

SVD之去噪

一、实验题目

生成一张带噪声的手写体图片，例如写几个数字，数字主体部分清晰，但是其余部位有噪声点(例如比较小的墨水痕迹等)，结合课堂讲述的SVD的内容，将噪声去掉，产生新的手写体图片。用可行的方式比较前后两张图片。

二、实验原理

课堂所学SVD知识

三、实验代码

MATLAB代码码如下

```
A=imread("noise2.jpg");%导入图像

figure;

imshow(A);

title('原始图片');

A=rgb2gray(A);%转变为灰度图

A=double(A);

[u s v]=svd(A);

k=8;%保留前k个奇异值（8仅为测试值）

left=u(:,1:k);

right=v(:,1:k);

val=s(1:k,1:k);

out=uint8(left*val*right');%重构图像

figure;

imshow(out);
```

```
title('去噪后图片');
```

四、运行结果及解释

1.对A进行SVD后，查看S矩阵，如下图所示

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	9.6399e+04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	7.8623e+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	6.9417e+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	5.6475e+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	4.4462e+03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	4.1037e-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	3.2967e-03	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	2.2885e+03	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	2.0473e+03	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1.5536e+03	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1.3121e+03	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1.1240e+03	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1.0994e+03	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0384e+03	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	989.3411	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	907.6152	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	866.4864	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	837.5120	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	767.9871	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	755.9738	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	734.

	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226
205	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
206	566	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
207	0	0.2324	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
208	0	0	0.1873	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
209	0	0	0	0.1359	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
210	0	0	0	0	0.1259	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
211	0	0	0	0	0	0.0547	0	0	0	0	0	0	0	0	0	0	0	0	0	0
212	0	0	0	0	0	0	7.6514e-11	0	0	0	0	0	0	0	0	0	0	0	0	0
213	0	0	0	0	0	0	0	1.1044e-11	0	0	0	0	0	0	0	0	0	0	0	0
214	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0	0	0	0	0	0
215	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0	0	0	0	0
216	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0	0	0	0
217	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0	0	0
218	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0	0
219	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0	0
220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0	0
221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0	0
222	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0	0
223	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0	0
224	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12	0
225	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9.6041e-12
226	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
227	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
228	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
229	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

可见，值在不同的几点发生了较大变化

行数	1	2-14	15-190	190-211	211
值	10 ⁴	10 ³	10 ² -10 ⁰	10 ⁻¹	10 ¹⁹

2.故选取的第一个K值为14，运行结果如下

原始图片



去噪后图片



可见去噪效果一般，由此继续向下尝试不同值的K

k=10

原始图片



去噪后图片



k=8

原始图片



去噪后图片



k=6

原始图片



去噪后图片



... ..

经过多次实验后，降噪的效果始终不够理想，在降噪的同时丢失了部分数字主体的细节，总之，对于图片降噪或许需要其他更好的方法