

## 转 H.264参考软件JM12.2RC代码详细流程

2013年09月24日 16:14:28 阅读量：7661

代码：JM12.2. 编码结构：IPP。GOP：3。  
跟踪代码：（注：表示编码主流程，表示RC流程）

进入 main() ,首先调用 init\_global\_buffers()。进入init\_global\_buffers(),调用两个函数 generic\_alloc( &generic\_RC ) //Dynamically allocate memory needed for generic rate control 和 rc\_alloc( &quadratic\_RC ) //Dynamically allocate memory needed for rate control。然后进入 rc\_init\_seq( rc\_quadratic \*prc ) //Initialize rate control parameters。这三个函数在整个流程中只调用一次。

编码循环开始。调用 rc\_init\_GOP( rc\_quadratic \*prc, int np, int nb ) //Initialize one GOP。

开始编码第一帧。I帧。

进入 encode\_one\_frame() ,调用 rc\_init\_pict( rc\_quadratic \*prc, int fieldpic,int topfield,int targetcomputation, float mult) //Initialize one picture。接着调用 updateQP( quadratic\_RC, 0 ) //compute a quantization parameter for each frame.调用 frame\_picture() //Encodes a frame picture。进入frame\_picture(), 调用 code\_a\_picture() //Encodes a picture。进入code\_a\_picture(),调用 encode\_one\_slice() //Encodes one slice。进入encode\_one\_slice(),调用 encode\_one\_macroblock() //Mode Decision for a macroblock, 进入encode\_one\_macroblock(), 调用 update\_rc() //Update Rate Control Parameters, 接着调用 handle\_qp( Macroblock \*currMB, short best\_mode ) //Update QP Parameters (critical in case of SKIP MBs or MBAFF)。

编码完一帧以后调用 rc\_update\_pict\_frame( rc\_quadratic \*prc, int nbits ) //update after frame encoding。最后调用 rc\_update\_pict( rc\_quadratic \*prc, int nbits ) //update one picture after frame/field encoding。

第一帧I帧编码完成。

开始编码第二帧。第一个P帧。

循环再次调用 encode\_one\_frame()。现在编码的是P帧。调用 rc\_init\_pict(),接着调用 updateQP(), 进入updateQP(),调用 updateFirstP( rc\_quadratic \*prc, int topfield ) //计算第一个P帧的量化参数。调用 frame\_picture()。进入frame\_picture(), 调用 code\_a\_picture()。进入code\_a\_picture(),调用 encode\_one\_slice()。进入encode\_one\_slice(), 对宏块进行编码循环。先调用 start\_macroblock(), 如果已编码宏块数大于0并且是基本单元宏块数的整数倍, 则调用 updateRCModel( rc\_quadratic \*prc ) //update the parameters of quadratic R-D model。进入updateRCModel(),调用 QP2QStep(),接着调用两次 RCModelEstimator()。之后调用 updateMADModel, 然后调用 updateQP()。进入updateQP(),调用 updateFirstP()。然后调用 encode\_one\_macroblock(), 进入encode\_one\_macroblock(), 调用 update\_rc() 接着调用 handle\_qp()。

\*\*\* 需要注意的一点：对于第一个基本单元, 进入updateRCModel之后, 在调用完两次RCModelEstimator之后, 没有调用updateMADModel。从第二个基本单元开始, 才进入updateMADModel。进入updateMADModel以后, 会调用MADModelEstimator。并且, 编码第一个P帧时调用updateMADModel没有任何作用。只有在编码第二个P帧开始, updateMADModel才会起作用。\*\*\*

编码完一帧以后调用 rc\_update\_pict\_frame。接着调用 rc\_update\_pict()。调用 updateRCModel(),进入 updateRCModel(),调用两次 RCModelEstimator()。之后调用 updateMADModel。进入 updateMADModel, 这个时候updateMADModel函数会起作用了。它调用 MADModelEstimator, 更新参数之后并再次调用 MADModelEstimator。

第一个P帧编码完成。

开始编码第三帧, 即第二个P帧。

调用 encode\_one\_frame()。调用 rc\_init\_pict(),接着调用 updateQP(), 进入updateQP(),由于是第一个基本单元, 调用 updateFirstBU( rc\_quadratic \*prc, int topfield )。之后调用 frame\_picture()。进入frame\_picture(), 调用 code\_a\_picture()。进入code\_a\_picture(),调用 encode\_one\_slice()。进入encode\_one\_slice(), 对宏块进行编码循环。调用 encode\_one\_macroblock(), 进入encode\_one\_macroblock(), 调用 update\_rc() 接着调用 handle\_qp()。编码的宏块数超过一个并且是基本单元宏块数的整数倍时, 调用在 start\_macroblock 里面的 updateRCModel。进入updateRCModel(),调用 QP2QStep(),接着调用两次 RCModelEstimator()。之后调用 updateMADModel, 进入updateMADModel,调用两次 MADModelEstimator。然后调用 updateQP()。进入updateQP, 如果Target<0, 调用 updateNegativeTarget( rc\_quadratic \*prc, int topfield, int m\_Qp ), 否则会：调用 predictCurrPicMAD, 接着调用 updateModelQPBU( rc\_quadratic \*prc, int topfield, int m\_Qp), 如果是最后一个基本单元, 则调用 updateLastBU()。

编码完一帧以后调用 rc\_update\_pict\_frame。接着调用 rc\_update\_pict()。调用 updateRCModel(),进入 updateRCModel(),调用两次 RCModelEstimator()。之后调用 updateMADModel。进入 updateMADModel, 这个时候updateMADModel函数会起作用了。它调用 MADModelEstimator, 更新参数之后并再次调用 MADModelEstimator。

以上是我跟踪代码所得。比较简单, 但是仍希望对大家有帮助。

文章标签：

JM

代码

流程

H.264

个人分类：JM

视频编码

此PDF由spyyg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com