媒体与认知第一次作业

李溢 2016011235

2019/4/7

---

1. 小波变换的矩阵形式

(a) $H_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$ $H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}1/\sqrt{2} & 1/\sqrt{2} & 0 & 0\\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1\\ 1 & -1 & 0 & 0\\ 0 & 0 & 1 & -1\end{bmatrix}$

$$H_3 = \begin{bmatrix}H_2 & 0\\ 0 & I_2\end{bmatrix}\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix} =$$

```
0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536    0.3536
0.3536    0.3536    0.3536    0.3536   -0.3536   -0.3536   -0.3536   -0.3536
0.5000    0.5000   -0.5000   -0.5000         0         0         0         0
     0         0         0         0    0.5000    0.5000   -0.5000   -0.5000
0.7071   -0.7071         0         0         0         0         0         0
     0         0    0.7071   -0.7071         0         0         0         0
     0         0         0         0    0.7071   -0.7071         0         0
     0         0         0         0         0         0    0.7071   -0.7071
```

$H_3^{-1} =$

```
0.3536    0.3536    0.5000         0    0.7071         0         0         0
0.3536    0.3536    0.5000         0   -0.7071         0         0         0
0.3536    0.3536   -0.5000         0         0    0.7071         0         0
0.3536    0.3536   -0.5000         0         0   -0.7071         0         0
0.3536   -0.3536         0    0.5000         0         0    0.7071         0
0.3536   -0.3536         0    0.5000         0         0   -0.7071         0
0.3536   -0.3536         0   -0.5000         0         0         0    0.7071
0.3536   -0.3536         0   -0.5000         0         0         0   -0.7071
```

(b)

$v = [7, 11, 9, 6, 4, 3, 1, 3]^T$

$w = H_3 * v =$[15.5563 7.7782 1.5000 1.5000 -2.8284 2.1213 0.7071 -1.4142]^T

2. PCA与LDA

(a) PCA

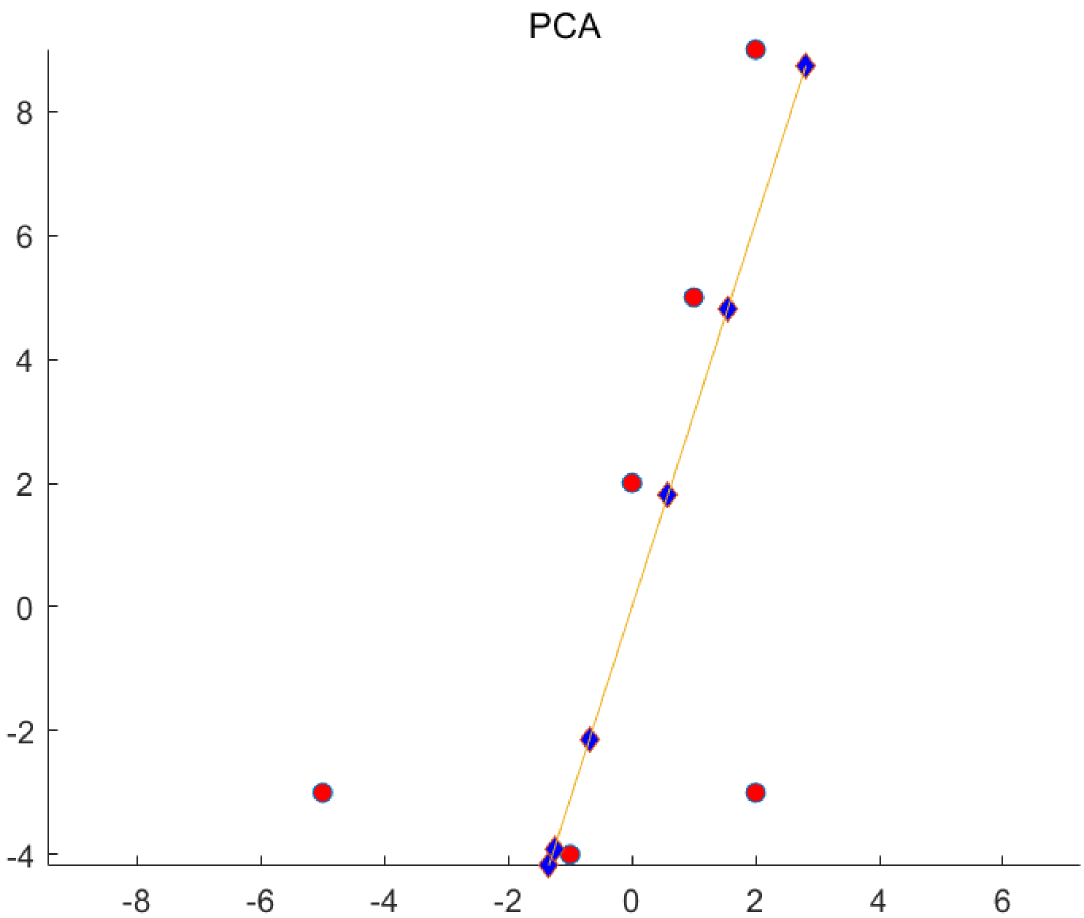
均值 $\mu = [-0.1667, 1.0000]^T$

协方差 $conv = \begin{bmatrix} 5.8056 & 6.1667 \\ 6.1667 & 23.000 \end{bmatrix}$

特征值分解：$\lambda_1 = 24.9829, \lambda_2 = 3.8226$

最大特征值对应的特征向量，也即投影方向 $w = [0.3061, 0.9520]^T$

投影后如图所示



(b) LDA

不同类别各自的均值为 $\mu_1 = [-0.6667, 3.6667]^T, \mu_2 = [0.3333, -1.6667]^T$

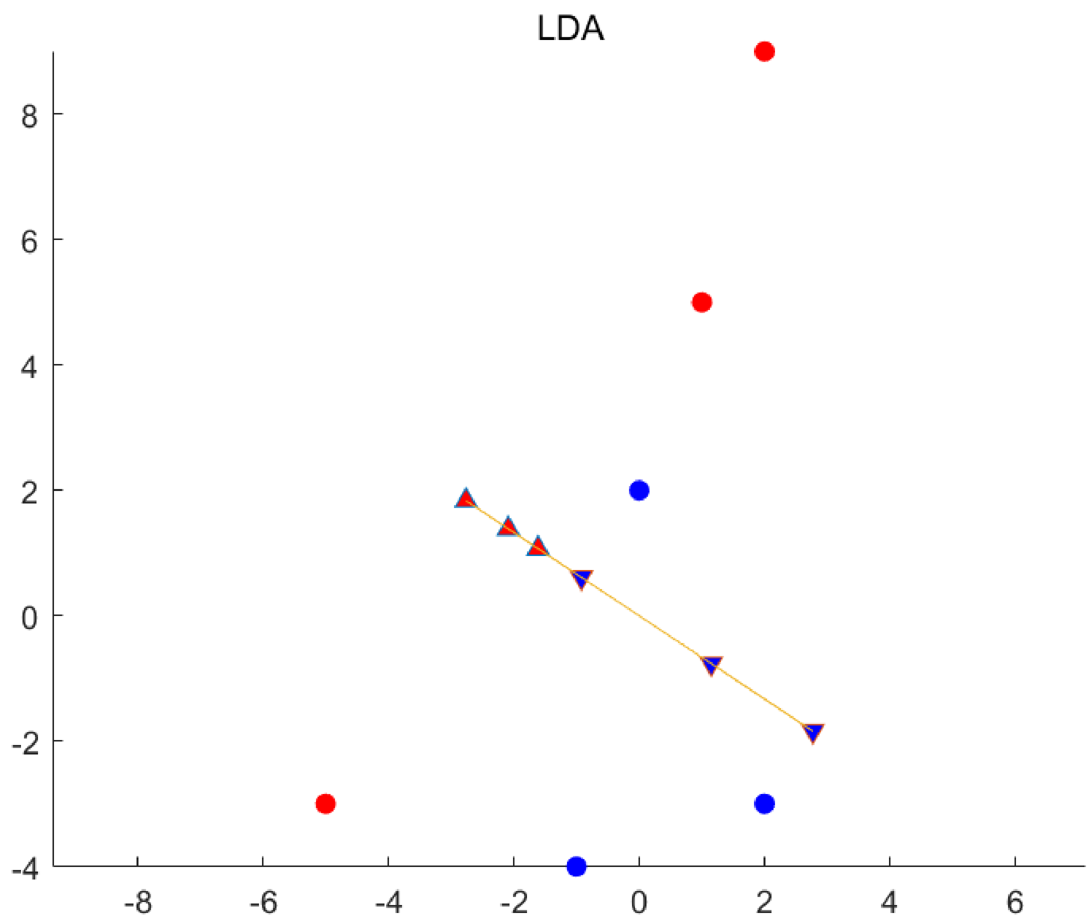
类内散度 $S_w = S_{w_1} + S_{w_2} = \begin{bmatrix} 33.3333 & 45.0000 \\ 45.0000 & 95.3333 \end{bmatrix}$

类间散度 $S_b = (\mu_1 - \mu_2) * (\mu_1 - \mu_2)^T = \begin{bmatrix} 1.0000 & -5.3333 \\ -5.3333 & 28.4444 \end{bmatrix}$

特征值分解 $\lambda = 1.3216$

对应的特征向量，也即投影方向 $w = [0.8329, -0.5534]^T$

投影后如图所示



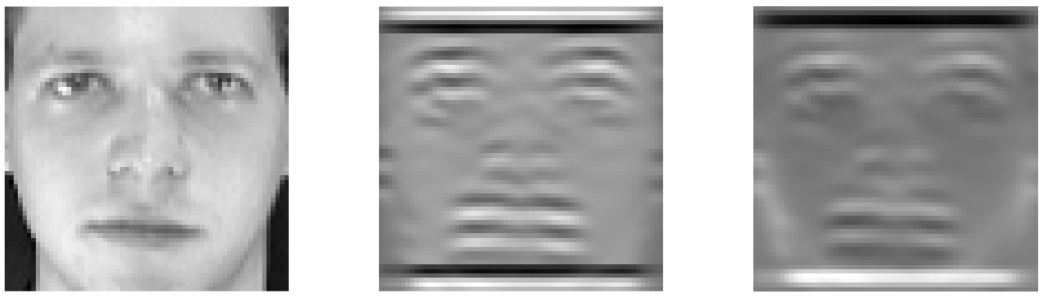
3. SIFT特征变换

如图所示

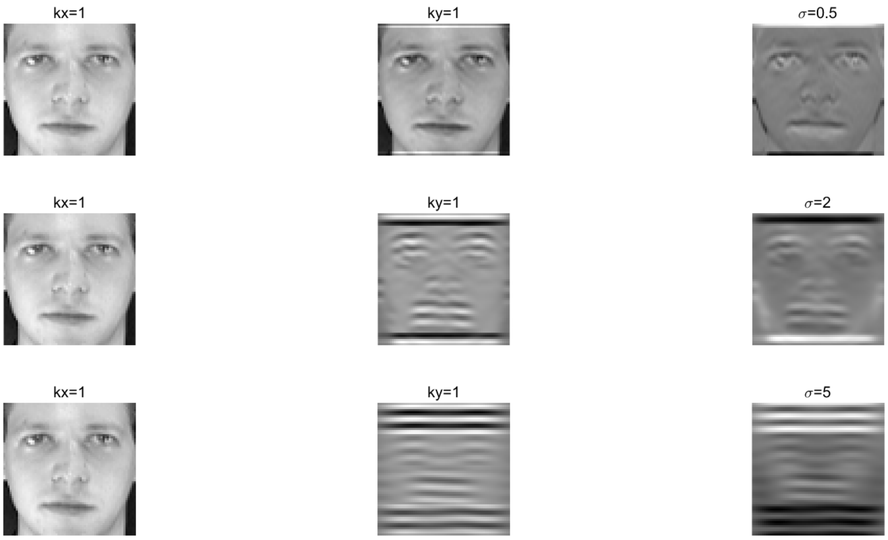
4. Gabor滤波器组

（a）设计参数为 $k_x = 1, k_y = 1, \sigma = 2$ 时，效果如图所示

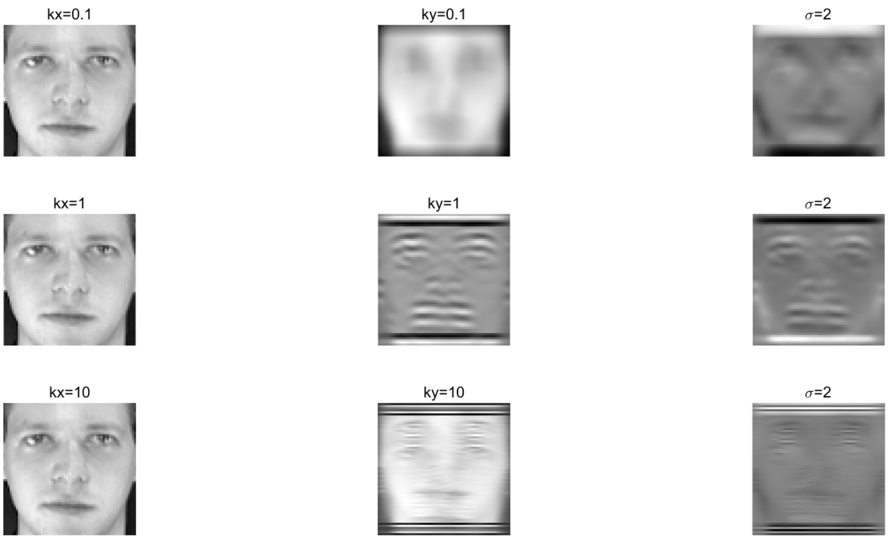
kx=1 ky=1 σ=2



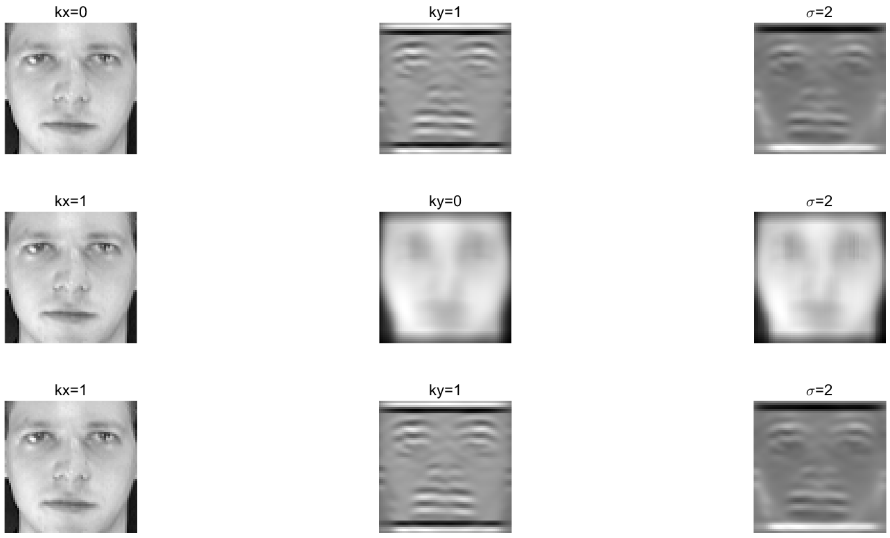
如下图所示，改变 σ 参数时，可以发现 σ 越大，生成的特征越模糊。因为Gabor滤波器中 σ 影响了高斯函数的衰减速度，σ 越大，衰减越慢，周围的图像对中心的影响越大，因此生成的特征也越模糊；



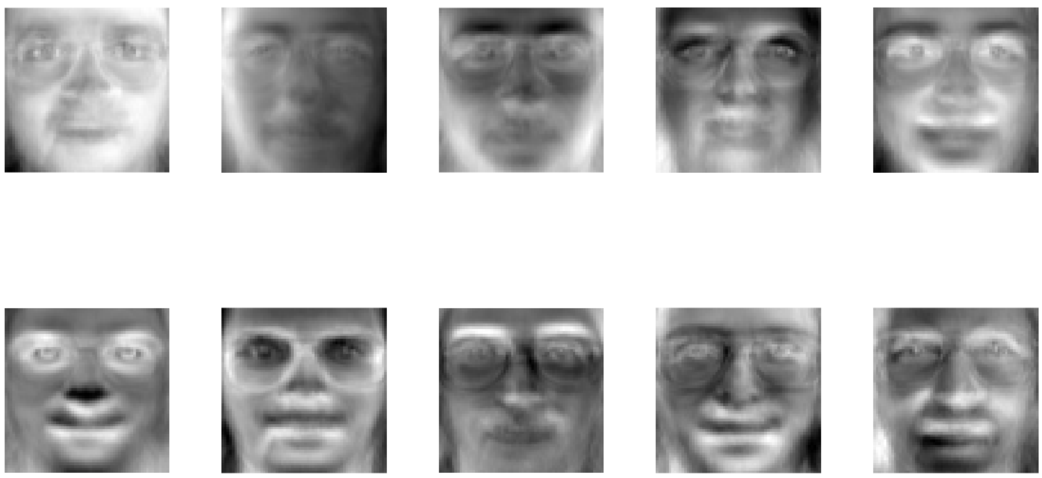
如下图示，同时改变 k_x, k_y 的值时，可以发现两者值越小，生成的特征越模糊，可以理解为低频分量在图片的各个地方分布较为平坦，因此不同地方得到的特征值差不多，所以特征图也更模糊；



如下图所示，在 k_x, k_y 取三种不同相对值的情况下得到的特征也明显不同。可以看出当 $k_x=1, k_y=0$ 时生成的特征非常模糊，原因是人脸中水平方向的高频分量很少，而 $k_x=1, k_y=0$ 对应的Gabor滤波器主要提取水平方向的纹理，因此最终的效果就是各个地方都只得到一个平坦的响应，因而最终图像较模糊。



(b) 生成的特征脸如图所示



用前10个主分量重建的人脸如图所示，可以看出前10个分量的重建效果很差，丢失了大部分人脸信息。

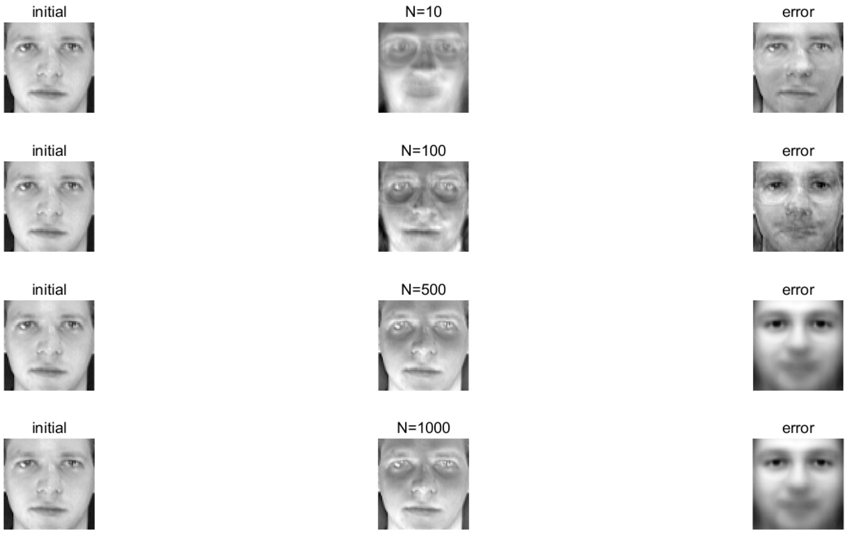
之后我查看了pca之后各个分量对应的特征值，如下所示，可以看出来前边的分量都很大，前十个不足以提取出主要分量。

lambda

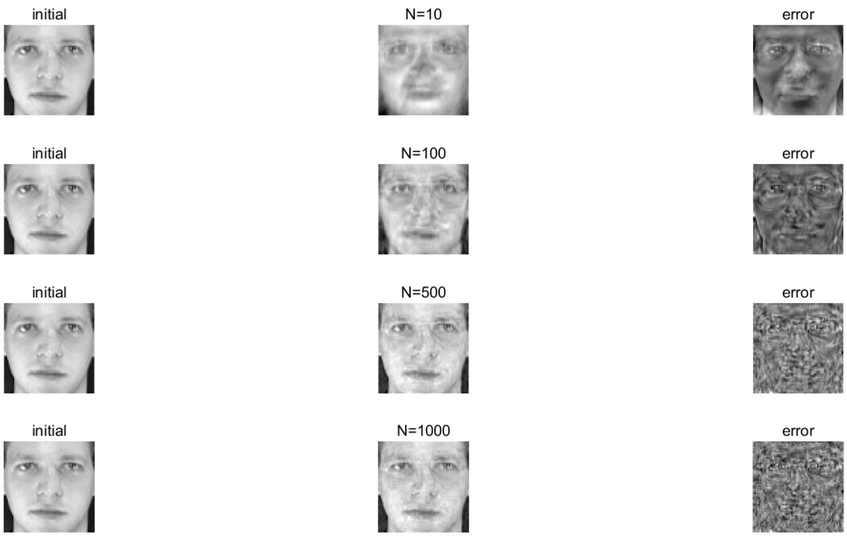
10x410 double

	1	2	3	4	5	6	7	8	9
1	1.1034e+06	7.3914e+04	3.2368e+04	1.8627e+04	1.2036e+04	9.1207e+03	7.2017e+03	5.9043e+03	4.9363
2	6.4841e+05	6.6582e+04	2.8479e+04	1.7801e+04	1.1769e+04	8.8701e+03	7.0537e+03	5.8489e+03	4.8510
3	3.6922e+05	5.7767e+04	2.7056e+04	1.6771e+04	1.1331e+04	8.5741e+03	6.9482e+03	5.7363e+03	4.7113
4	2.3160e+05	5.3152e+04	2.6398e+04	1.5501e+04	1.1113e+04	8.2911e+03	6.8442e+03	5.6220e+03	4.5770
5	1.6726e+05	4.9248e+04	2.5308e+04	1.4845e+04	1.0698e+04	8.2294e+03	6.6077e+03	5.4621e+03	4.5168
6	1.4628e+05	4.5302e+04	2.4643e+04	1.4644e+04	1.0491e+04	7.9919e+03	6.5145e+03	5.3230e+03	4.4653
7	1.1245e+05	4.2584e+04	2.3808e+04	1.4139e+04	1.0300e+04	7.8113e+03	6.3451e+03	5.2907e+03	4.4296
8	9.4356e+04	3.7789e+04	2.2975e+04	1.3837e+04	1.0214e+04	7.6067e+03	6.2076e+03	5.1053e+03	4.3235
9	9.0729e+04	3.4931e+04	2.1206e+04	1.3073e+04	9.6803e+03	7.5009e+03	6.0659e+03	5.0822e+03	4.2252
10	7.7477e+04	3.4611e+04	2.0442e+04	1.2696e+04	9.4978e+03	7.3392e+03	5.9642e+03	5.0330e+03	4.2098
11									

于是我尝试了不同数量的分量来重建人脸，可以发现N越大，重建效果越好。



但是有一个奇怪的问题就是：从上面的图片可以看出来，即使N取到了1000，重建的误差还是比较大，我不知道是不是因为函数 princomp 的问题，于是我又自己实现了一个 PCA 函数 mypca（代码附在最后）。重新计算后我发现我实现的 mypca 的运算结果与 matlab 自带函数 princomp 的运算结果，得到的特征向量、特征值、投影后的系数都相同，但是重建人脸的效果却差别较大，如下是利用 mypca 得到的重建人脸，可以看出效果更好。我最终也没有分析出这两者为什么会有如此大的差别，这也是我比较困惑的一点。



附录：源代码

为求解各个题目的matlab的源代码如下

```
%% Harr WT
% harr小波分析矩阵
clear all; close all; clc;
H0 = 1;
I0 = eye(1);
for N = 1:3
```

```
    shape = 2.^(N-1);
    L = zeros(shape,shape*2);
    B = zeros(shape,shape*2);
    for i = 1:size(L,1)
        L(i,2*(i-1)+1:2*i) = 1;
        B(i,2*(i-1)+1) = 1;
        B(i,2*i) = -1;
    end
    A = 1/sqrt(2)*[H0,zeros(size(H0,1),size(I0,2)); zeros(size(I0,2),size(H0,1)),I0];
    H1 = A * [L; B]; % H1 = sym(A * [L; B]);
    H0 = H1;
    I0 = eye(shape*2);
    disp(['----------- N = ',num2str(N),' ----------']);
    disp(L);    disp(B);    disp(H1);    disp(inv(H1));
end

v = [7,11,9,6,4,3,1,3]';
w = H1 * v;
disp(w');

%% PCA and LDA
clear all; close all; clc;
X1 = [1, 2, -5; 5, 9, -3];
X2 = [2, -1, 0; -3, -4, 2];
% PCA
X = [X1,X2];
m = size(X,2);
Xmean = mean(X,2)*ones(1,size(X,2));
S = 1/m * ((X-Xmean) * (X-Xmean)');     % convariance matrix
[P,D] = eig(S);                         % 对角化
[idx,~] = find(D == max(max(D)));       % 寻找最大特征值对应的index
Y = P(:,idx) * (P(:,idx)' * X);  % project
disp('--------- PCA ---------');
disp(mean(X,2)); disp(S); disp(P); disp(D);
figure; hold on;
title('PCA');
plot(X(1,:),X(2,:),'o','MarkerFaceColor','r');
plot(Y(1,:),Y(2,:),'d','MarkerFaceColor','b');
plot(Y(1,:),Y(2,:));
axis equal;

% LDA
mu1 = mean(X1,2)*ones(1,size(X1,2));
mu2 = mean(X2,2)*ones(1,size(X2,2));
Sw = (X1-mu1)*(X1-mu1)' + (X2-mu2)*(X2-mu2)';             % 类内散度
Sb = (mean(X1,2)-mean(X2,2)) * (mean(X1,2)-mean(X2,2))';  % 类间散度
S = Sw\Sb;
[P,D] = eig(S);                         % 对角化
[idx,~] = find(D == max(max(D)));       % 寻找最大特征值对应的index
Y1 = P(:,idx) * (P(:,idx)' * X1);       % project
Y2 = P(:,idx) * (P(:,idx)' * X2);       % project
disp('--------- LDA ---------');
disp(mu1); disp(mu2); disp(Sw); disp(Sb);
```

```matlab
disp(S); disp(P); disp(D);
figure; hold on;
title('LDA');
plot(X1(1,:),X1(2,:),'ro','MarkerFaceColor','r');
plot(X2(1,:),X2(2,:),'bo','MarkerFaceColor','b');
plot(Y1(1,:),Y1(2,:),'^','MarkerFaceColor','r');
plot(Y2(1,:),Y2(2,:),'v','MarkerFaceColor','b');
plot([Y1(1,:),Y2(1,:)],[Y1(2,:),Y2(2,:)]);
axis equal;

%% Gabor filters
clear all; close all; clc;
load('ORL_64x64.mat');
im = reshape(fea(1,:),[64,64]);
kxx = [0,1,1];
kyy = [1,0,1];
sigma = [0.5,2,5];

figure(1);
for i = 1:3
    sigm = sigma(2);
    kx = kxx(i);
    ky = kyy(i);
    [x,y] = meshgrid(-round(3*sigm):round(3*sigm));
    G_even = cos(kx*x + ky*y) * exp(-(x.^2+y.^2)/2/sigm.^2);
    G_odd = sin(kx*x + ky*y) * exp(-(x.^2+y.^2)/2/sigm.^2);
    fe = filter2(G_even, im);
    fo = filter2(G_odd, im);

    subplot(3,3,3*(i-1)+1); imshow(im, []); title(['kx=',num2str(kx)]);
    subplot(3,3,3*(i-1)+2); imshow(fe, []); title(['ky=',num2str(ky)]);
    subplot(3,3,3*(i-1)+3); imshow(fo, []); title(['\sigma=',num2str(sigm)]);
end

[U,Z,lambda] = princomp(fea);    % pca变换

figure(2);
for i = 1:10
    eigface = Z(1,i)*U(:,i)';
    im = reshape(eigface,[64,64]);
    subplot(2,5,i); imshow(im, []);
end

eigface = Z(1,1:10)*U(:,1:10)';
im = reshape(eigface,[64,64]);
figure(3); imshow(im, []);
```

```matlab
% mypca.m
function [P,Z,D] = mypca(X)
% X 每一行是一组数据
    mu = mean(X);
    Y = X - ones(size(X,1),1) * mu;
    Sigma = Y' * Y / size(X,1);
    [P,D] = eig(Sigma);          % 对角化,特征值从小到大排列
    P = rot90(P)';               % 镜像对称特征向量
    D = diag(rot90(rot90(D)));   % 重排特征值
    Z = X * P;
end
```