**Project4**

**Constrained Least Squares Filtering**

张元鑫 2018210902

**实验内容**

1. Image blurring by Motion Model:

Input：ImageOrg & H, a=0.1, b=0.1 and T=1

Output：ImageBlur

1. Constrained Least Squares Filtering

Input：ImageBlur & H, a=0.1, b=0.1 and T=1, you may estimate

Output：ImageOrg

**实验原理**

1. Image blurring by Motion Model

图像退化函数：



生成的退化滤波器如下图所示：



关键代码如下图所示：

1. %退化函数
2. a = 0.1;
3. b = 0.1;
4. T = 1;
5. H = zeros([m,n]);
6. **for** u =1:m
7. **for** v = 1:n
8. u\_t = u-0.5\*m-1;
9. v\_t = v-0.5\*n-1;
10. temp =(u\_t\*a+v\_t\*b)\*pi;
11. **if** temp == 0
12. H(u,v) = 1;
13. **else**
14. H(u,v)=T/temp\*sin(temp)\*exp(-1i\*temp);
15. end
16. end
17. end
18. Constrained Least Squares Filtering

自动求解最优的主要步骤：

指定的一个初始值

计算

若满足，则停止；否则，若，增大，若，减小，然后返回步骤2.

关键代码如下：

1. p =zeros([m,n]);
2. p(1:3,1:3) = [0,-1,0;-1,4,-1;0,-1,0];
3. p = myfftshift(p);
4. P = myDFT2(p);
6. eta\_square = sum(m\*n\*(MU.^2+var));
7. Fc = zeros([m,n,d]);
8. gamma =0.001;
9. eps = 1e6;
10. r\_square = 0;
11. H\_c = conj(H);
12. **while**(r\_square<eta\_square-eps || r\_square>eta\_square+eps)
13. Fc = (H\_c./(H\_c.\*H + gamma\*abs(P).^2)).\*G;
14. R = G-H.\*Fc;
15. r = real(myIDFT2(R));
16. r = myfftshift(r);
17. r\_square = sum(sum(r.^2));
18. **if** gamma == 0
19. **break**;
20. elseif r\_square<eta\_square-eps
21. gamma = gamma\*1.05;
22. **else**
23. gamma = gamma\*0.95;
24. end
25. display(gamma);
26. display(eta\_square-eps);
27. display(r\_square);
28. end
29. 图像噪声估计

根据教材5.2.4节算法，图像噪声方差可以由合理的恒定值灰度值的一小部分来估计。通过计算灰度值归一化直方图，可以得到条带的均值和方差：

关键代码如下

1. Band = g(125:130,210:250);
2. [bm,bn] = size(Band);
3. figure;
4. imshow(uint8(Band));
5. title('小条带');
6. ps = zeros(1,256);  %概率直方图
7. **for** i =1:bm
8. **for** j = 1:bn
9. ps(Band(i,j)+1)=ps(Band(i,j)+1)+1;%统计每个灰度值出现的频率
10. end
11. end
12. ps = ps/(bm\*bn);
13. mu = 0;
14. figure;
15. plot(0:255,ps);
16. title('概率直方图');
17. **for** i =0:255
18. mu = mu+i\*ps(i+1); %计算均值
19. end
20. var = 0;
21. **for** i =0:255
22. var = var+(i-mu)^2\*ps(i+1); %计算噪声方差
23. end
24. MU = mean(mean(Band-mean(Band(:)))); %计算噪声均值

选取的小条带如下如所示：

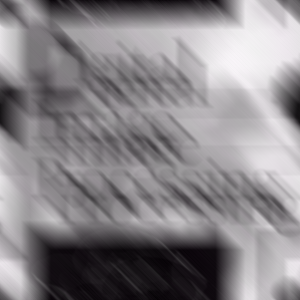
统计的概率图如下所示



**实验结果**

Task(1):

退化后的图像如下所示：



完整代码见task1.m

Task(2):

使用估计的噪声方差均值迭代得到的值为：0.0095，此时的恢复效果如下图所示：



另外手动指定一些的值，观察恢复效果如下

大，恢复出来的图像比较模糊，太小，恢复出来的图像颗粒感就比较重，应当适中。

完整代码见task2.m