数字图像处理实验报告

**实验项目名称: 实验5 图像变换（2）**

**姓名: myp 学号 20201202075 班级 20计科1**

**提交时间: 2023.3.31**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**（一）实验目的**

1、了解图像的K-L变换的基本原理；

2、掌握K-L变换的Matlab编程。

**（二）实验内容和实验原理**

1、编程实现图像的K-L变换；

**（三）实验步骤**

1、K-L变换的基本原理。

K-L变换简单来讲就是找到一个矩阵，让原始矩阵和该矩阵的相乘得到一个新矩阵，降低矩阵的维度，从而实现降维，去噪的效果，类似于PCA的效果（网上有人认为K-L变换实际上就是PCA主成分分析法，当变换矩阵为协方差矩阵，两者等价）。问题在于如何找到这个矩阵，所以K-L变换基于统计特征原理，将恢复图像和原图像的均方差最小为目标，从而得到这个矩阵。

2、利用Matlab实现K-l变换

代码：

KL\_compress("cameraman.tif");

function KL\_compress(imgpath);

% 使用K-L变换/主成分分析来对图像压缩

f=imread(imgpath);

f=im2double(f);

y=[];

[m,n]=size(f);

for i=1:m/8

for j=1:n/8

ii=(i-1)\*8+1;

jj=(j-1)\*8+1;

y\_append=reshape(f(ii:ii+7,jj:jj+7),1,64);

y=[y;y\_append];

end

end

% K-L变换

[coeff,score,latent]=pca(y);

k1=y\*coeff;

k11=k1;

k12=k1;

k13=k1;

k14=k1;

% 压缩过程，不需要的地方置为0

k11(:,17:64)=0;

k12(:,9:64)=0;

k13(:,5:64)=0;

k14(:,3:64)=0;

% K-L逆变换

k11\_inverse=k11\*coeff';

k12\_inverse=k12\*coeff';

k13\_inverse=k13\*coeff';

k14\_inverse=k14\*coeff';

k=1;

% 重组图像

for i=1:m/8

for j=1:n/8

y1=reshape(k11\_inverse(k,1:64),8,8);

y2=reshape(k12\_inverse(k,1:64),8,8);

y3=reshape(k13\_inverse(k,1:64),8,8);

y4=reshape(k14\_inverse(k,1:64),8,8);

ii=(i-1)\*8+1;

jj=(j-1)\*8+1;

image1(ii:ii+7,jj:jj+7)=y1;

image2(ii:ii+7,jj:jj+7)=y2;

image3(ii:ii+7,jj:jj+7)=y3;

image4(ii:ii+7,jj:jj+7)=y4;

k=k+1;

end

end

imshow(image1),title('64 压缩至25%');

figure,imshow(image2),title('32 压缩至12.5%');

figure,imshow(image3),title('16 压缩至6.25%');

figure,imshow(image4),title('8 压缩至3.125%');

end

结果：

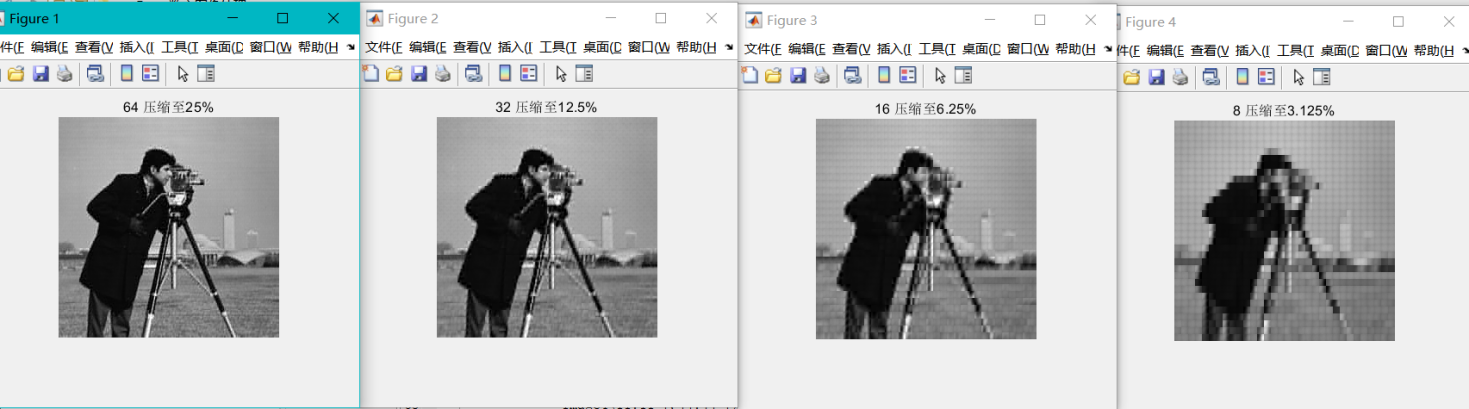


图 1 K-L变换后结果

**四、实验心得体会**

## 实验过程中，使用pincomp函数时显示找不到函数，通过查询相关资料发现这个函数已经淘汰了，实际上现在使用的多是pca函数。同时在查找相关内容的同时，对于kl变换的作用和与pca之间的差别有了更深的了解。Kl变换实际上像是pca的二维形式，pca主要用于一维的降维，而kl变换则是二维图形的压缩降维。