# 数字图像处理实验报告六

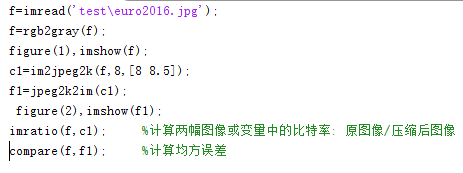
学号：17020031016 专业：计算机科学与技术 姓名：胡帅

1.实验描述

对test 目录下的图像进行图像压缩测试，调节参数查看效果。

2.实验过程

###### 图片压缩程序：



Compressiontest.m

###### 原图：



###### 输出结果：

**;Picture：euro2016**

当**c1=im2jpeg2k(f,8,[8 8.5]);**压缩比 = 114.1585均方误差 = 18.419

当**c1=im2jpeg2k(f,**6**,[8 8.5]);** 压缩比 = 25.5724均方误差 = 8.0604

当**c1=im2jpeg2k(f,**4**,[8 8.5])** 压缩比 = 9.6588均方误差 = 3.2080

灰度图



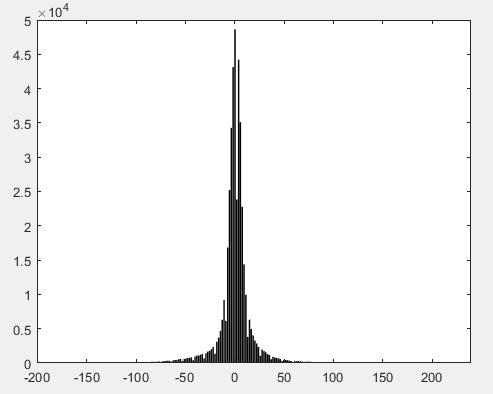
压缩后的图像

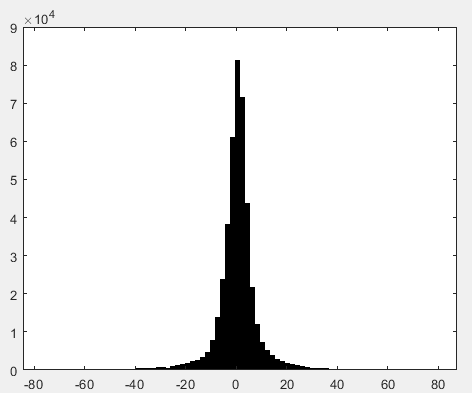


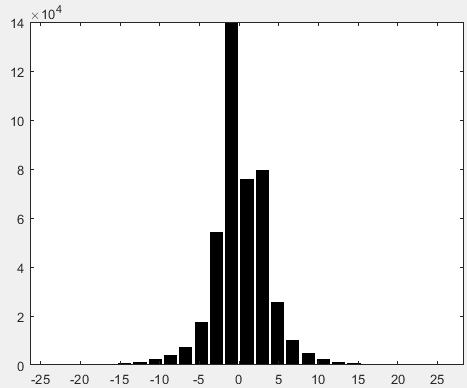




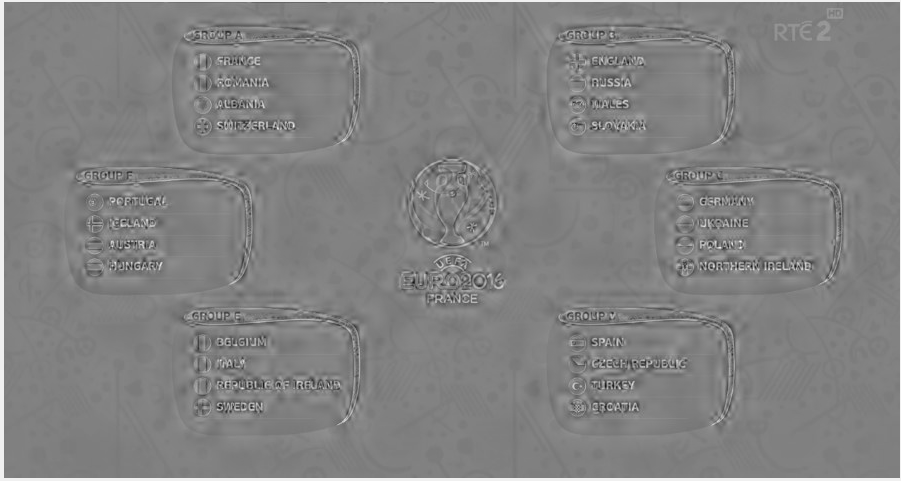
误差直方图

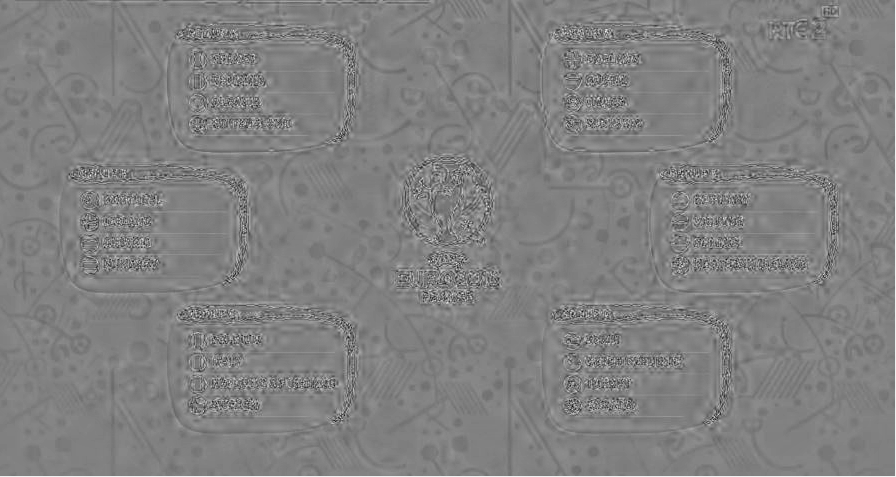


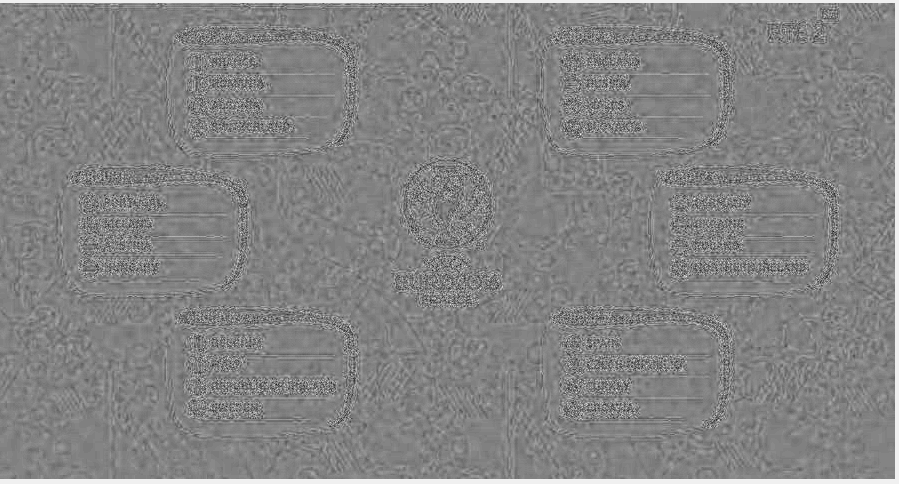




误差图







**Picture：euro**

c1=im2jpeg2k(f,8,[8 8.5]); **压缩比：35.9637；均方误差：30.9474**

c1=im2jpeg2k(f,6,[6 8.5]); **压缩比：8.2114；均方误差：12.7164**

c1=im2jpeg2k(f,6,[4 8.5]); **压缩比：3.5348；均方误差：4.1788**

**灰度图**

****

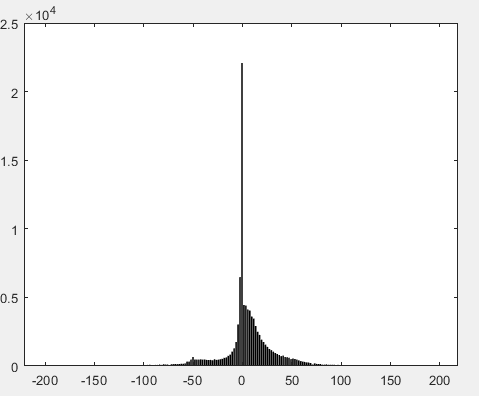
**压缩后的图像**

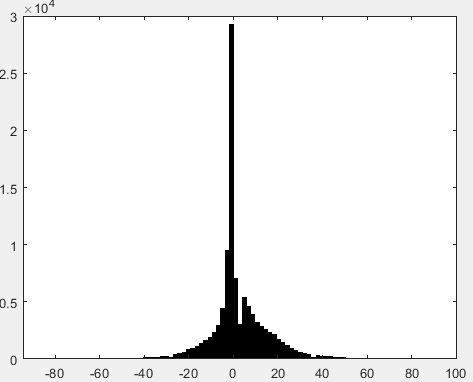
****

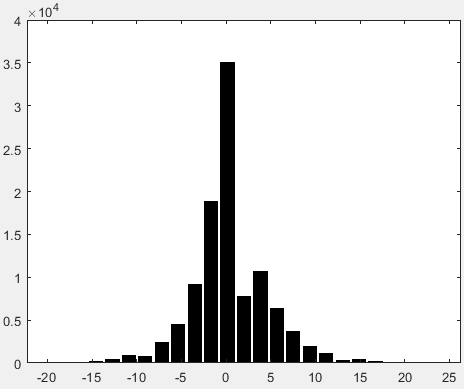
****

****

**误差直方图**

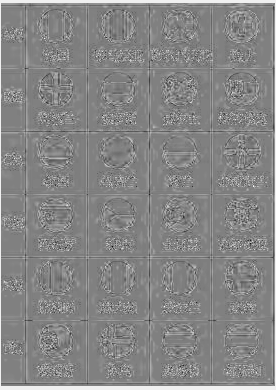
****

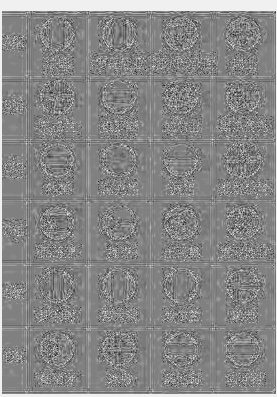
****

****

**误差图**







**Picture：wor1dcup 压缩比：45.6148；均方误差：33.3686**

**Picture：wor1dcup 压缩比：8.9692； 均方误差：11.8029**

**Picture：wor1dcup 压缩比：4.2244； 均方误差：3.8062**

**灰度图**

****

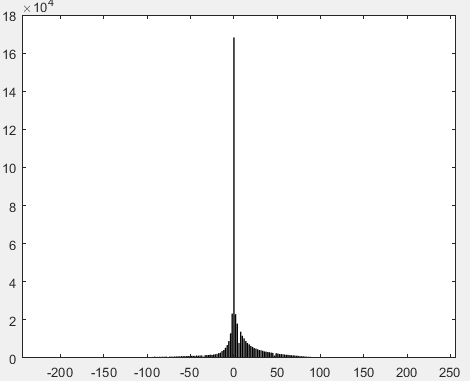
**压缩后的图像**

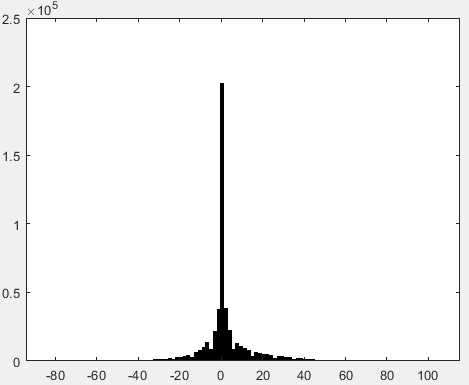
****

****

****

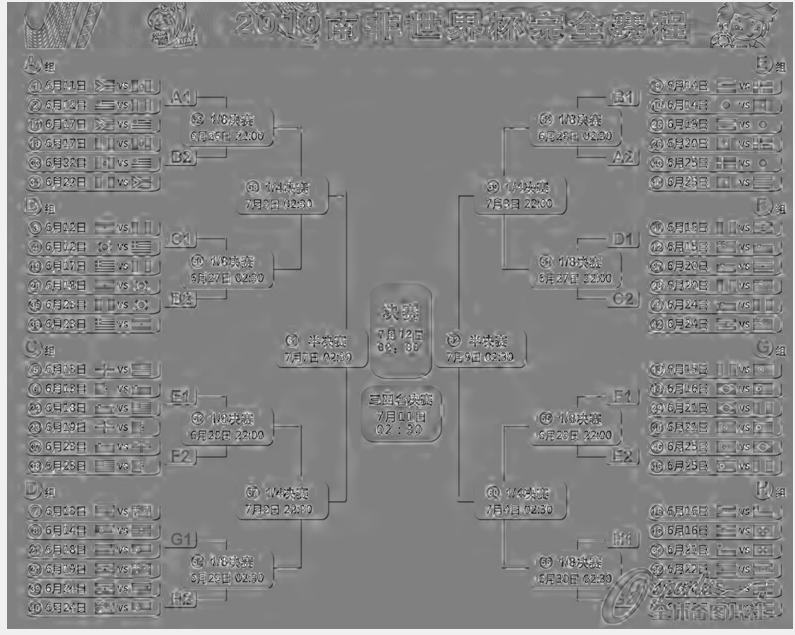
误差直方图

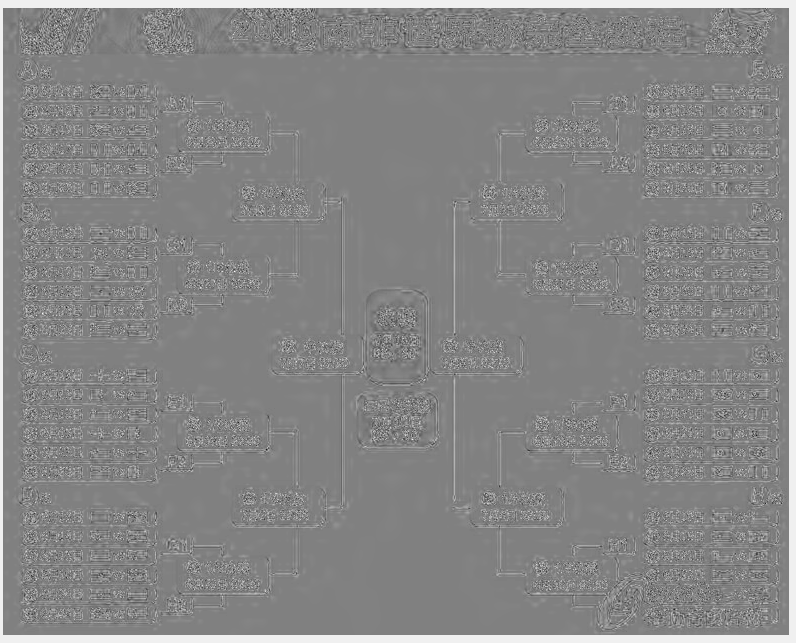


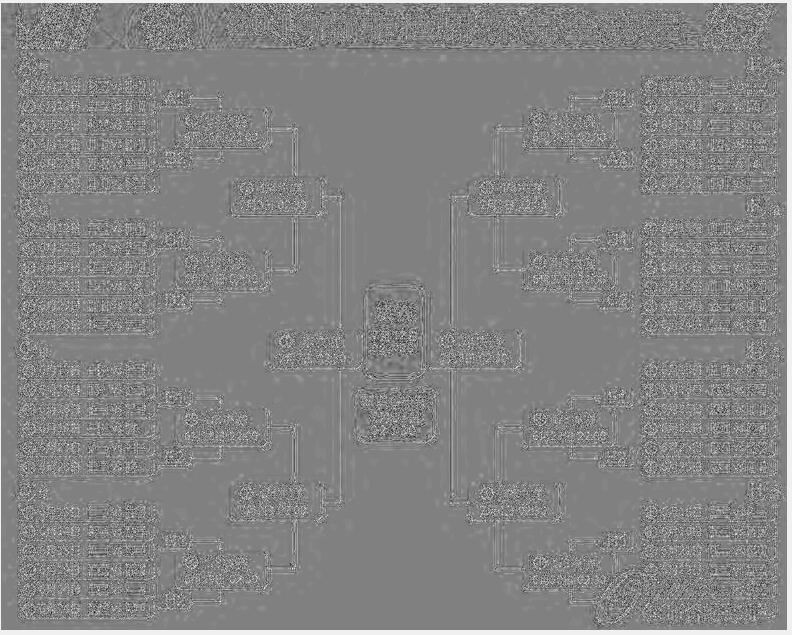




误差图







3.关键代码理解

##### im2jpeg2k.m

1. %im2jpeg2k源程序
2. function y=im2jpeg2k(x,n,q)
3. %im2jpeg2k使用JPEG2000近似值压缩图像
4. %y=im2jpeg2k(x,n,q)使用N尺度jpeg2k小波变换，不明确和明确系数量化和零步长解码去压缩图像
5. %如果量化向量仅包含两个元素，它们被认为不明确量化参数。
6. %否则被认为包含明确子带步长。
7. %y是包含霍夫曼编码数据和加上为编码通过JPEG2K2IM得到的参数的一个编码结构。
9. global RUNS
10. error(nargchk(3,3,nargin));%检查输入参数
12. **if** nidms(x) ~=2 | ~isreal(x) | ~isnumeric(x) | ~isa(x,'uint8')
13. error('The input must be a UINT8 image.');
14. end
16. **if** length(q) ~=2 & length(q) ~=3\*n+1
17. error('The quantization step size vector is bad.');
18. end
19. %水平平移输入和计算它的小波变量
20. x=**double**(x)-128;
21. [c,s]=wavefast(x,n,'jpeg9.7');
23. %量化小波系数
24. q=stepsize(n,q);
25. sgn=sign(c);sgn(find(sgn==0))=1;c=abs(c);
26. **for** k=1:n
27. qi=3\*k-2;
28. c=wavepaste('h',c,s,k,wavecopy('h',c,s,k)/q(qi));
29. c=wavepaste('v',c,s,k,wavecopy('v',c,s,k)/q(qi+1));
30. c=wavepaste('d',c,s,k,wavecopy('d',c,s,k)/q(qi+2));
31. end
32. c=wavepaste('a',c,s,k,wavecopy('a',c,s,k)/q(qi+3));
33. c=floor(c);c=c.\*sgn;
35. %通过创造一个特别的编码为0执行和结束编码来开始和做一个步长表
36. zrc=min(c(:))-1;eoc=zrc-1;RUNS=[65535];
38. %找到步长变换指针：'plus'包含零运行开始的指针与相适应的minus是它的结束+1
39. z=c==0;z=z-[0 z(1:end-1)];
40. plus=find(z==1);minus=find(z==-1);
42. %从c中删除零运行
43. **if** length(plus) ~=length(minus)
44. c(plus(end):end)=[];c=[c eoc];
45. end
47. %删除从c中所有其他零运行(建立在'plus'和'minus')
48. **for** i=length(minus):-1:1
49. run=minus(i)-plus(i);
50. **if** run>10
51. ovrflo=floor(run/65535);run=run-ovrflo\*65535;
52. c=[c(1:plus(i)-1) repmat([zrc 1],1,ovrflo) zrc ...
53. runcode(run) c(minus(i):end)];
54. end
55. end
57. %霍夫曼编码和加混合。解码信息
58. y.runs=uint16(RUNS);
59. y.s=uint16(s(:));
60. y.zrc=uint16(-zrc);
61. y.q=uint16(100\*q');
62. y.n=uint16(n);
63. y.huffman=mat2huff(c);

##### im2jpeg2k.m

1. %jpeg2k2im源程序
2. function x=jpeg2k2im(y)
3. %解码一个im2jpeg压缩图像
4. %x=jpeg2k2im(y)解码压缩图像y，重构初始图像x的一个近似值。
5. %y是通过im2jpeg2k返回编码结构
7. %也看im2jpeg2k
9. error(nargchk(1,1,nargin));%检查输入参数
11. %得到解码参数：尺度、量化向量、步长表大小、零运行编码、数据结束编码、小波簿记数组和步长表
12. n=**double**(y.n);
13. q=**double**(y.q)/100;
14. runs=**double**(y.runs);
15. rlen=length(runs);
16. zrc=-**double**(y.zrc);
17. eoc=zrc-1;
18. s=**double**(y.s);
19. s=reshape(s,n+2,2);
21. %计算小波变换大小
22. c1=prod(s(1,:));
23. **for** i=2:n+1
24. c1=c1+3\*prod(s(i,:));
25. end
27. %通过零运行编码执行霍夫曼编码
28. r=huff2mat(y.huffman);
30. c=[];zi=find(r==zrc);i=1;
31. **for** j=1:length(zi)
32. c=[c r(i:zi(j)-1) zeros(1,runs(r(zi(j)+1)))];
33. i=zi(j)+2;
34. end
36. zi=find(r==eoc);%撤消终止零运行或最后非零运行
37. **if** length(zi)==1
38. c=[c r(i:zi-1)];
39. c=[c zeros(1,c1-length(c))];
40. **else**
41. c=[c r(i:end)];
42. end
44. %非标准化系数
45. c=c+(c>0)-(c<0);
46. **for** k=1:n
47. qi=3\*k-2;
48. c=wavepaste('h',c,s,k,wavecopy('h',c,s,k)\*q(qi));
49. c=wavepaste('v',c,s,k,wavecopy('v',c,s,k)\*q(qi+1));
50. c=wavepaste('d',c,s,k,wavecopy('d',c,s,k)\*q(qi+2));
51. end
52. c=wavepaste('a',c,s,k,wavecopy('a',c,s,k)\*q(qi+3));
53. %计算逆小波变换和水平平移
54. x=waveback(c,s,'jpeg9.7',n);
55. x=uint8(x+128);

参考：<https://blog.csdn.net/yangyayan/article/details/90018544>

4.实验结论

尺度参数数值设置的越大则图像的压缩率越高，压缩后的图像和原图相比均方误差越大，看上去越模糊。