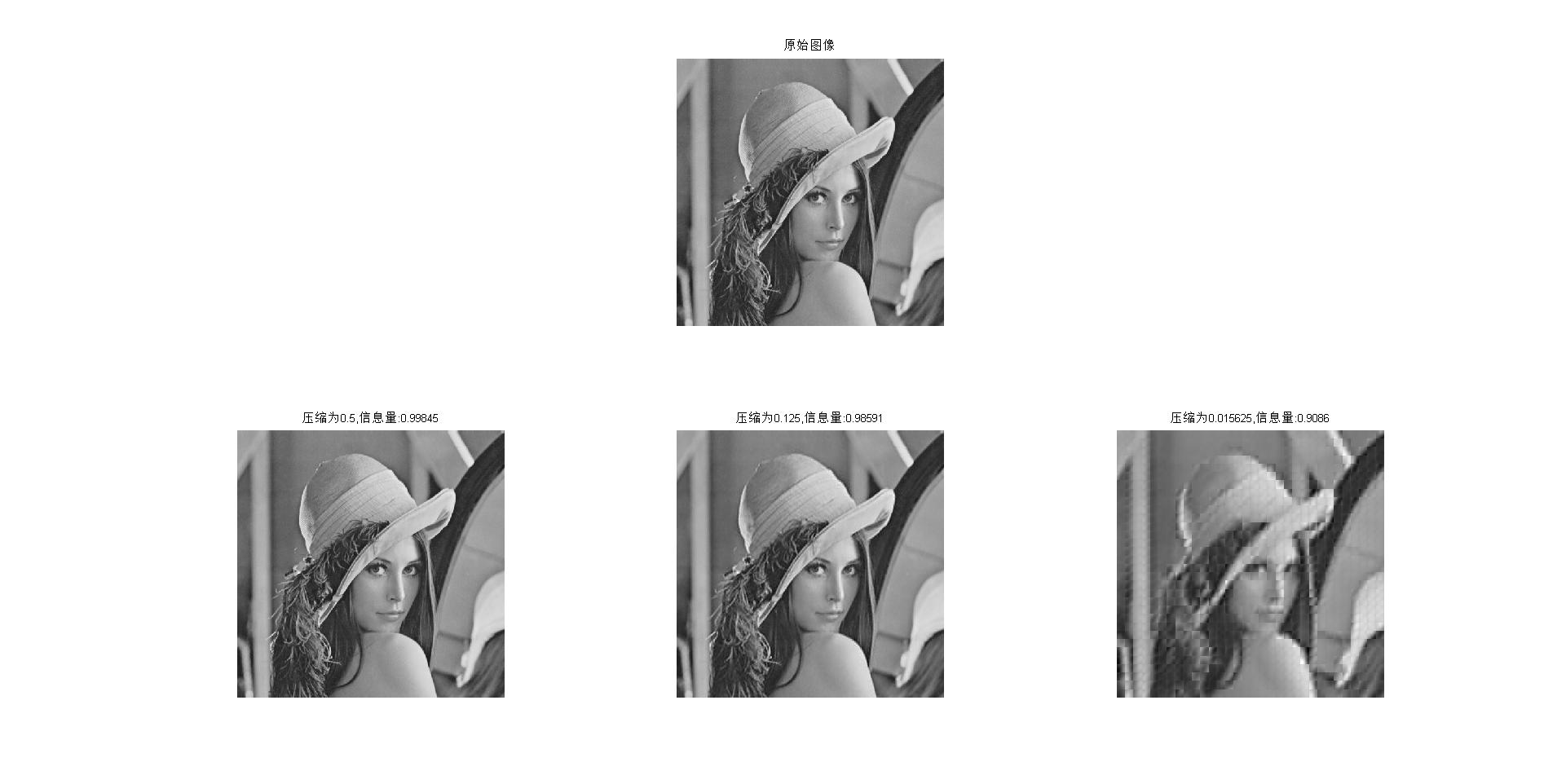
Project3实验报告

16337040戴文俊

完成日期：2018/11/25

1. 实验要求

使用PCA（主成分分析）对输入的一张灰度图片lena，将其放大到256\*256，使用PCA方法把原始图片分别按照2:1、8:1、32:1进行压缩，即压缩后的数据量为原始图片的1/2、1/8、1/32。分析压缩后的数据所含信息量大小，并比较压缩数据再经过重建后与原始图片的视觉差异。示例：



1. 知识准备

在研究某一事物的过程中，为了更全面、准确地反映事物的特征及其发展规律，往往需要考虑与其相关的多个指标（变量），但是随着指标数量的增加，分析以及计算问题的复杂度会变得很高，并且有些特征信息会有重合，因此这时候我们考虑在保留原始变量主要信息的前提下，对目标问题的特征进行降维以及简化，因此这里引入了主成分。主成分具有以下信息：

* 每一个主成分都是各原始变量的线性组合；
* 主成分数目远小于原始变量数目；
* 主成分保留了原始变量绝大多数信息；
* 各主成分之间互不相关。

PCA的目标就是要找到这样的主元，最大程度的去除冗余和噪声的干扰。具体的推导过程就不详细介绍了，这里主要简述一下PCA的算法流程，输入时n维样本集为要降到的维数，输出为降维后的样本。

1. 对所有的样本进行中心化，
2. 计算样本的协方差矩阵
3. 对矩阵进行特征值分解
4. 将特征向量按照对应特征值从上到下按行排列成矩阵，取前k行组成矩阵P
5. Y=PX即为降维到K维后的数据

除此之外，在实际研究中，为了描述所选取的特征的贡献率（或者说所包含的信息量），我们通常选取p个（p<=m）个主成分，计算主成分贡献率以及累计贡献率作为，称

为主成分的信息贡献率，同时，有

为主成分,的累计贡献率，从而可以对P个主成分进行综合分析。

3、PCA压缩图片实现

1）把图分成N个不重叠的16\*16的小块，也就是变成了N\*256的数据集

2）根据对应的压缩率对图片进行PCA压缩，例如压缩率为1/2，就是用PCA降维度先变成N\*128

3）将压缩之后的数据进行存储或者传输之后，解压的过程就相当于重建，例如压缩率为1/2，也就是N\*128重新变成N\*256（N\*16\*16），然后又将小块重新组装回一张图，即可得到重建后的图像

本次实验使用MATLAB实现，相关代码见末尾文档附录。

4、实验结果及分析

将压缩率分别设置为1/2，1/8，1/32，运行MATLAB程序可以得到以下效果：



由上面的结果我们可以看到，当压缩率比较的时候，可能压缩之后再复原的结果跟原图的差别似乎不是特征大， 但是当压缩率比较小的时候，也就是压缩的比较严重的时候，可以明显地看到图片有了比较大的模糊，这也进一步说明了PCA的有损压缩的特性。通过代码计算得到，

|  |  |
| --- | --- |
|  |  |
| 压缩率 | 累计贡献率 |
| 1/2 | 0.9934 |
| 1/8 | 0.9424 |
| 1/32 | 0.8050 |

就信息量（特征的累计贡献率）对比而言， 随着压缩率的减小，也就是压缩地越多，所提取的主成分（特征）的贡献越小，也就是说信息丢失地更为严重，用更少的信息还原成原图像的难度更大。

5、实验总结

这次的主成分分析（PCA）实验总的来说难度并不大，目标也算明确，但是要达到非常透彻地理解算法，非常熟练的实现这个算法，还是得下一番功夫。要实现这个PCA，首先要求自身线性代数的矩阵知识方面要掌握地比较好，其次还要理解PCA的算法过程，明确哪些成分是主要的，哪些成分是次要的甚至是冗余或者噪声。利用PCA对图片进行压缩也让我们进一步了解了图片特征的提取和后续的处理过程，尝试不同形式的特征提取对于能够进一步加深我们对此的理解。

6、附录

img=imread('lena.bmp');

[M, N] = size(img);

f = double(img);

bs = 16; %block size

p = 128; %remaining dimension

% PCA

x = im2col(f, [bs bs], 'distinct'); %xet 1024 blocks

x\_m = ones(size(x,1),1)\*mean(x); %calculate the mean of each block

x = x - x\_m; %x\_i - x\_mean

cov = x\*x'/(size(x,2)-1);

[E, D] = eig(cov);

size(E)

[~,order] = sort(diag(D),'descend');

E = E(:,order);

d = diag(D);

D = diag(d(order));

%calculate the contribution rate

info = sum(d(256-p:256))/sum(d)

E\_proj = E(:,1:p); %find p\_max eigenvectors

x\_proj = x'\*E\_proj; %map dim(bs\*bs) to p dimension

% recover the image

x\_rec = x\_proj\*E\_proj';

s = x\_rec' + x\_m;

s = col2im(s, [bs bs], [M N], 'distinct');

subplot(121),imshow(img,[]);title('Original Image');

subplot(122),imshow(s,[]);title(['r = ',num2str(p/256)]);