

微纳系统综合设计课程作业 结题答辩

Guetzli图像压缩算法

罗恬 刘健伟 杨文曦

January 10th, 2020
Shanghai Jiao Tong University

微纳系统综合设计课程作业 结题答辩

图像压缩算法

罗恬 刘健伟 杨文曦

January 10th, 2020
Shanghai Jiao Tong University

目录

- 内容： 尝试和最终选择
- 实现： 算法， HLS与嵌入式
- 结果与分析： 压缩结果与HLS优化
- 创新点
- 分工



内容

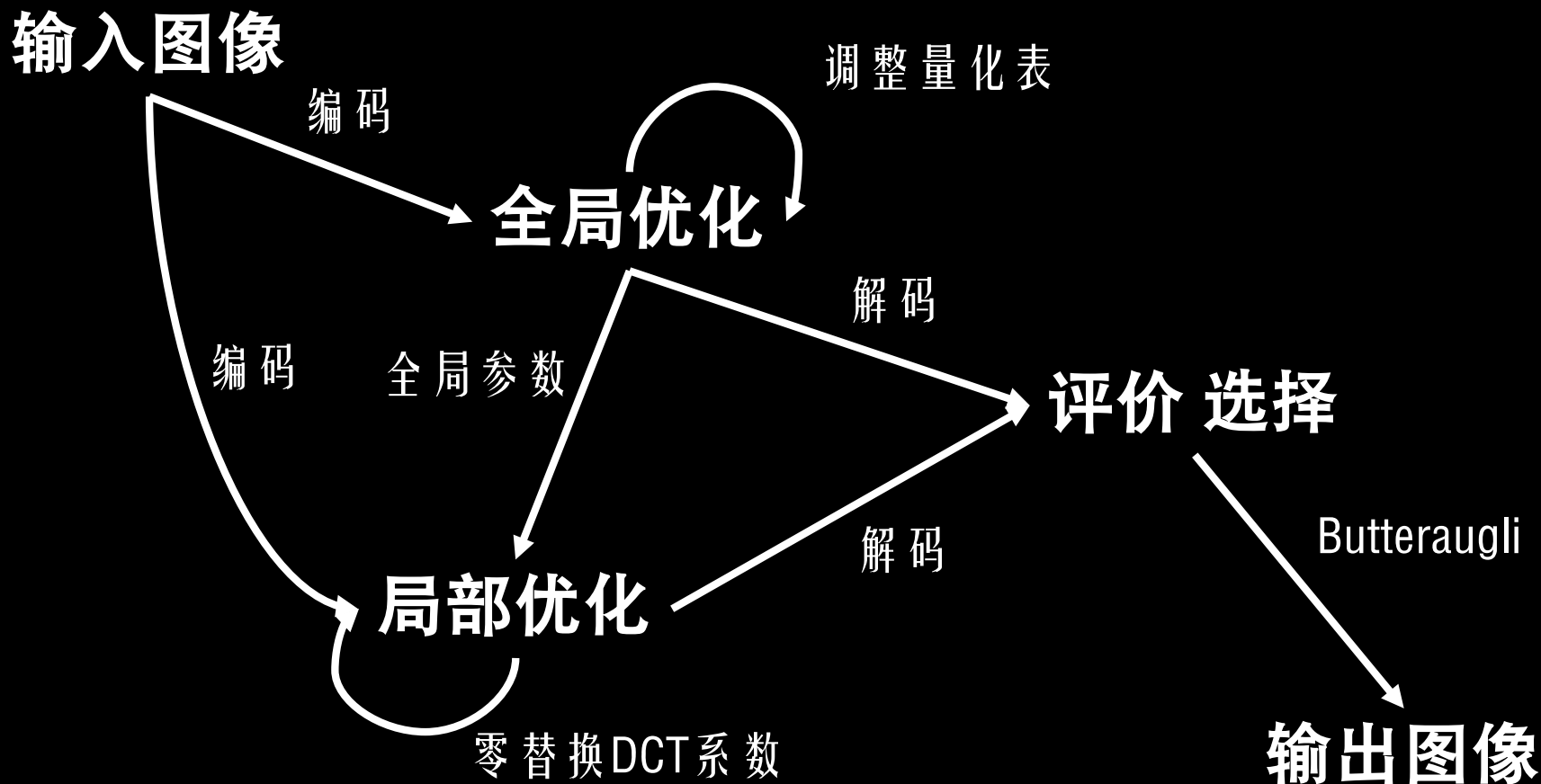
最终选择并实现的算法

前期尝试实现的算法

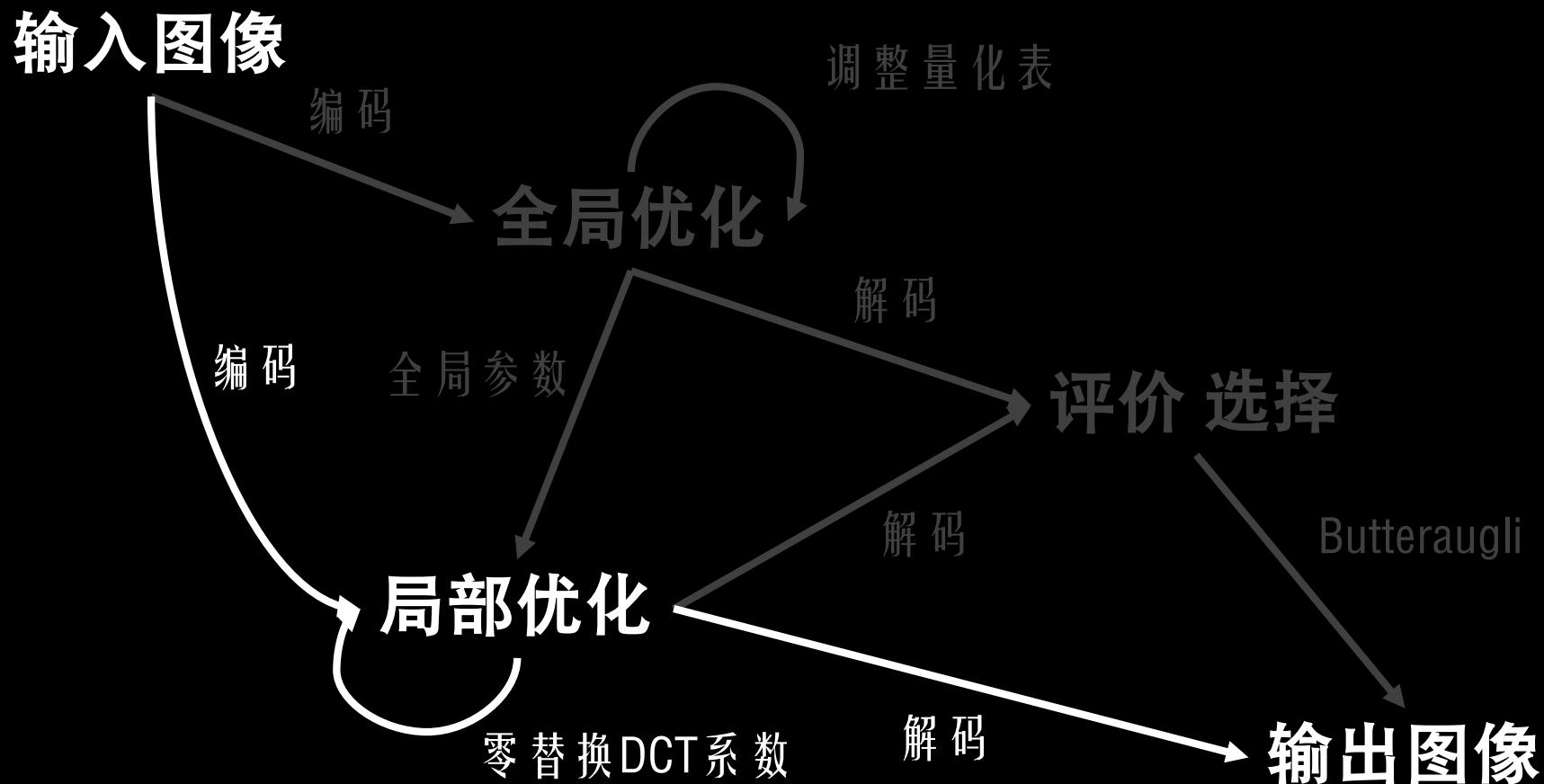


前期尝试

Guetzli图像压缩算法^{[1][2]}



修改 Guetzli图像压缩算法



原因

为什么没有选择Guetzli

代码量过大

46个文件，共256KB

C++版本问题

大量代码需要修改



所需资源过多

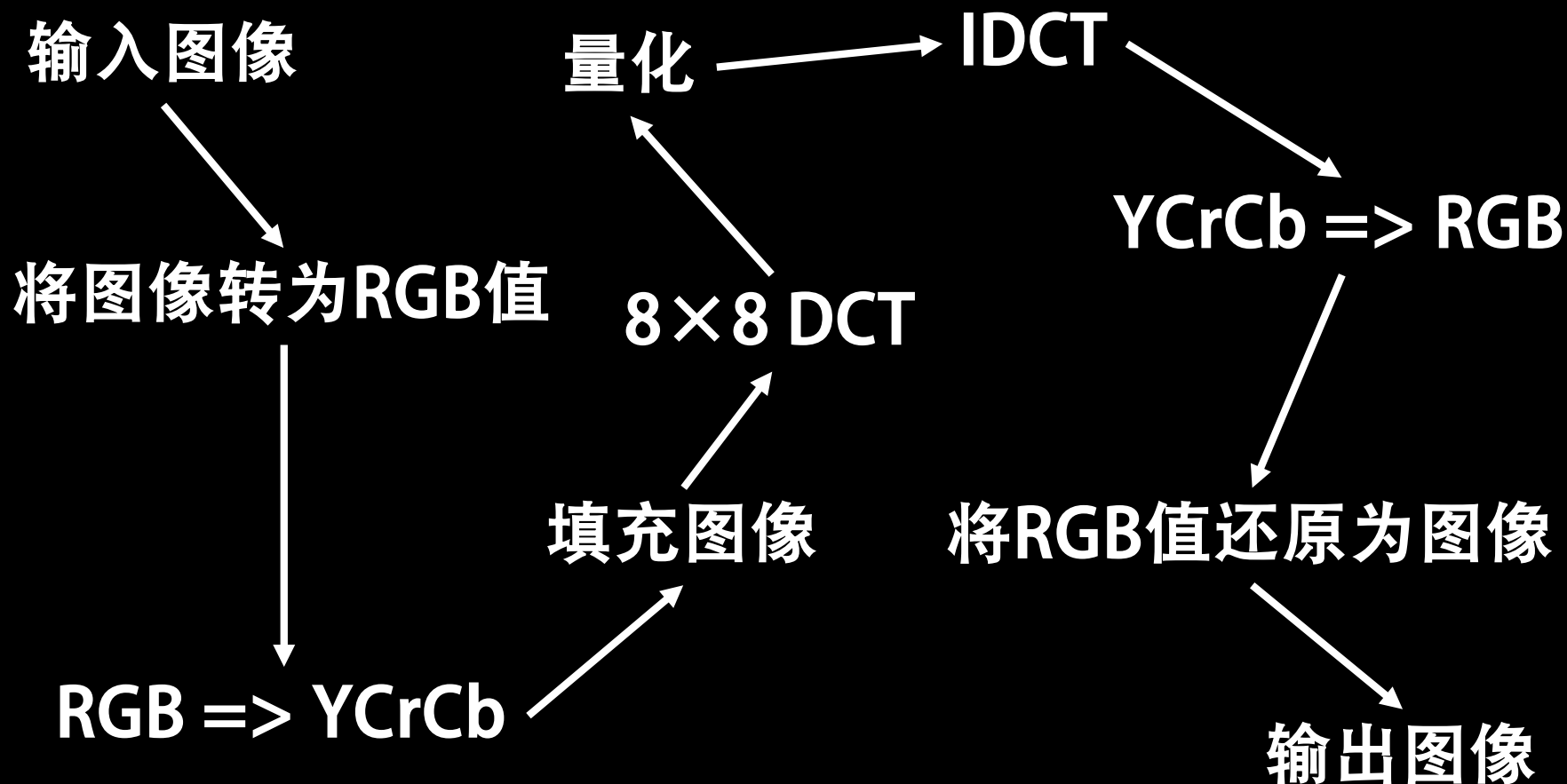
预计板上资源不够

能力不足

遇到一些代码编写上的问题

最终实现算法

图像压缩算法^{[3][6][7]}

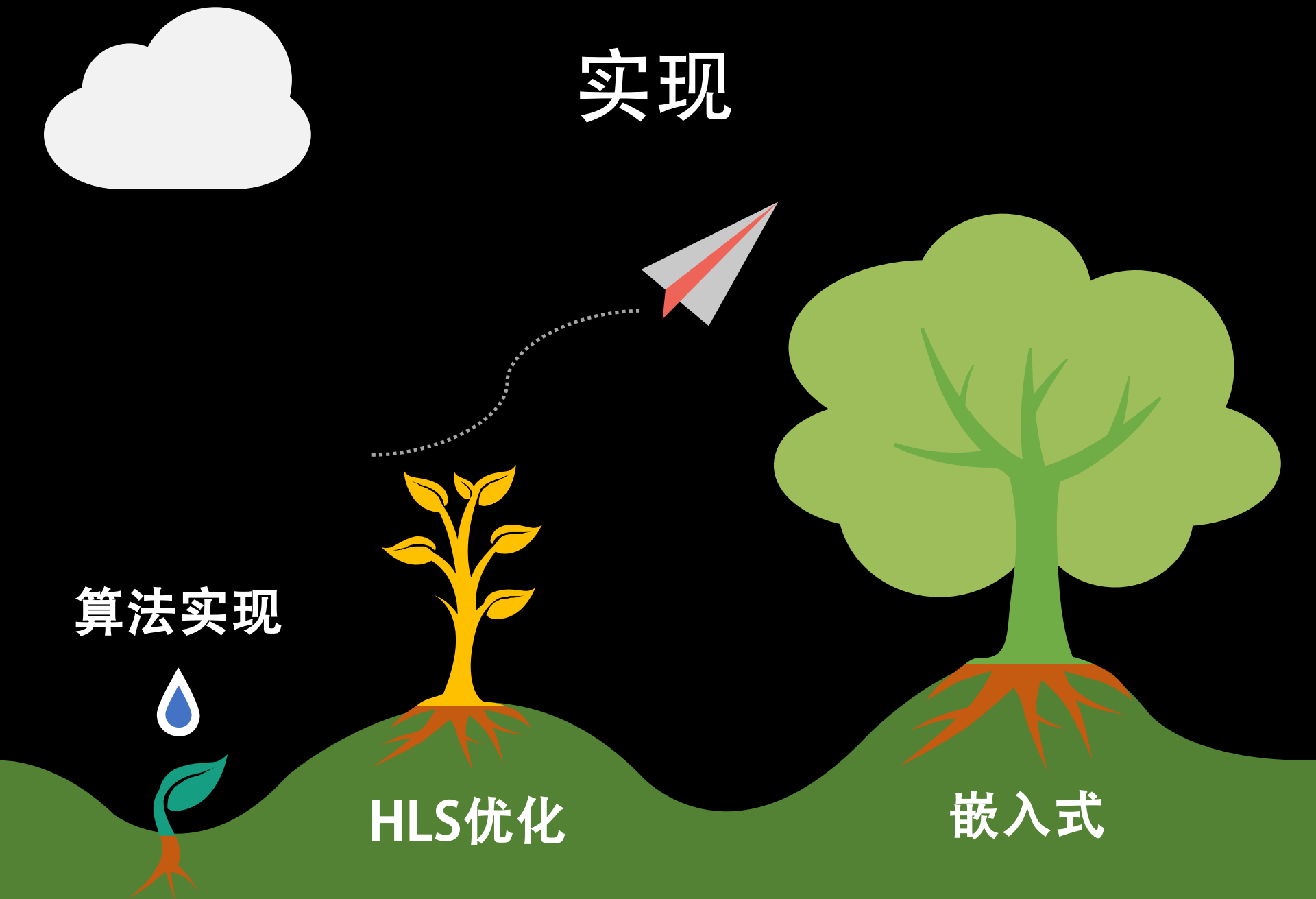


实现

算法实现

HLS优化

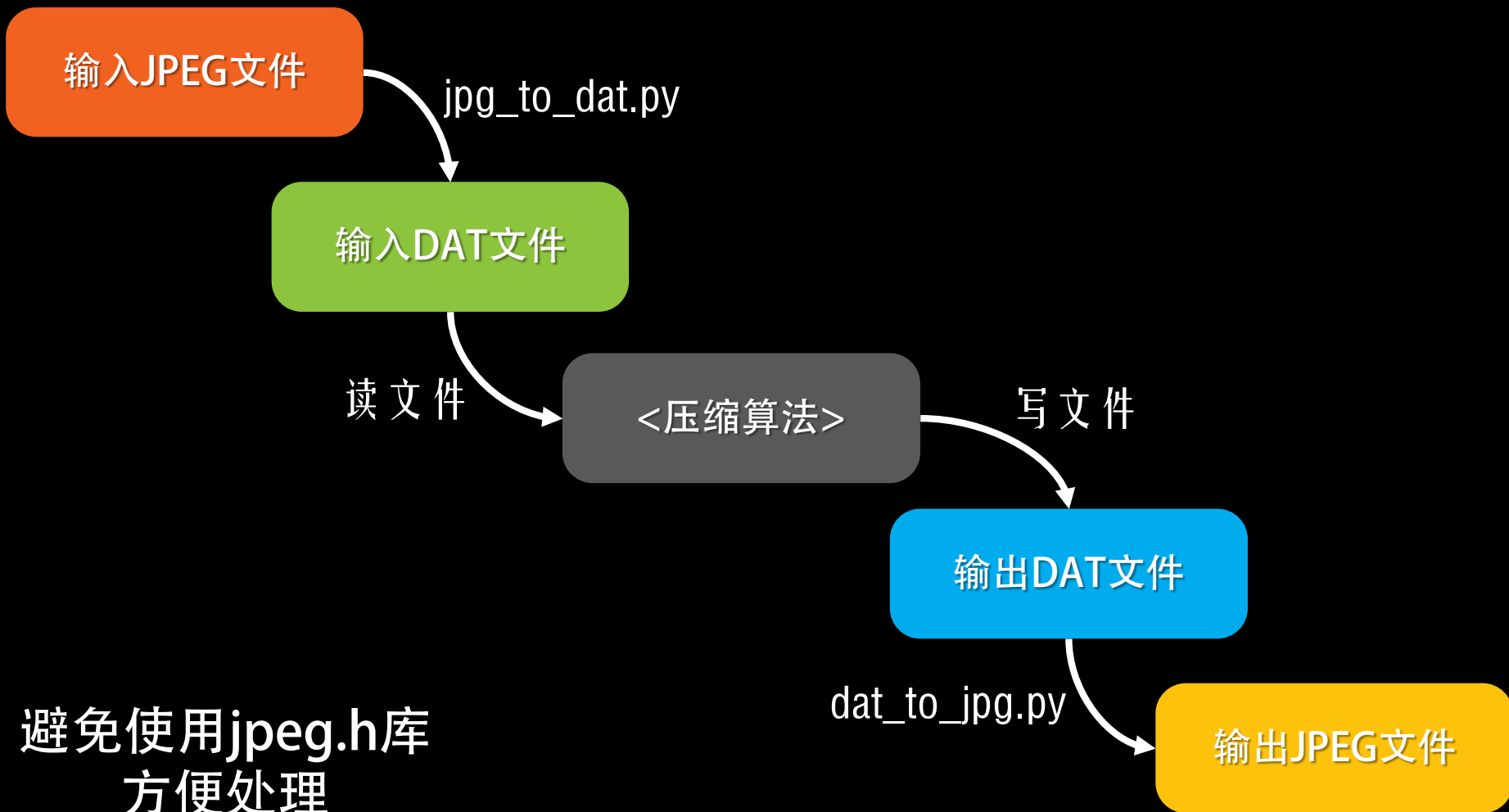
嵌入式



负责人：杨文曦

预处理

图像与RGB值之间的转换



负责人：杨文曦

预处理

图像与RGB值之间的转换^[4]

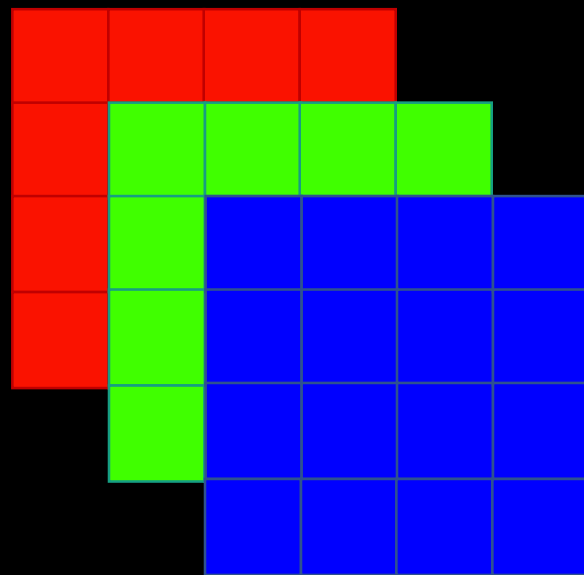
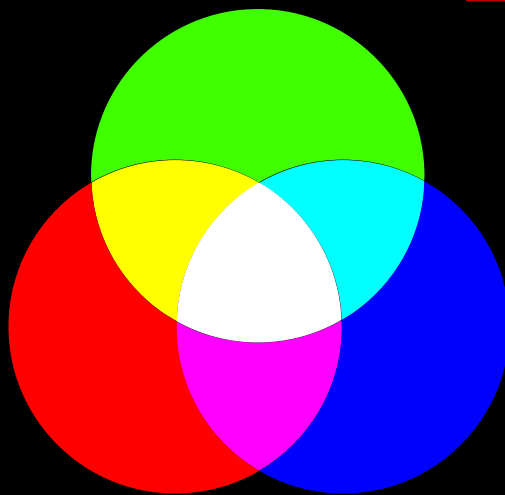
DAT文件格式

Line 1: 图像的宽和高
(Pixels)

Others: 红绿蓝三色亮度
(0-255)

DAT文件格式示例

```
392 392
141 135 82
165 159 106
137 131 78
189 183 130
... ..
```

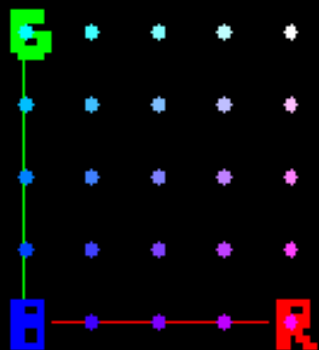


使用OpenCV库在Python中实现

算法实现

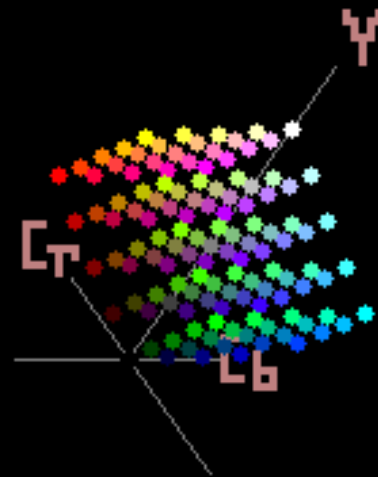
RGB与YCrCb色彩空间之间的转换^[4]

RGB色彩空间



R: 红色亮度
G: 绿色亮度
B: 蓝色亮度

灰阶-色度色彩空间^[3]



Y : 灰阶/色度
Cr : 红色浓度
Cb : 蓝色浓度

算法实现

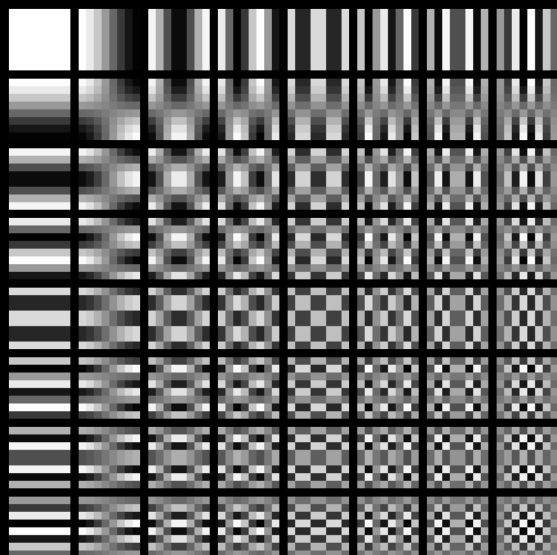
RGB与YCrCb色彩空间之间的转换

$$\left\{ \begin{array}{l} Y = 16 + \frac{65.738R}{256} + \frac{129.057G}{256} + \frac{25.064B}{256} \\ C_B = 128 - \frac{37.945R}{256} - \frac{74.494G}{256} + \frac{112.439B}{256} \\ C_R = 128 + \frac{112.439R}{256} - \frac{94.154G}{256} - \frac{18.285B}{256} \end{array} \right. \quad [4]$$

负责人：罗恬

算法实现

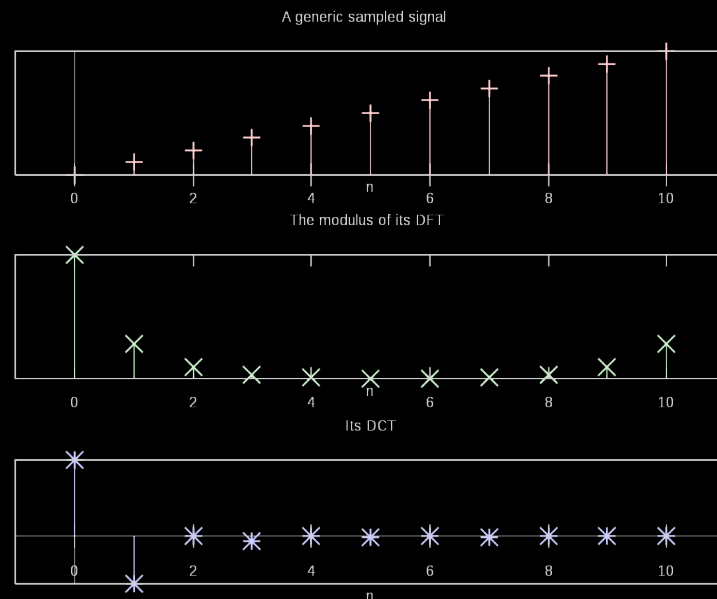
归一化二维离散余弦变换^[5]



↑
最终
输出

↑
加权
函数

↑
当前
函数



$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

负责人：罗恬

算法实现

量化矩阵^[5]

$$B_{j,k} = \text{round} \left(\frac{G_{j,k}}{Q_{j,k}} \right) \text{ for } j, k = 0, 1, 2, \dots, 7$$

G：未量化的DCT系数

Q：量化矩阵

B：量化的DCT系数

round()：取整

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

将高频元素(空间频率较大)量化为0

负责人：刘健伟

算法实现

HLS优化：Directive

类型	对应操作
函数	HLS PIPELINE
for循环	HLS UNROLL
多维数组	HLS ARRAY_PARTITION
设置中间变量	对中间变量进行进一步优化

负责人：罗恬

算法实现 嵌入式实现

目标效果

通过SDKFPGA开发板上运行算法
证明板上运行能达到更快的速度

未实现！

原因：不能产生对应的IP核
函数改写后出现一些

结果与分析

原始图像

697KB

1184×1184



预处理

87.9KB

392×392

Q = 80

27.0KB

392×392

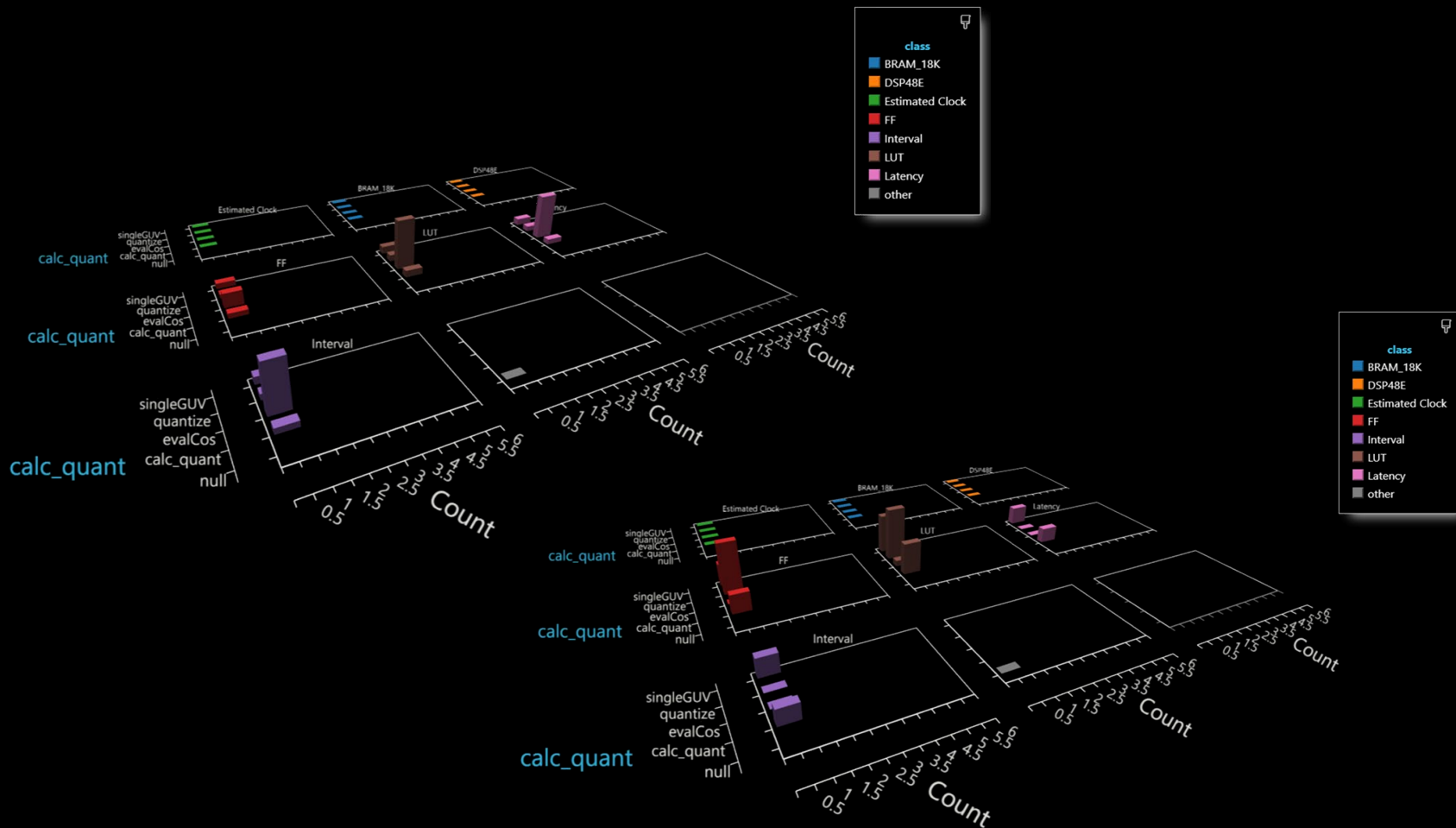


Q = 60

12.1KB

392×392

结果与分析



结果与分析

函数	指标	优化前	优化后
quantize()	Latency	2001	63
evalCos()	Clock	7.76	2.71
evalCos()	FF	5428	32
evalCos()	LUT	18893	692
evalCos()	Latency	17137	31

创新点

实际意义

有一定实际意义的可运行的
图像压缩算法

速度提升

运用HLS优化
对代码进行提速



研究意义

对图像压缩算法和HLS中代码规范
有了更深入的了解

分工

 杨文曦

 罗恬

输入图像

将图像转为RGB值

RGB => YCrCb

IDCT

量化

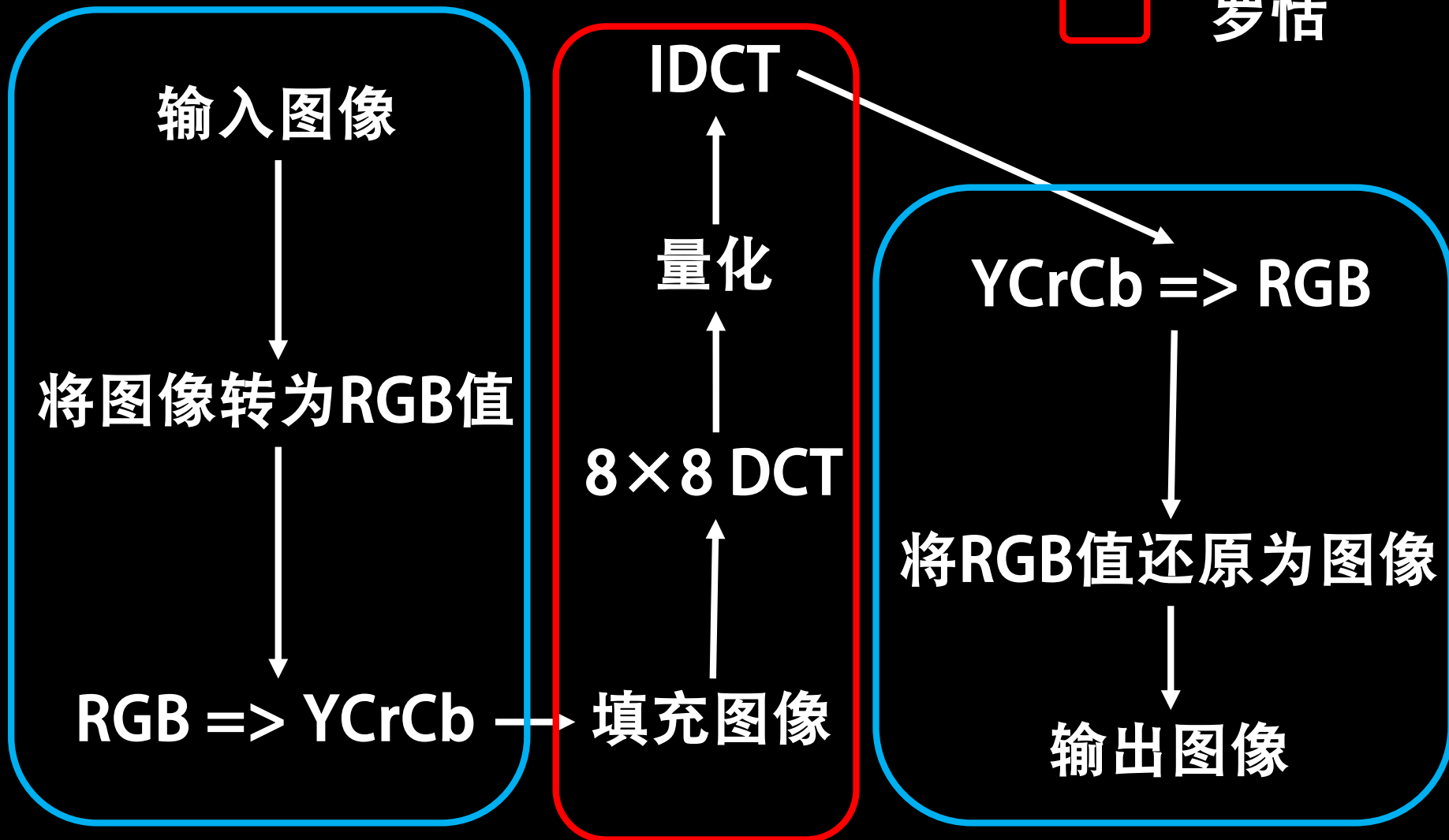
8×8 DCT

填充图像

YCrCb => RGB

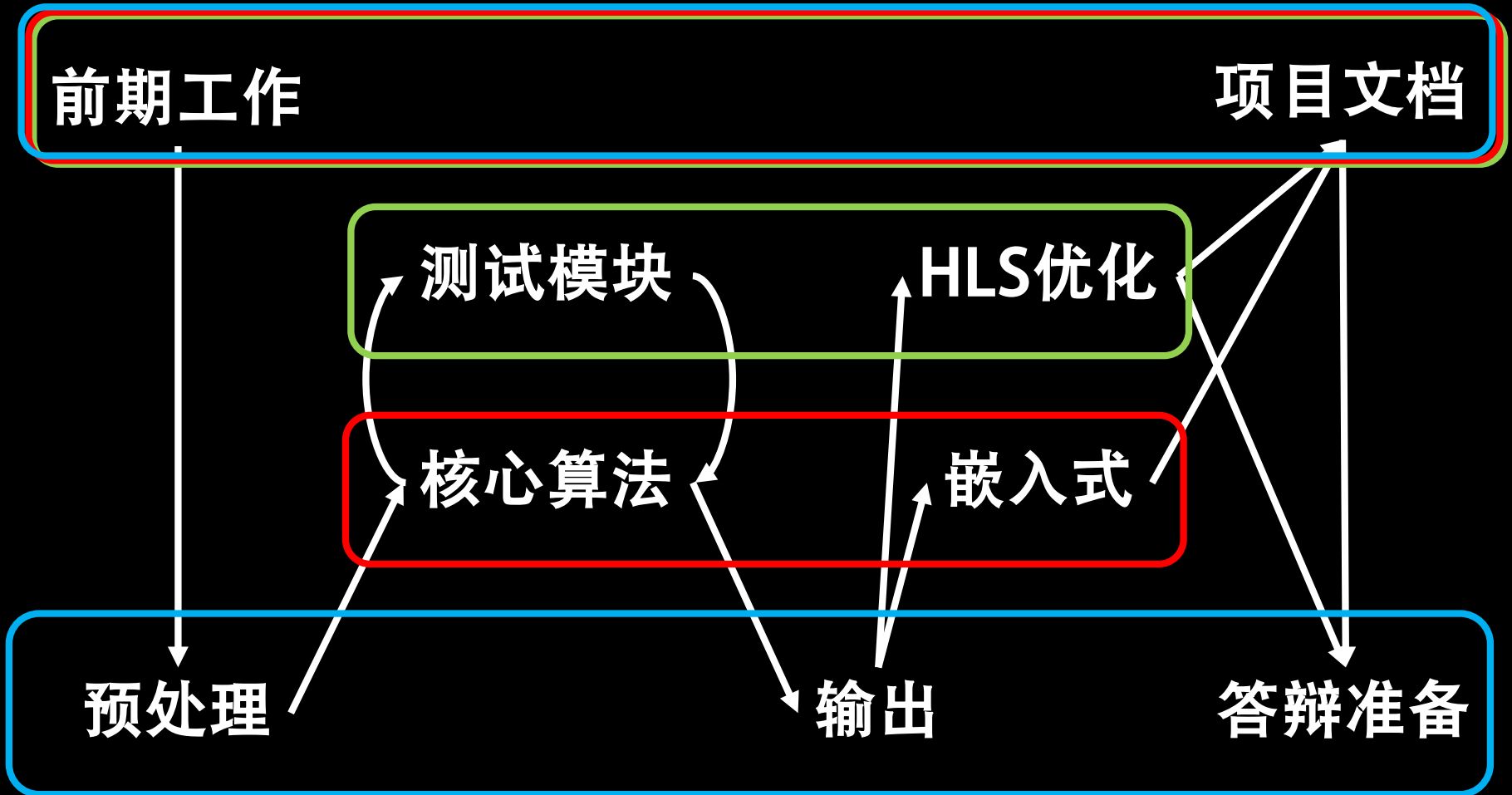
将RGB值还原为图像

输出图像



分工

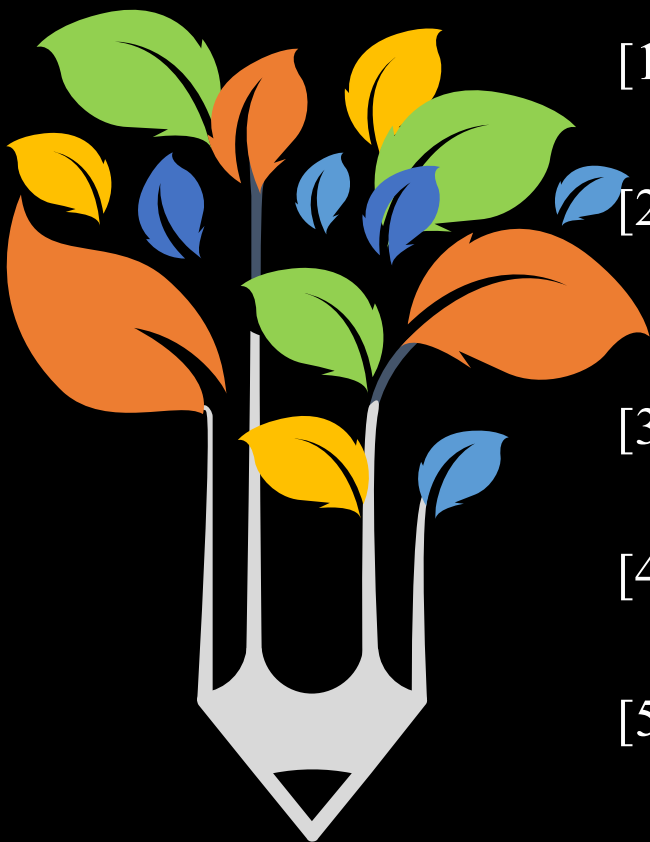
 杨文曦 罗恬
 刘健伟



分工

组员	前期	主要项目	后期	
杨文曦	文献阅读 代码修改	预处理 色域转换 debug	答辩材料	报告撰写
罗恬		DCT/IDCT 辅助函数 (拓展, 量化) 嵌入式修改 debug	继续尝试嵌入式	
刘健伟		testbench HLS优化	HLS优化总结	

参考资料



[1] Guetzli, <https://github.com/google/guetzli>

[2] Alakuijala, Jyrki, et al. "Guetzli: Perceptually guided jpeg encoder." preprint arXiv: 1703.04421 (2017).

[3] YCbCr, <https://en.wikipedia.org/wiki/YCbCr>

[4] RGB to YCbCr, <https://sistenix.com/rgb2ycbcr.html>

[5] Discrete cosine transform, https://en.wikipedia.org/wiki/JPEG#Discrete_cosine_transform

参考资料



- [6] Fu, Sizhe, Ping Shi, and Da Pan. "A modified algorithm of Guetzli encoder." 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018
- [7] Hudson, Graham, et al. "JPEG-1 standard 25 years: past, present, and future reasons for a success." Journal of Electronic Imaging 27.4 (2018): 040901.



Thanks for watching.

January 10th, 2020
Shanghai Jiao Tong University