AMATH 482
Homework 4 Report
Fangzheng Sun
1239942

Abstract

In this project, I am going to deal with some videos of string movements filmed by three cameras, using SVD methods. In test 1 and 2, only vertical movements are dominant and in test 3 and 4, the motions in 3 dimensions are considered.

Introduction

Professor execute some string experiments and filmed them by 3 cameras in 3 directions. The 4 sets of films, separately are ideal case, noised case, with horizontal displacement and with horizontal displacement and rotation.

Theoretical background

The following theories are included in this project:

1. The 4-D matrix of the video: after reading the film in MATLAB, there will be a 4-D matrix containing 8-unit integers. The 4 dimensions are separately Y-position, X-position, RGB number and phrase.
2. Singular Value Decomposition (SVD): SVD is a factorization of matrix into a number of constitutive components all of which have a specific meaning in applications. For a matrix A (m×n),

$$\mathrm{A} = \mathrm{U}\Sigma V^*$$

   The SVD decomposition of the matrix A thus shows that the matrix first applies a unitary transformation preserving the unit sphere via V*. This is followed by a stretching operation that creates an ellipse with principal semiaxes given by the matrix Σ. Finally, the generated hyperellipse is rotated by the unitary transformation U.

3. Principle Component Analysis (PCA): PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. In this 3-camera case, the data are collected in the matrix X as the following,

$$X = \begin{cases} X1 \\ Y1 \\ X2 \\ Y2 \\ X3 \\ Y3 \end{cases}$$

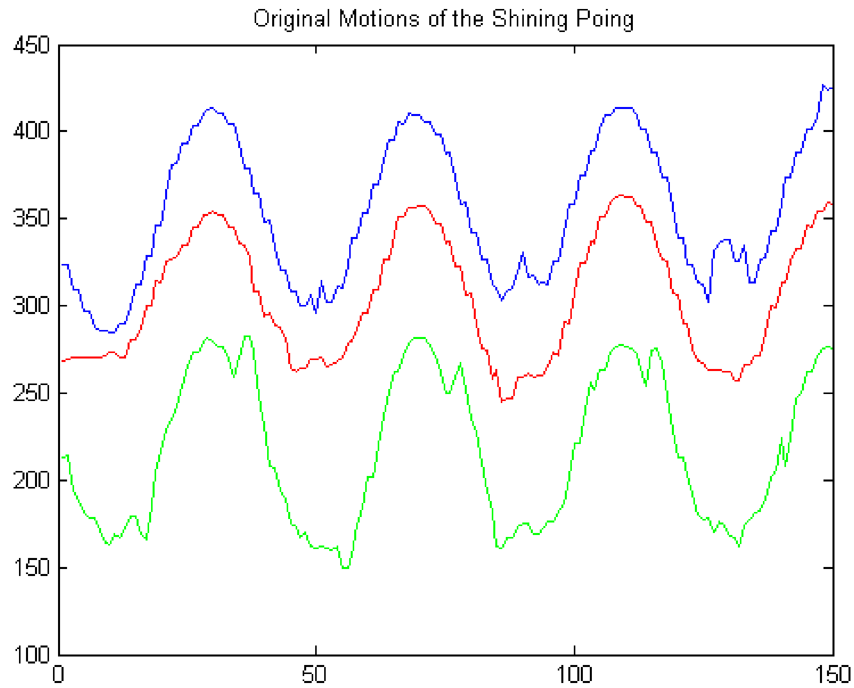
X1 and Y1 are x and y positions of the target in the film of camera 1, X2 and Y2 are x and y positions of the target in the film of camera 2, X3 and Y3 are x and y positions of the target in the film of camera 3. As I get the sigma value, I can compute the energy, which is the ratio of the principle value(s) in the sum of all sigma values. Finally, from the graph of the motion after projection (matrix Y), we can easily see the dominant motion of one row or some rows (in red).

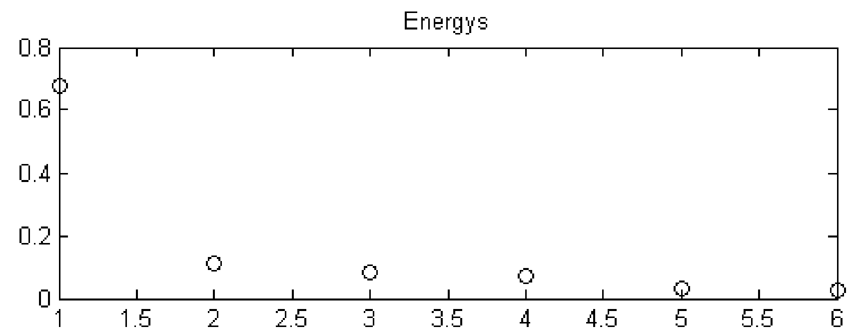
Test 1:

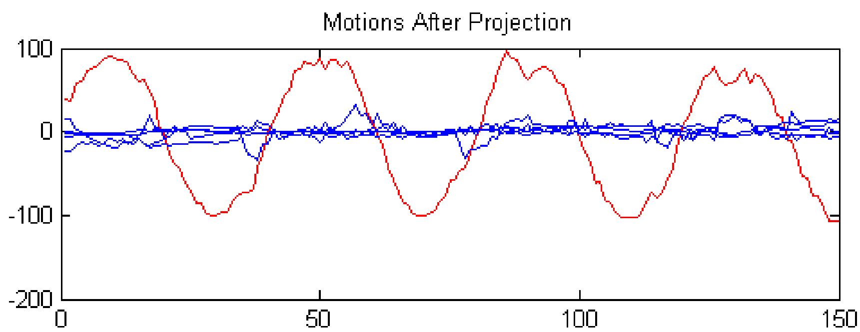
Algorithm Implementation and Development

In this test, the films are pretty ideal and noises are low. I first transfer the matrix RGB to gray and set thresholds for each cameras to track the shining point above the bucket and record the x (vertical) positions in camera 1 and 2 and y (horizontal) positions in camera 3 because camera3 is filmed 90 degree different from the other 2 cameras. Some other things, such as professor's belt and the bottom of the chair are as light as the point. Thus I need to crop the films for each cameras, camera 1: [211:370, 301, 360], camera 2: [81:300, 251:350], camera3: [231:300, 251:450]. And in a for-loop where I go through all the phrases, I use a method in which I set a threshold of the color number which is pretty high thus filtering all other points except for the shining area. The averaging of the coordinates is exactly the coordinate of the center of shining are. And I add up the cropped coordinate so that the x and y coordinates I get from the for-loop are back to the original size of the films. To ensure the three sin-wave-like motion tracks are simultaneous, I cut the first 10 phrases of the camera 2 after reading the figure so that they start basically at the same time. Since the films have different lengths, 226, 284 and 232, I only take 150 phases so that they also stop simultaneously. After then, I put the 6 rows (X1, Y1, X2, Y2, X3, Y3) coordinates into X and execute PCA to the matrix X. Finally I gain 6 lambda values and their square root sigma values, where the first one (largest one) should be the principle component. For figure 2, I choose to plot the energy of the 6 sigma values and then plot the motion after the projection (Y matrix).

Computational Results







The above figures are from test 1. The energy of the principle sigma value is 0.6757. From figure 2, we can see that in the ideal case, the principle value has the dominant effect on the motion and the first row of Y has the dominant effect on the motion (in red), thus illustrating that the motion in the experiment is basically a 1 dimension motion in vertical direction.

Test 2:

Algorithm Implementation and Development

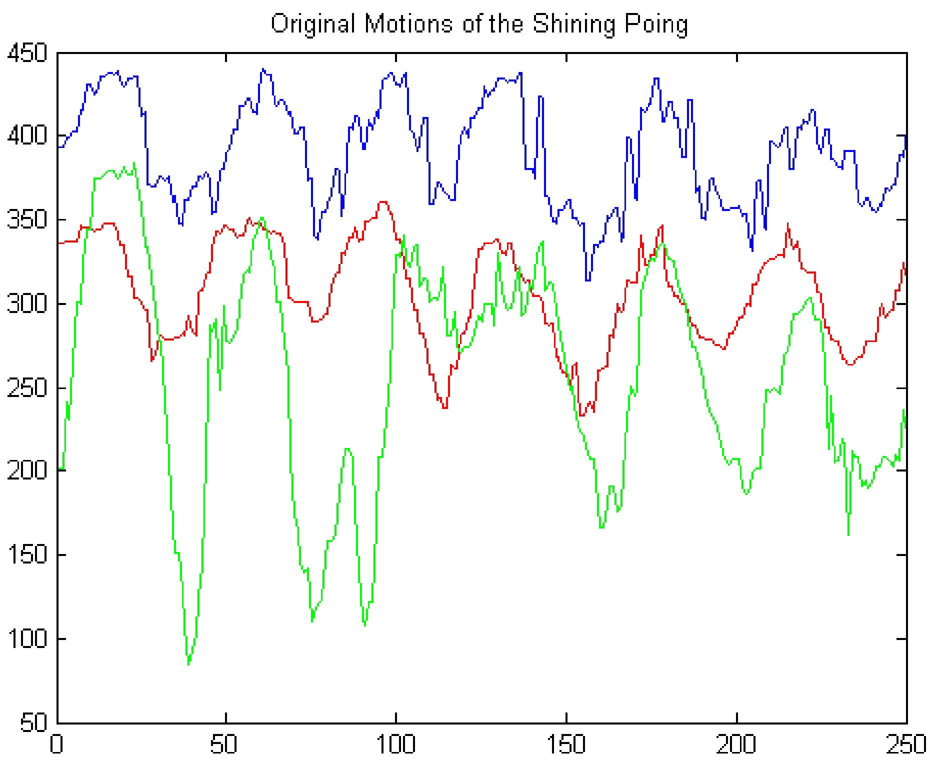
The difference between tests 1 and 2 is that in test 2, shakings exists in the films so that there are a lot of noises. The basic idea of this test is the same as the test 1, only some details are changed here:

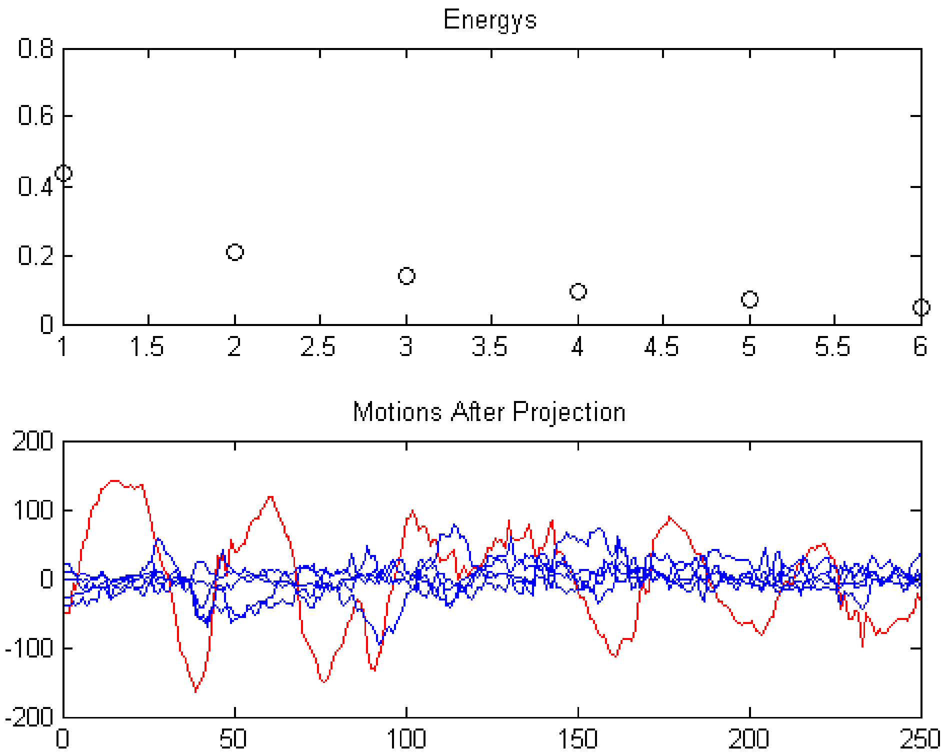
Cropping the film: camera 1: [211:370, 301, 400], camera 2: [:, 201:400], camera3: [201:300, 251:450].

To make the three curves start simultaneously and be in the same phase, I cut the first 20 phrases in the camera 2.

The lengths of the three films are 314,356,327. I take first 250 phases so that they also stop simultaneous.

Computational Results





The above figures are from test 2. From the first figure, we can see that although the curves are basically sin-wave-like, they are much less regular than the figure in the first test. And the energy of the principle sigma value is 0.4366, much lower than the energy in the first test. These this can be explained by the existence of the noises in forms of shaking. From figure 2, we can see that in the noised case, the principle value is less dominant. The red curve in the figure of motions after projection starts to lose its dominance from around the 100^{th} phrase to the 150^{th} phrase, where the shakings are large enough to disturb the observation of the motion.

Test 3:

Algorithm Implementation and Development

In the third test, the horizontal displacement is added to the experiment, which makes the main idea a little different from the first and second test in the following items:

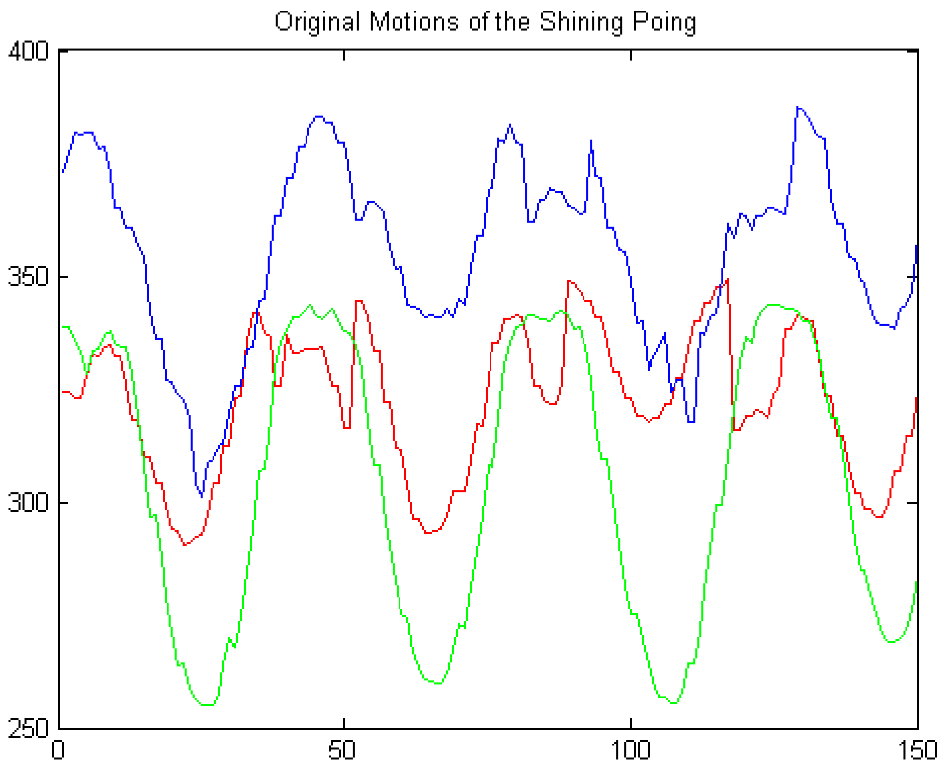
Cropping the film: camera 1: [231:350, 351,350], camera 2: [151:350, 201:400], camera3: [151:300, 251:400].

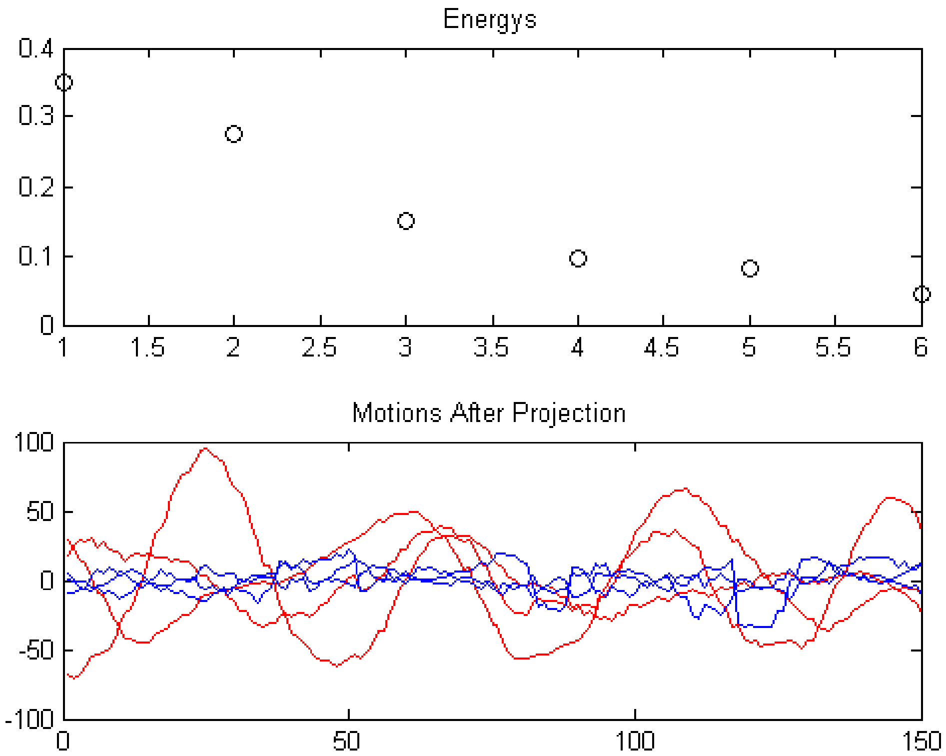
To make the three curves start simultaneously and be in the same phase, I cut the first 15 phrases in the camera 1 and the first 5 phases in camera 3.

The lengths of the three films are 239,281,237. I take first 150 phases so that they also stop simultaneously.

For the part of PCA, the difference is that after addition of horizontal motion, the mass is released off-center so as to produce motion in the x−y plane as well as the z direction, where 3 dimensions are dominant in this case. As a result, there is no longer a singular principle component. Instead, the first 3 largest sigma values should be considered, thus 3 principle components in this case. And we compute for the energy of the 3 principle components.

Computational Results





The above figures are from test 3. The energy of the 3 principle components are 0.3507, 0.2760 and 0.1512, adding up to 0.7779. The red curve in the figure of motions after projection shows how the three dominant rows in the matrix Y affect the overall motion of the mass, in three dimensions.

Test 4:

Algorithm Implementation and Development

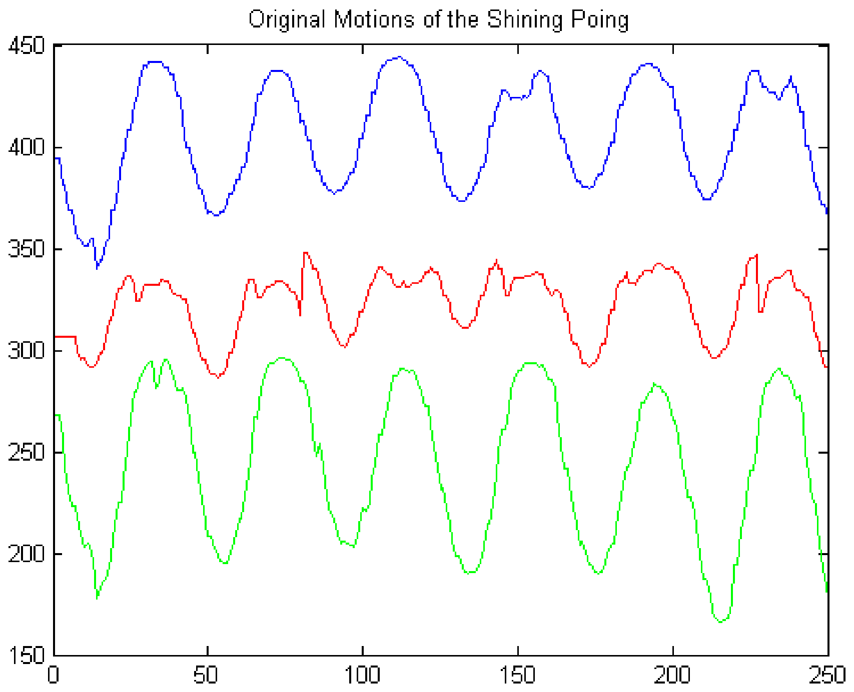
Similar as test 3, the motion in this test is considered as a 3D one. With the horizontal rotation, the movement is like a helix.

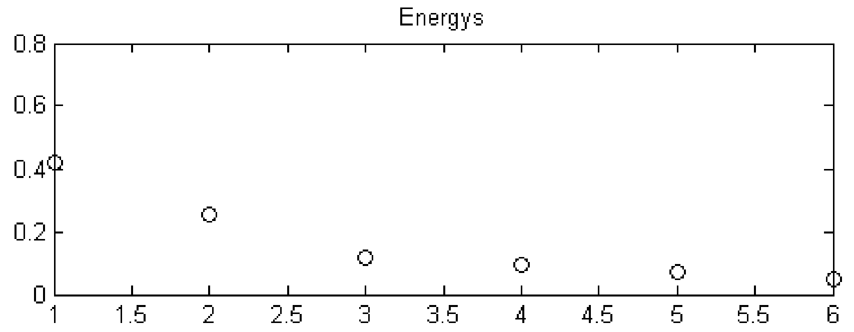
Cropping the film: camera 1: [231:350, 301,350], camera 2: [101:300, 201:450], camera3: [151:300, 251:450].

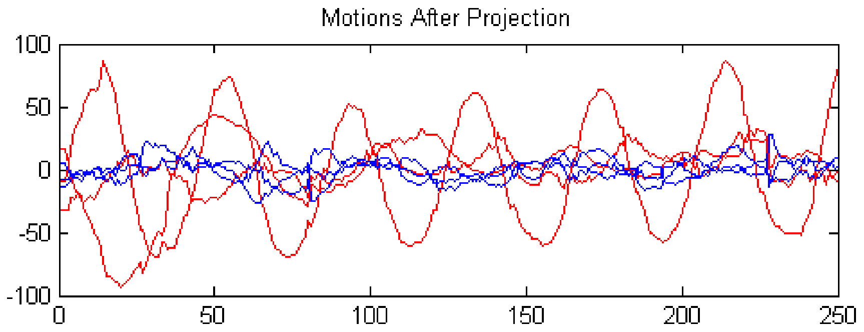
To make the three curves start simultaneously and be in the same phase, I cut the first 5 phrases in the camera 2.

The lengths of the three films are 392,405,394. I take first 250 phases so that they also stop simultaneously.

Computational Results







The above figures are from test 4. The energy of the 3 principle components are 0.4207, 0.2518 and 0.1179, adding up to 0.7905. The red curve in the figure of motions after projection shows how the three dominant rows in the matrix Y affect the overall motion of the mass, in three dimensions.

Summary and conclusions

Applying the PCA to the films can help us understand how the information of the cameras overlap with each other and select the dominant motions. The report help me understand the logarithm of PCA and how to use PCA to real-life problems.

Appendix A: MATLAB functions

find - access and display values of simulation results

repmat - replicate and tile categorical array

eig - eigenvalues and eigenvectors

Appendix B: MATLAB codes

```
%% Test 1
clear all; close all; clc;

figure(1) %track of the motion of the shining point
%camera1
load('cam1_1.mat');
vidFrames1_1=vidFrames1_1(211:370,301:360,:,:);
[m,n,p,t]=size(vidFrames1_1);
for j=1:t
    grayFrames1_1=rgb2gray(vidFrames1_1(:,:,:,j));
    [I,J] = find(grayFrames1_1>253);
    X1_1(j)=210+mean(I(:));
```

```
    Y1_1(j)=300+mean(J(:));
end
plot(1:150,X1_1(1:150),'r');
hold on;

%camera2
load('cam2_1.mat');
vidFrames2_1=vidFrames2_1(81:300,251:350,:,:);
[m,n,p,t]=size(vidFrames2_1);
for j=1:t
    grayFrames2_1=rgb2gray(vidFrames2_1(:,:,:,j));
    [I,J] = find(grayFrames2_1>250);
    X2_1(j)=80+mean(I(:));
    Y2_1(j)=250+mean(J(:));
end
plot(1:150,X2_1(11:160),'g');
hold on;

%camera3
load('cam3_1.mat');
vidFrames3_1=vidFrames3_1(231:300,251:450,:,:);
[m,n,p,t]=size(vidFrames3_1);
for j=1:t
    grayFrames3_1=rgb2gray(vidFrames3_1(:,:,:,j));
    [I,J] = find(grayFrames3_1>246);
    X3_1(j)=230+mean(I(:));
    Y3_1(j)=250+mean(J(:));
end
plot(1:150,Y3_1(1:150));
title('Original Motions of the Shining Poing')

%SVD
X(1,:)=X1_1(1:150);
X(2,:)=Y1_1(1:150);
X(3,:)=X2_1(11:160);
X(4,:)=Y2_1(11:160);
X(5,:)=X3_1(1:150);
X(6,:)=Y3_1(1:150);
[m,n]=size(X); % compute data size
mn=mean(X,2); % compute mean for each row
X=X-repmat(mn,1,n); % subtract mean
Cx=(1/(n-1))*X*X'; % covariance
[V,D]=eig(Cx); % eigenvectors(V)/eigenvalues(D)
lambda=diag(D); % get eigenvalues
```

```
[dummy,m_arrange]=sort(-1*lambda); % sort in decreasing order
lambda=lambda(m_arrange);
sig=sqrt(lambda);
V=V(:,m_arrange);
Y=V'*X; % produce the principal components projection

figure(2)
%Plot the energys
subplot(2,1,1)
plot(1:6,sig/(sum(sig(:))),'ko');
title('Energys');
energy1=sig(1)/(sum(sig(:)));
%Plot the motions after applying the principal components projection
subplot(2,1,2)
plot(Y(1,:),'r');hold on %donimant phases ,in red
plot(Y(2,:));hold on %other phases, in blue
plot(Y(3,:));hold on
plot(Y(4,:));hold on
plot(Y(5,:));hold on
plot(Y(6,:));
title('Motions After Projection');

%% Test 2
clear all; close all; clc;

figure(1) %track of the motion of the shining point
%camera1
load('cam1_2.mat');
[m,n,p,t]=size(vidFrames1_2);
vidFrames1_2=vidFrames1_2(211:370,301:400,:,:);
[m,n,p,t]=size(vidFrames1_2);
for j=1:t
   grayFrames1_2=rgb2gray(vidFrames1_2(:,:,:,j));
    [I,J] = find(grayFrames1_2>252);
   X1_2(j)=210+mean(I(:));
   Y1_2(j)=300+mean(J(:));
end
plot(1:250,X1_2(1:250),'r');
hold on;

%camera2
load('cam2_2.mat');
vidFrames2_2=vidFrames2_2(:,201:400,:,:);
[m,n,p,t]=size(vidFrames2_2);
```

```
for j=1:t
    grayFrames2_2=rgb2gray(vidFrames2_2(:,:,:,j));
    [I,J] = find(grayFrames2_2>249);
    X2_2(j)=mean(I(:));
    Y2_2(j)=200+mean(J(:));
end
plot(1:250,X2_2(21:270),'g');
hold on;

%camera3
load('cam3_2.mat');
vidFrames3_2=vidFrames3_2(201:300,251:450,:,:);
[m,n,p,t]=size(vidFrames3_2);
for j=1:t
    grayFrames3_2=rgb2gray(vidFrames3_2(:,:,:,j));
    [I,J] = find(grayFrames3_2>246);
    X3_2(j)=200+mean(I(:));
    Y3_2(j)=250+mean(J(:));
end
plot(1:250,Y3_2(1:250));
title('Original Motions of the Shining Poing')

%SVD
X(1,:)=X1_2(1:250);
X(2,:)=Y1_2(1:250);
X(3,:)=X2_2(21:270);
X(4,:)=Y2_2(21:270);
X(5,:)=X3_2(1:250);
X(6,:)=Y3_2(1:250);
[m,n]=size(X); % compute data size
mn=mean(X,2); % compute mean for each row
X=X-repmat(mn,1,n); % subtract mean
Cx=(1/(n-1))*X*X'; % covariance
[V,D]=eig(Cx); % eigenvectors(V)/eigenvalues(D)
lambda=diag(D); % get eigenvalues
[dummy,m_arrange]=sort(-1*lambda); % sort in decreasing order
lambda=lambda(m_arrange);
sig=sqrt(lambda);
V=V(:,m_arrange);
Y=V'*X; % produce the principal components projection

figure(2)
%Plot the energys
subplot(2,1,1)
```

```
plot(1:6,sig/(sum(sig(:))),'ko');
title('Energys');
energy2=sig(1)/(sum(sig(:)));
%Plot the motions after applying the principal components projection
subplot(2,1,2)
plot(Y(1,:),'r');hold on %donimant phases ,in red
plot(Y(2,:));hold on %other phases, in blue
plot(Y(3,:));hold on
plot(Y(4,:));hold on
plot(Y(5,:));hold on
plot(Y(6,:));
title('Motions After Projection');

%% Test 3
clear all; close all; clc;

figure(1) %track of the motion of the shining point
%camera1
load('cam1_3.mat');
vidFrames1_3=vidFrames1_3(231:350,251:350,:,:);
[m,n,p,t]=size(vidFrames1_3);
for j=1:t
    grayFrames1_3=rgb2gray(vidFrames1_3(:,:,:,j));
    [I,J] = find(grayFrames1_3>248);
    X1_3(j)=230+mean(I(:));
    Y1_3(j)=250+mean(J(:));
end
plot(1:150,X1_3(16:165),'r');
hold on;

%camera2
load('cam2_3.mat');
vidFrames2_3=vidFrames2_3(151:350,201:400,:,:);
[m,n,p,t]=size(vidFrames2_3);
for j=1:t
    grayFrames2_3=rgb2gray(vidFrames2_3(:,:,:,j));
    [I,J] = find(grayFrames2_3>250);
    X2_3(j)=150+mean(I(:));
    Y2_3(j)=200+mean(J(:));
end
plot(1:150,X2_3(1:150),'g');
hold on;

%camera3
```

```
load('cam3_3.mat');
vidFrames3_3=vidFrames3_3(151:350,251:400,:,:);
[m,n,p,t]=size(vidFrames3_3);
for j=1:t
    grayFrames3_3=rgb2gray(vidFrames3_3(:,:,:,j));
    [I,J] = find(grayFrames3_3>249);
    X3_3(j)=150+mean(I(:));
    Y3_3(j)=250+mean(J(:));
end
plot(1:150,Y3_3(6:155));
title('Original Motions of the Shining Poing')

%SVD
X(1,:)=X1_3(16:165);
X(2,:)=Y1_3(16:165);
X(3,:)=X2_3(1:150);
X(4,:)=Y2_3(1:150);
X(5,:)=X3_3(6:155);
X(6,:)=Y3_3(6:155);
[m,n]=size(X); % compute data size
mn=mean(X,2); % compute mean for each row
X=X-repmat(mn,1,n); % subtract mean
Cx=(1/(n-1))*X*X'; % covariance
[V,D]=eig(Cx); % eigenvectors(V)/eigenvalues(D)
lambda=diag(D); % get eigenvalues
[dummy,m_arrange]=sort(-1*lambda); % sort in decreasing order
lambda=lambda(m_arrange);
sig=sqrt(lambda);
V=V(:,m_arrange);
Y=V'*X; % produce the principal components projection

figure(2)
%Plot principle sigma square
subplot(2,1,1)
plot(1:6,sig/(sum(sig(:))),'ko');
title('Energys');
energy3=sig(1:3)/(sum(sig(:)));
%Plot the principle component signal
subplot(2,1,2)
plot(Y(1,:),'r');hold on %three donimant phases ,in red
plot(Y(2,:),'r');hold on
plot(Y(3,:),'r');hold on
plot(Y(4,:));hold on    %other phases, in blue
plot(Y(5,:));hold on
```

```
plot(Y(6,:));
title('Motions After Projection');

%% Test 4
clear all; close all; clc;

figure(1) %track of the motion of the shining point
%camera1
load('cam1_4.mat');
vidFrames1_4=vidFrames1_4(231:350,301:450,:,:);
[m,n,p,t]=size(vidFrames1_4);
for j=1:t
    grayFrames1_4=rgb2gray(vidFrames1_4(:,:,:,j));
    [I,J] = find(grayFrames1_4>233);
    X1_4(j)=230+mean(I(:));
    Y1_4(j)=300+mean(J(:));
end
plot(1:250,X1_4(1:250),'r');
hold on;

%camera2
load('cam2_4.mat');
vidFrames2_4=vidFrames2_4(101:300,201:450,:,:);
[m,n,p,t]=size(vidFrames2_4);
for j=1:t
    grayFrames2_4=rgb2gray(vidFrames2_4(:,:,:,j));
    [I,J] = find(grayFrames2_4>250);
    X2_4(j)=100+mean(I(:));
    Y2_4(j)=200+mean(J(:));
end
plot(1:250,X2_4(6:255),'g');
hold on;

%camera3
load('cam3_4.mat');
vidFrames3_4=vidFrames3_4(151:300,251:450,:,:);
[m,n,p,t]=size(vidFrames3_4);
for j=1:t
    grayFrames3_4=rgb2gray(vidFrames3_4(:,:,:,j));
    [I,J] = find(grayFrames3_4>234);
    X3_4(j)=150+mean(I(:));
    Y3_4(j)=250+mean(J(:));
end
plot(1:250,Y3_4(1:250));
```

```
title('Original Motions of the Shining Poing')

%SVD
X(1,:)=X1_4(1:250);
X(2,:)=Y1_4(1:250);
X(3,:)=X2_4(6:255);
X(4,:)=Y2_4(6:255);
X(5,:)=X3_4(1:250);
X(6,:)=Y3_4(1:250);
[m,n]=size(X); % compute data size
mn=mean(X,2); % compute mean for each row
X=X-repmat(mn,1,n); % subtract mean
Cx=(1/(n-1))*X*X'; % covariance
[V,D]=eig(Cx); % eigenvectors(V)/eigenvalues(D)
lambda=diag(D); % get eigenvalues
[dummy,m_arrange]=sort(-1*lambda); % sort in decreasing order
lambda=lambda(m_arrange);
sig=sqrt(lambda);
V=V(:,m_arrange);
Y=V'*X; % produce the principal components projection

figure(2)
%Plot principle sigma square
subplot(2,1,1)
plot(1:6,sig/(sum(sig(:))),'ko');
title('Energys');
energy4=sig(1:3)/(sum(sig(:)));
%Plot the principle component signal
subplot(2,1,2)
plot(Y(1,:),'r');hold on %three donimant phases ,in red
plot(Y(2,:),'r');hold on
plot(Y(3,:),'r');hold on
plot(Y(4,:));hold on    %other phases, in blue
plot(Y(5,:));hold on
plot(Y(6,:));
title('Motions After Projection');
```