

信源编码

无 37

刘家硕

2013011212

Part I 分工

量化

1. 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。
2. 练习 JPEG/H.261 量化器，绘制 R-D 图。
3. 设计非均匀量化器，绘制 R-D 图。
4. 对三类量化器进行评价。

变长编码（独立符号）

1. 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。
2. 给出编码前后的比特数，计算压缩比。

变长编码（两符号联合）

1. 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。
 2. 给出编码前后的比特数，计算压缩比。
-

Part II 模块实现

我负责的是量化部分，主要通过提供的程序测量

- 均匀量化
- H261 量化
- 非均匀量化

的 bitrate(processed bits) 和 PSNR

然后通过 Performance.xls 中的宏计算 bdrate 比较量化性能以及作图

我设计了两种非均匀量化方法，这里给出较优（几乎没有差别）的一种量化序列设计程序

```
import sys

step = int(sys.argv[1])
array = []

idx = 0
while idx <= 255:
    array.append(idx)
    if idx < 123:
        step += 1
        idx += step
    else:
        step -= 1
        idx += step

for i in array:
    print(i,end=' ')
```

可根据输入步长设计由密变疏再变密的中部稀疏量化

另一种量化方式是中部密集量化，效果不如均匀量化（差距很小）

Part III 实验结果

	step factor	均匀量化		other				BD-rate(%)
		Bitrate(kbps)	psnr	compress rate	Bitrate(kbps)	psnr	compress 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	

Figure 1: bdrate

可见 H261 远优于均匀量化，另两种非均匀量化性能和均匀差不多。其实应该存在最优量化方式，使得重建数值在区间的重心而区间边界为相邻重建数值的中值，但由于所提供程序中重建数值为区间中值，故无法实现这种最优。

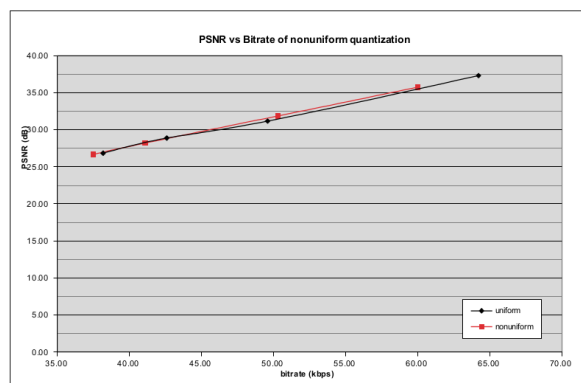
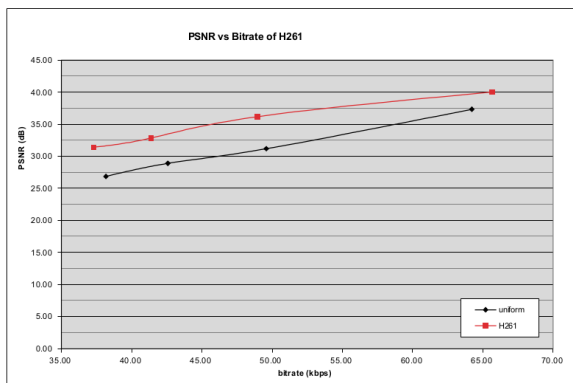


Figure 2: plot