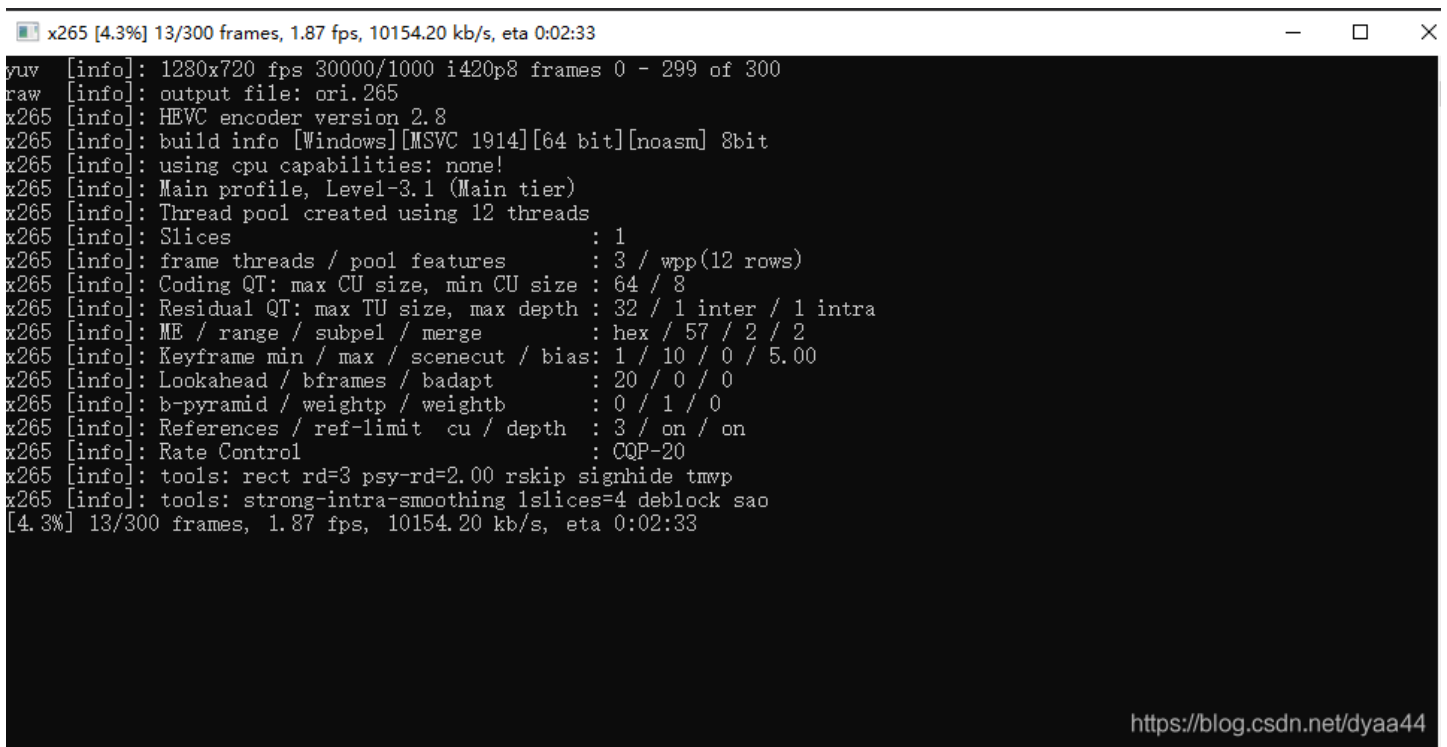
```
-p 5 --rect --input ./SlideEditing_1280x720_30.yuv --input-res 1280x720 --fps 30 --output ori.265 --profile main --keyint 10 --no-scenecut --b-adapt 0 --bframes 0 --qp 20
```

再说一次，任何命令行参数都可在官方文档查到<https://x265.readthedocs.io/en/master/>,command line options

点击本地window调试器，进入调试运行。

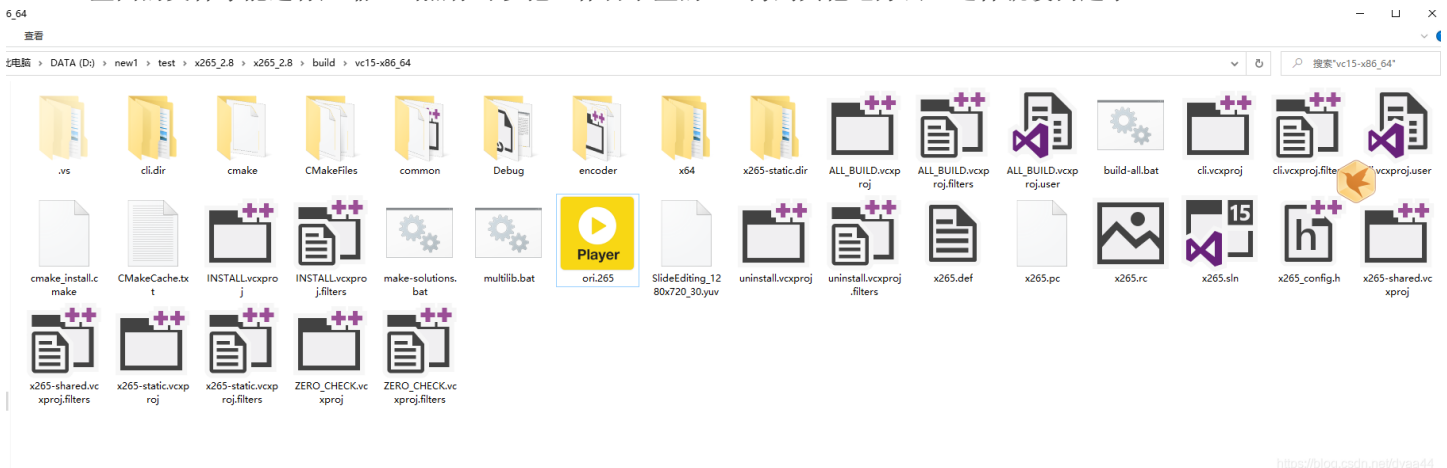


出现以下命令即压缩过程成功



最终在工作目录下面生成ori.265。x265初步运行即成功。ori.265可以直接用potplayer观看，也可用eleccard进行查看参数

注意事项：勿在工作目录中的ori.265和yuv被其他程序占用时进行压缩，会报错，特别是之后用eleccard看265时，需要关闭eleccard里面的文件才能进行压缩。当然你可以把工作目录里的265拷到其他地方去，这样就没问题了。



2.elecard HEVC analyzer的基本使用

从官网上或开头的网盘中下载elecard软件，打开。

左上角file->open media,找到你刚才生成的ori.265

The screenshot displays the Elecard HEVC Analyzer v.1.14 interface. The top bar shows the file path: D:\new1\test\v265_2.8\build\v15-x86_64\ori.265. The main window is divided into several sections:

- prediction**: A table showing metrics for various prediction modes. The 'stream %' column is highlighted in blue.
- decoded**: A sequence of video frames from 0/0 to 28/28, with frame 10/10 highlighted in red.
- ctu presenter**: A table showing CTU (Coding Tree Unit) statistics.
- stream**: A table showing stream statistics for different resolutions and prediction modes.

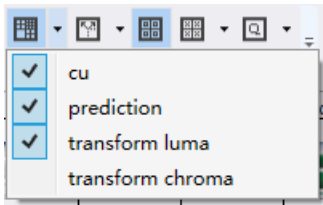
Overlaid on the interface are three documents:

- LibreOffice Writer**: A document titled "MPEG-4 part 2 Video" containing text about copyright, submission procedures, and test material requirements.
- Video 3.odp - LibreOffice Impress**: A presentation slide titled "MPEG-4 part 2 Video" with bullet points regarding the text of 14496-5:2001/Amd.5 DCOR1 (N11614) and the replacement of software.
- LibreOffice Writer**: A document titled "MPEG-4 AVC" containing text about defect reports and editorial work.

一般，HEVC隐写分为帧内隐写和帧间隐写，帧内隐写的常见载体有，帧内预测模式，帧内的DCT系数，帧内的分块结构。帧间隐写有运动向量，分块结构等等。下面介绍一下常见的这些参数在哪里看。

2.1 CUPUTU划分

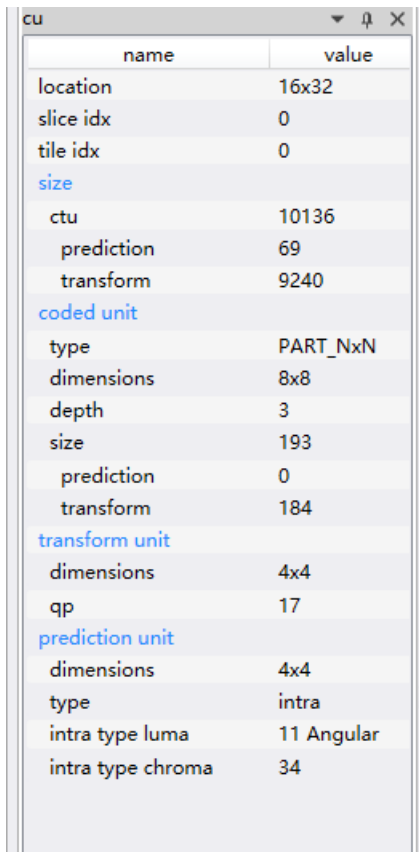
软件上方按此结构选择



这样，橘黄色代表帧内预测块，无色代表skip，蓝色代表inter预测块。示例中序号7这一帧看到三种块同时出现。

右侧cu栏显示了我们鼠标指向的CU对应的结构。

根据不同的算法，我们可以对type,CU的大小进行任意修改，从而让整个CTU的划分按我们想要的隐写后的结构来。具体后面文章进行示例。



name	value
location	16x32
slice idx	0
tile idx	0
size	
ctu	10136
prediction	69
transform	9240
coded unit	
type	PART_NxN
dimensions	8x8
depth	3
size	193
prediction	0
transform	184
transform unit	
dimensions	4x4
qp	17
prediction unit	
dimensions	4x4
type	intra
intra type luma	11 Angular
intra type chroma	34

比如第一帧这个位置，是个8x8的CU划分成了4个4x4的PU和TU

2.2 帧内预测模式

如上图，我们可以看到这一块CU中我们鼠标指向的PU的intra type luma是11，即帧内方向是11，隐写需要改的就是这个11。一般不修改色度块，只改亮度块。

2.3 运动向量

同样，右侧也显示了对应的运动向量信息，如下图，这一块是一个merge模式的，运动向量是0,-16的PU块，CU划分模式是NX2N，即竖着划。我们隐写需要改的就是这个0,-16，当然，一般不改merge模式块下的运动向量，而是改非merge的运动向量。

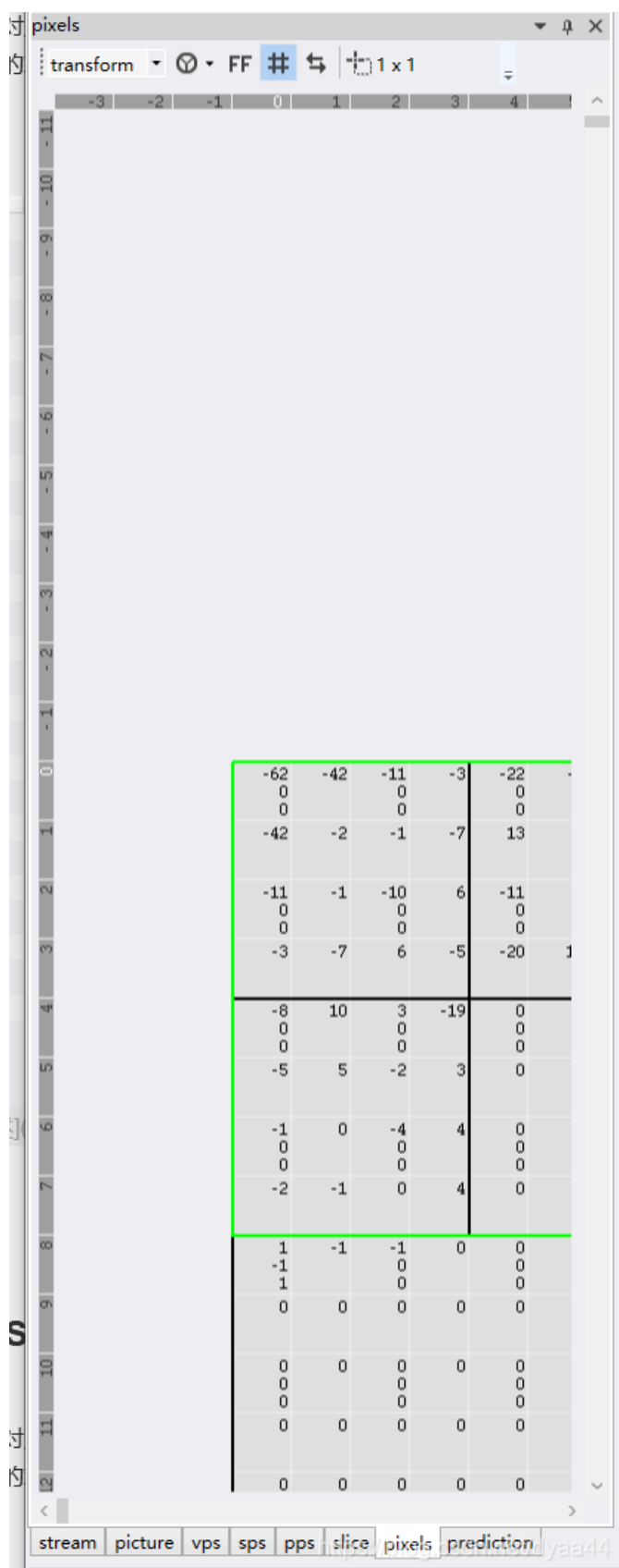
name	value
location	704x448
slice idx	0
tile idx	0
size	
ctu	16
prediction	8
transform	4
coded unit	
type	PART_Nx2N
dimensions	32x32
depth	1
size	8
prediction	4
transform	2
transform unit	
dimensions	32x32
qp	20
prediction unit	
dimensions	16x32
merge_flag	1
merge_idx	1
inter type	0 (Pred_L0)
L0 mv	0, -16, 0
candidates	
[0] LEFT	-
mvL0	0,-28,0
[1] ABOVE	
mvL0	0,-16,0

2.4 DCT/DST系数

我们可以在左侧pixels栏，上方下拉transform看到所有量化后的DCT/DST系数（解码端），一般进行DCT/DST隐写即修改这部分参数。

每个小白块有1-3个值，第一行是我们需要修改的亮度DCT/DST的值，如第一个-62。后面的两个0,0代表U和V通道的值，可以发现这个YUV是个YUV420格式的YUV，4个方形块共享一个U和V的值。UV值一般我们不关注，隐写只修改Y分量，即第一行的DCT/DST值。

这一栏还可以看到好多其他的数值，如解码端的残差和解码值。



基本工具就介绍到这里，之后介绍如何在X265中进行以上各个参数位置的隐写算法的基本修改方式。