

编码引论第三次仿真实验报告

无35 陈馨瑶 2013011166

2015年12月28日

1 分工

本次实验的分工如下：

1.1 PART I 量化

- 画 R-D 曲线图。设计 4 个不同步长的均匀量化器，将其比特率，PSNR 绘制在 R-D 图中，用线段连接。横轴为比特率，纵轴为 PSNR。
- 练习 JPEG/H.261 量化器，绘制 R-D 图。
- 设计非均匀量化器，绘制 R-D 图。
- 对三类量化器进行评价。

1.2 PART II 变长编码（独立符号）

- 设计变长编码器，用变长码对量化后的图象编码。输入符号（象素）进行独立编码。
- 给出编码前后的比特数，计算压缩比。

1.3 PART III 变长编码（两符号联合）

- 设计变长编码器，用变长码对量化后的图象编码。输入符号（象素）进行独立编码。
- 给出编码前后的比特数，计算压缩比。

其中，由我完成的是第二部分：变长编码（独立符号），由于各部分实验内容较为独立，故而在模块实现部分仅对我自己完成的内容进行了总结。

2 模块实现

所采用的熵编码为Huffman码，原理在书上和课件上都有所讲解。实验时，统一均匀量化，步长为20，则量化后的灰度值只可能是10, 30, 50……Huffman码利用不同符号的概率进行编码，在这里的概率即具体图像中不同灰度值所占的比率。为简化计算，我编写了如下的程序计算编码：

```
1 close all;
2 clear all;
3 clc;
4
5 img = imread('1.bmp');
6 symbols = [0];
7 p = [0];
8 N = size(img,1)*size(img,2);
9
10 for i = min(min(img)): 20: max(max(img))
11     symbols = [symbols, i];
12     [r, ~, ~] = find(img == i);
13     p = [p, length(r)/N];
14 end
15
16 [dict, avglen] = huffmandict(symbols, p);
```

3 实验结果

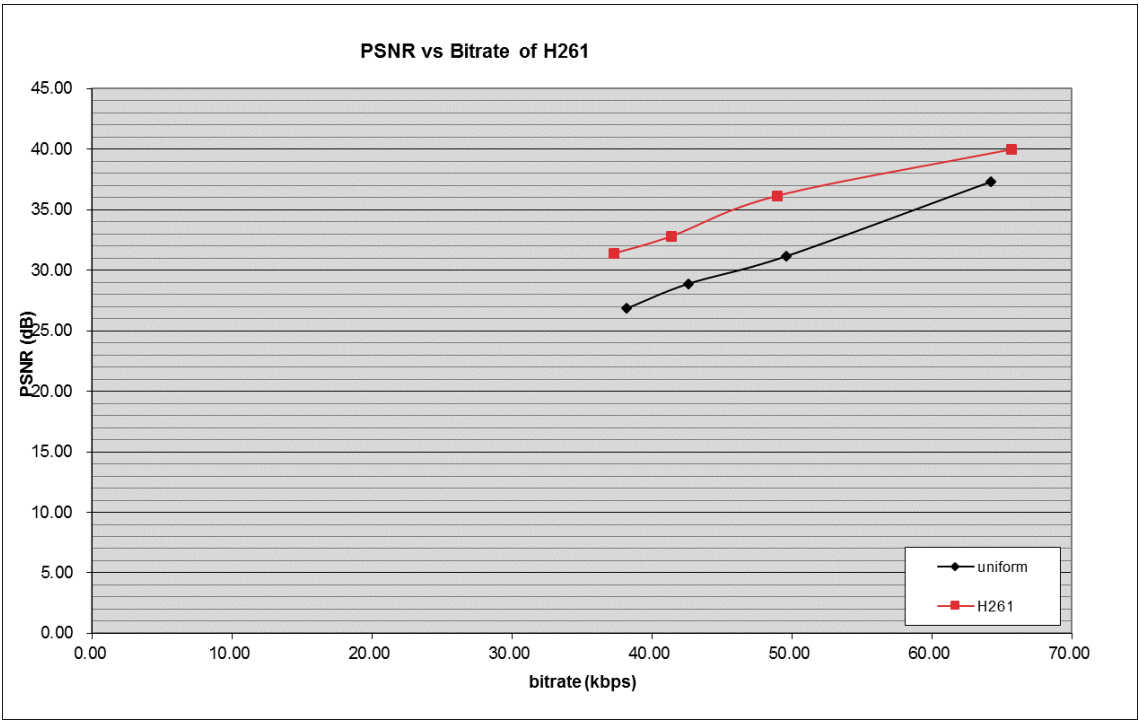
3.1 PART I 量化

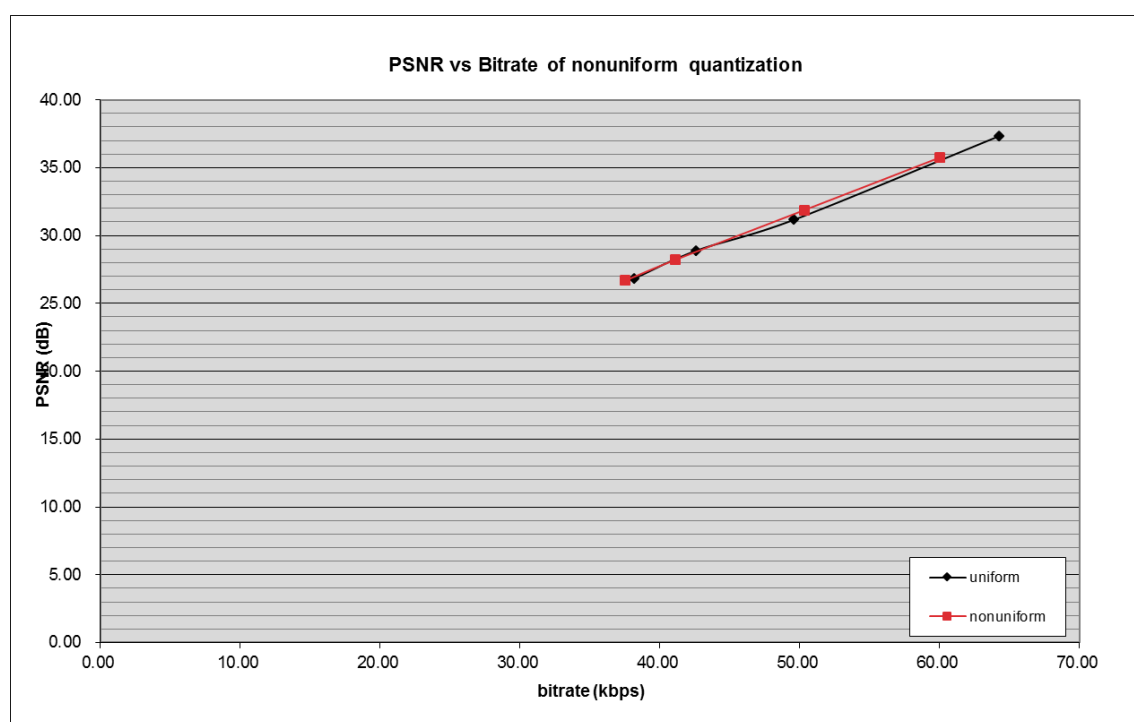
¹用H261，两侧细化和中部细化三种量化器分别和均匀量化进行比较，结果如下：

¹本部分完成人：刘家硕

B	C	D	E	F	G	H	I	J
		均匀量化			other			
	step factor	Bitrate(kbps)	psnr	compress rate	Bitrate(kbps)	psnr	compress rate	BD-rate(%)
vs H261	12 1	64.22	37.32	0.49	65.67	40.00	0.50	-22.5%
	24 2	49.58	31.17	0.38	48.97	36.17	0.37	
	32 3	42.59	28.89	0.32	41.39	32.83	0.32	
	40 4	38.17	26.85	0.29	37.29	31.39	0.28	
vs 两侧细化		64.22	37.32	0.49	59.99	35.75	0.46	-1.4%
		49.58	31.17	0.38	50.30	31.87	0.38	
		42.59	28.89	0.32	41.08	28.21	0.31	
		38.17	26.85	0.29	37.49	26.68	0.29	
vs 中部细化（未画图）		64.22	37.32	0.49	65.57	37.13	0.50	1.4%
		49.58	31.17	0.38	49.26	30.87	0.38	
		42.59	28.89	0.32	42.20	28.65	0.32	
		38.17	26.85	0.29	38.72	27.02	0.30	

H261和两侧细化量化器的R-D曲线分别如下：





据此，对三类量化器的评价如下：

H261比均匀量化好很多，尝试的两种非均匀量化方式和均匀量化相比几乎没有区别，最优的非均匀量化应该是重建数值在量化区间重心，量化区间边界在重建数值中点的情况，由于软件设计的是重建数值为量化区间中值故无法实现最优。

3.2 PART II 变长编码（独立符号）

（）最后得到的VLC码表如下：

```

30 000110
50 010
70 0110
90 0000
110 100
130 001
150 11
170 101
190 00010
210 0111
230 000111
0001111

```

编码前（量化后）和编码后的图像对比如下：



(a) 编码前图像

(b) 编码后图像

图 1: 编码前后图像对比

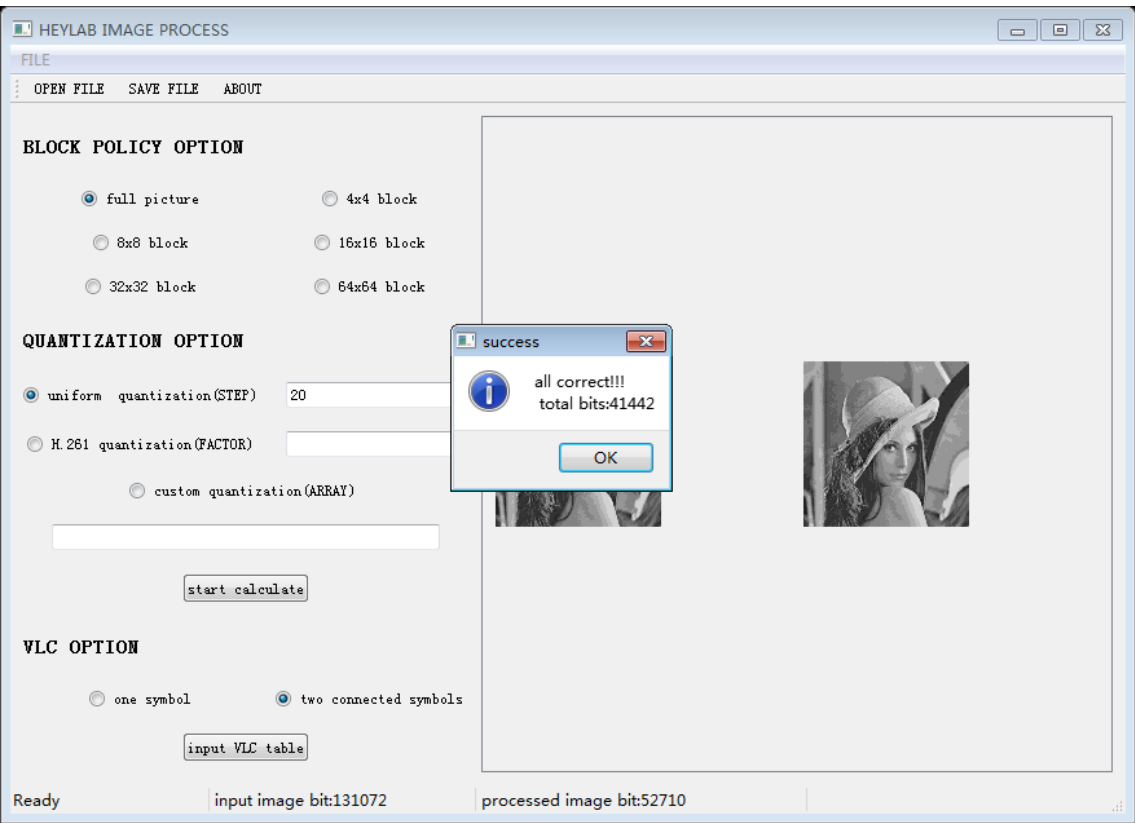
从上图可以看出，解码后的图像成功恢复了原有图像，二者并无差别。编码前图像为131072bit，量化与编码后为52643bit，压缩比为40.16%。

对Huffman码的评价：作为一种熵编码，能在不丢失信息的同时达到这样的压缩比，说明这种编码是较为有效的。但我认为其可能存在问题在于实现上，由于每一步都需要对符号的概率进行排序，在信源符号数较多时会带来较大的计算复杂度，并且要直到最后一步才能向前倒推编码，所以在实现时也存在存储的问题。

3.3 PART III 变长编码（两符号联合）

² 同样采用均匀量化，步长20，得到的结果如下：

²本部分完成人：李思涵



可以看出采用两符号联合方式编码的压缩比是小于独立符号编码的。