AMATH 582 Homework 3

Junzhao Liu

February 23, 2021

Abstract

The goal of this homework is analyzing the spring-mass system as described in lecture, observed by three different cameras to conclude the effects of noise on the PCA algorithms.

1 Introduction and Overview

the principal component analysis (PCA) algorithms are a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets. In this a specific situation, In the frame of each video, there are two dimensions, therefore we have six-dimensions in total to represent the one-dimension motion. The SVD will help us weed out the redundancies and take out the core information.

1.1Find the Coordinates of the mass position

The light spot in the video can be considered as mass position, by find the coordinate of that spot can reach our goal, first of all, we need to create a location function which takes in vidFrame, filter and scale and gives us an output of two-dimensional matrix. Inside the function, we should turn each vidFrames to a grey picture, apply the reasonable filter as needed and locate the X, Y coordinate of the white spot

1.2 Determining how much energy from the full system is contained in each rank

Assuming the full matrix is rank r, the energy contained in the rank-N approximation is represented by:

$$\text{energy}_N = \frac{\sigma_1^2 + \sigma_2^2 + \dots \sigma_N^2}{\sigma_1^2 + \sigma_2^2 + \dots \sigma_r^2} = \frac{\|X_N\|_F^2}{\|X\|_F^2}.$$

1.3Lower the dimension of matrix

SUV decomposition gives us a decomposition of the matrix A through unitary matrices U and V and a diagonal matrix Σ

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

2 Theoretical Background

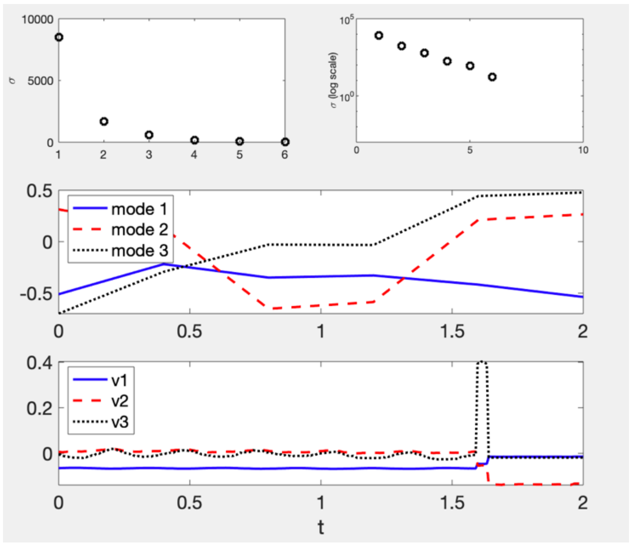
V is a unitary matrix withv1andv2as its columns,U is unitary matrix withu1andu2as its columns, and Σ is a diagonal matrix with σj on its diagonals.

3 Algorithm Implementation and Development

1. Use `function rgb2gray turn rgb vidFrame graph to gray`
2. Use find function to locate the center of mass
3. Use SUV decomposition to find the diagonal matrix
4. analyze diagonal vector by the function plot() and semilogy()
5.plot the graph based on previous analysis and make the conclusion

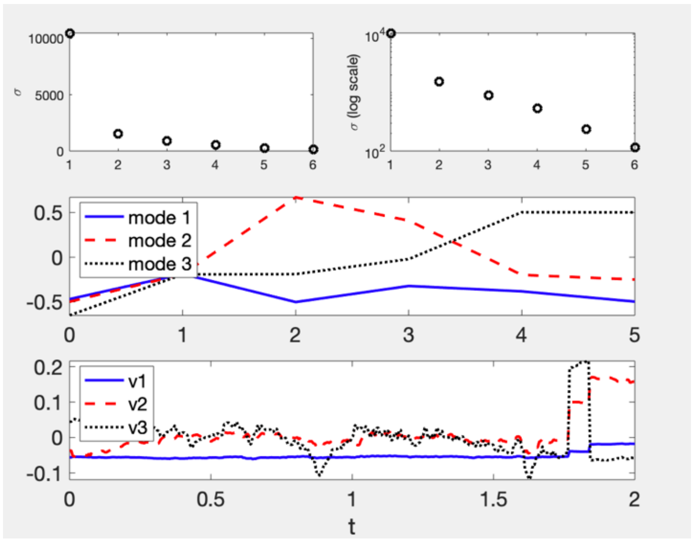
4 Computational Results

Test 1



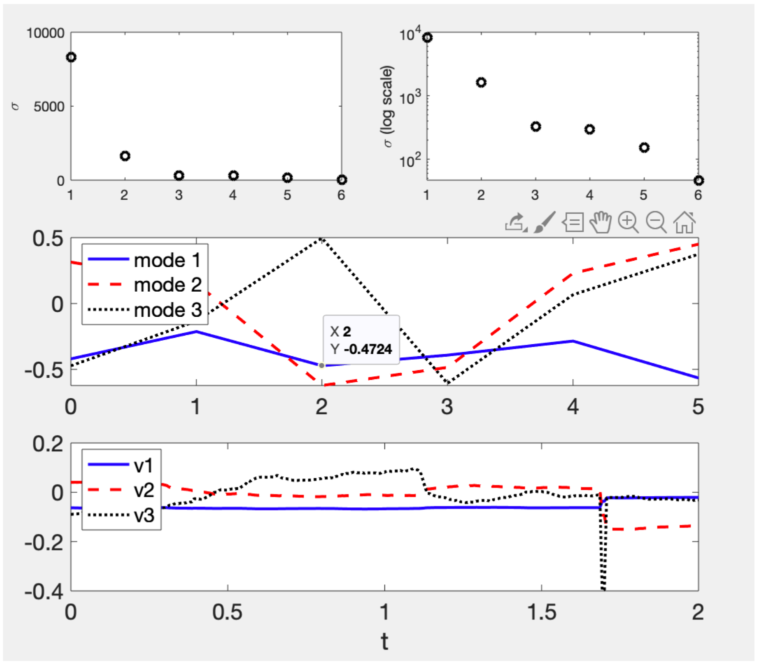
Only the first two singular values are significant, the rest values are close to zero, mode2 looks more smooth than other two modes

Test 2



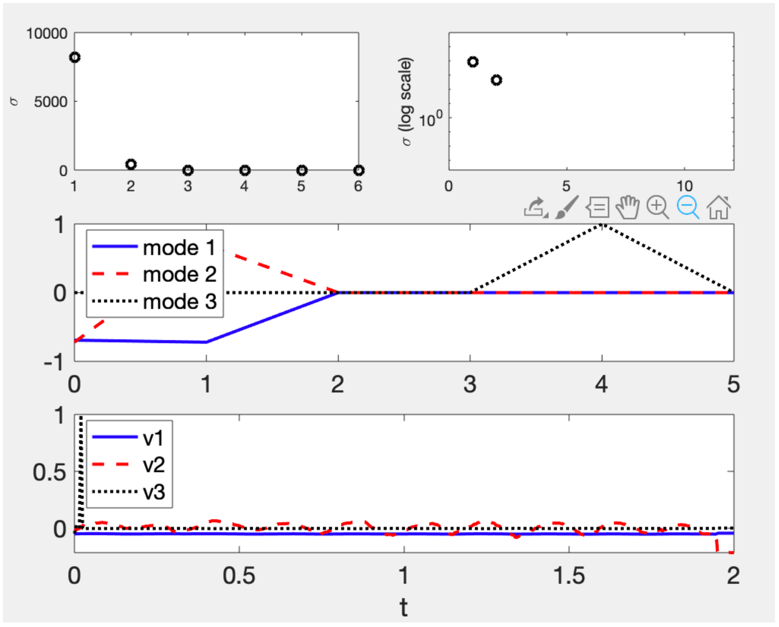
Because of the noise singular values are all very significant, and the second plot can clear see v3 looks like noise

Test 3



singular values are all very significant, mode 3 looks more smooth than other ons

Test4



Only first two singular values are significant, hard to observe any trend.

Summary and Conclusions

Based on the four graphs of different cases, we can observed the noise affects the result of SVD decomposition

Appendix A MATLAB Functions

[U,S,V] = svd(A) performs a singular value decomposition of matrix A, such that A = U*S*V'.

y = linspace(x1,x2,n) generates n points. The spacing between the points is (x2-x1)/(n-1).

semilogy(X,Y) plots *x*- and *y*-coordinates using a linear scale on the *x*-axis and a base-10 logarithmic scale on the *y*-axis.

- To plot a set of coordinates connected by line segments, specify X and Y as vectors of the same length.
- To plot multiple sets of coordinates on the same set of axes, specify at least one of X or Y as a matrix.

plot(X,Y) creates a 2-D line plot of the data in Y versus the corresponding values in X.

I = rgb2gray(RGB) converts the truecolor image RGB to the grayscale image I. The rgb2gray function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

Appendix B MATLAB code

```
clear all; clc
%question one
load cam1_1.mat;
load cam2_1.mat;
load cam3_1.mat;
load cam1_2.mat;
load cam2_2.mat;
load cam3_2.mat;
load cam1_3.mat;
load cam2_3.mat;
load cam3_3.mat;
load cam1_4.mat;
load cam2_4.mat;
load cam3_4.mat;

matrix1 = zeros(284,6);
matrix1(1:226,1:2)  = location(vidFrames1_1,1,200);
matrix1(1:284,3:4) = location(vidFrames2_1,1,200);
matrix1(1:232,5:6) = location(vidFrames3_1,1,200);
%question 1
[U,S,V] = svd(matrix1','econ');
sig = diag(S);
subplot(3,2,1)
plot(sig,'ko','Linewidth',2)
ylabel('\sigma')
```

```
subplot(3,2,2)
semilogy(sig,'ko','Linewidth',2)
axis([0 10 10^-3 10^5])
ylabel('\sigma (log scale)')
set(gca,'Fontsize',8,'Xtick',0:5:25,'Ytick',logspace(-15,5,5))
subplot(3,1,2)
x = linspace(0,2,6);
t = linspace(0,2,284);
plot(x,U(:,1),'b',x,U(:,2),'--r',x,U(:,3),':k','Linewidth',2)
set(gca,'Fontsize',16)
legend('mode 1','mode 2','mode 3','Location','northwest')
subplot(3,1,3)
plot(t,V(:,1),'b',t,V(:,2),'--r',t,V(:,3),':k','Linewidth',2)
legend('v1','v2','v3','Location','northwest')
xlabel('t')
set(gca,'Fontsize',16)

%question 2
matrix2 = zeros(356,6);
matrix2(1:314,1:2) = location(vidFrames1_2,1,200);
matrix2(1:356,3:4) = location(vidFrames2_2,1,200);
matrix2(1:327,5:6)= location(vidFrames3_2,1,200);
[U,S,V] = svd(matrix2','econ');
sig = diag(S);
subplot(3,2,1)
plot(sig,'ko','Linewidth',2)
ylabel('\sigma')
subplot(3,2,2)
semilogy(sig,'ko','Linewidth',2)
%axis([0 10 10^-3 10^5])
ylabel('\sigma (log scale)')
subplot(3,1,2)
x = linspace(0,5,6);
t = linspace(0,2,356)
plot(x,U(:,1),'b',x,U(:,2),'--r',x,U(:,3),':k','Linewidth',2)
set(gca,'Fontsize',16)
legend('mode 1','mode 2','mode 3','Location','northwest')
subplot(3,1,3)
plot(t,V(:,1),'b',t,V(:,2),'--r',t,V(:,3),':k','Linewidth',2)
legend('v1','v2','v3','Location','northwest')
xlabel('t')
set(gca,'Fontsize',16)





%
% %queation3
matrix3 = zeros(281,6);
matrix3(1:239,1:2)= location(vidFrames1_3,1,200);
matrix3(1:281,3:4)= location(vidFrames2_3,1,200);
matrix3(1:237,5:6) = location(vidFrames3_3,1,200);
[U,S,V] = svd(matrix3','econ');
sig = diag(S);
subplot(3,2,1)
plot(sig,'ko','Linewidth',2)
ylabel('\sigma')
```

```
subplot(3,2,2)
semilogy(sig,'ko','Linewidth',2)
ylabel('\sigma (log scale)')
subplot(3,1,2)
x = linspace(0,5,6);
t = linspace(0,2,281)
plot(x,U(:,1),'b',x,U(:,2),'--r',x,U(:,3),':k','Linewidth',2)
set(gca,'Fontsize',16)
legend('mode 1','mode 2','mode 3','Location','northwest')
subplot(3,1,3)
plot(t,V(:,1),'b',t,V(:,2),'--r',t,V(:,3),':k','Linewidth',2)
legend('v1','v2','v3','Location','northwest')
xlabel('t')
set(gca,'Fontsize',16)


% %question4
matrix4 = zeros(405,6);
matrix4(1:392,1:2) = location(vidFrames1_4,1,200);
matrix4(1:405,1:2) = location(vidFrames2_4,1,200);
matrix4(1:394,1:2) = location(vidFrames3_4,1,200);
[U,S,V] = svd(matrix4','econ');
sig = diag(S);
subplot(3,2,1)
plot(sig,'ko','Linewidth',2)
ylabel('\sigma')
subplot(3,2,2)
semilogy(sig,'ko','Linewidth',2)
axis([0 10 10^-3 10^5])
ylabel('\sigma (log scale)')
subplot(3,1,2)
x = linspace(0,5,6);
t = linspace(0,2,405)
plot(x,U(:,1),'b',x,U(:,2),'--r',x,U(:,3),':k','Linewidth',2)
set(gca,'Fontsize',16)
legend('mode 1','mode 2','mode 3','Location','northwest')
subplot(3,1,3)
plot(t,V(:,1),'b',t,V(:,2),'--r',t,V(:,3),':k','Linewidth',2)
legend('v1','v2','v3','Location','northwest')
xlabel('t')
set(gca,'Fontsize',16)


function location = data(vidFrames,filter,scale)
   [height width rgb num_frames] = size(vidFrames);
        location = zeros(num_frames,2);
        for j = 1:num_frames
        X=vidFrames(:,:,:,j);
        Xgray = double(rgb2gray(X));
        X_filtered = Xgray.*filter;
        threshold = X_filtered>scale;
        [Y,X]=find(threshold);
        location(j,1)=mean(X);
        location(j,2)=mean(Y);
        end
 end
```