AMATH 482 Homework 3

Yiqing He

February 24^{th}, 2021

Abstract

This report discusses the illustration of various aspects of the Principal Component Analysis, its practical usefulness, and the effects of noise on the PCA algorithms. I explored the PCA method on 4 different datasets and compare and contrast different cases.

Introduction and Overview

At the beginning of the project, I extracted the mass positions from the video frames that were created from three different cameras. I loaded each video and created loops for them to run. I analyzed 4 tests. The first one is ideal case, in which the entire motion is in the z direction with simple harmonic motion being observed. The second one is noisy case, in which I repeated the ideal case experiment and introduce camera shakes into the video recording. Though this time it is more difficult to extract the simple harmonic motion, the dynamics will still be extracted with the PCA algorithms. The third one is horizontal displacement, in which the mass is released off-center so as to produce motion in the x-y plane as well as the z direction. The last one is horizontal displacement and rotation, in which the mass is released off-center and rotates so as to produce rotation in the x-y plane and motion in the z direction.

Theoretical Background

This project mainly uses the Singular Value Decomposition,

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

where **U** and **V** are unitary matrices and $\mathbf{\Sigma}$ is a diagonal matrix.

From this equation, the values σ_n on the diagonal of $\mathbf{\Sigma}$ are called the singular values of the matrix A. the vectors $\mathbf{u}_n$ which make up the columns of **U** are called the left singular values of **A**. The vectors v_n which make up the columns of **V** are called the right singular vectors of **A**.

The reduced SVD is derived,

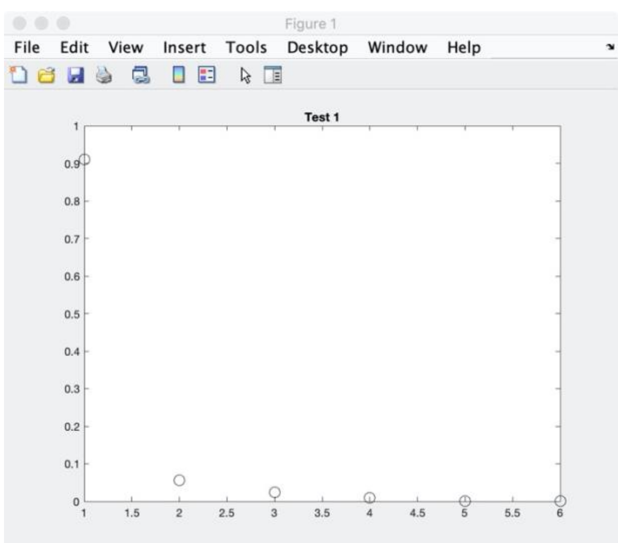
$$\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*.$$

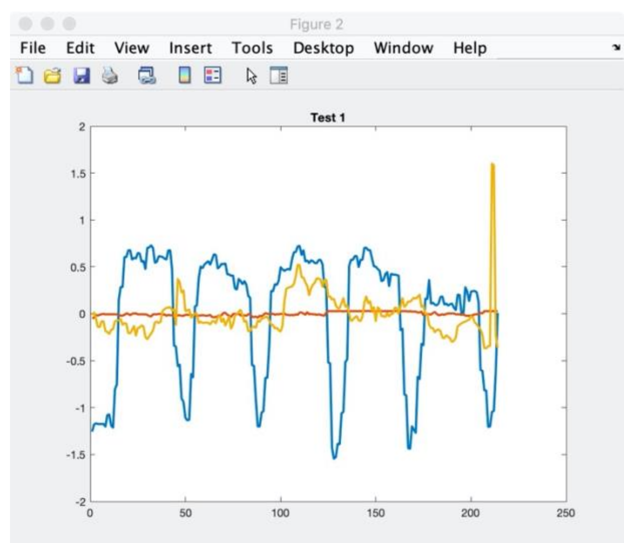
Algorithm Implementation and Development

I used similar techniques for all four cases by applying Singular Value Decomposition. I first defined the filter and then converted the colormap to grayscale. After that, I applied the filter and used it to find the threshold. Thus, I could get the subscripts. After that, I found the svd and extracted singular values from them. Thus, I could then plot the graph and get the results.

Computational Results

Test 1





Test 2

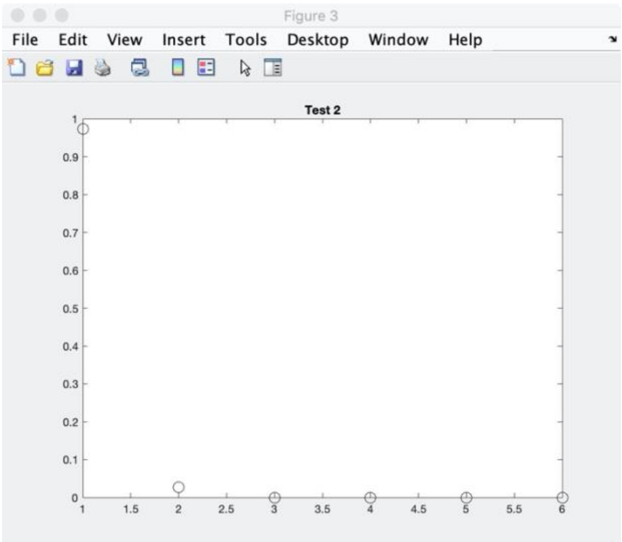


Figure 3
File Edit View Insert Tools Desktop Window Help
Test 2

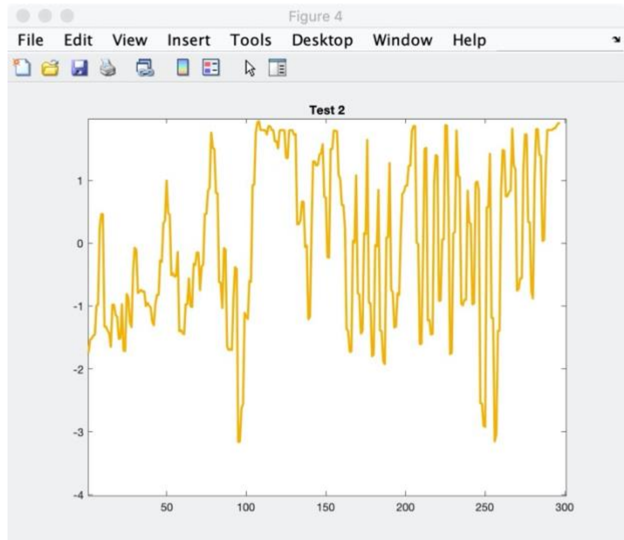


Figure 4
File Edit View Insert Tools Desktop Window Help
Test 2

Test 3

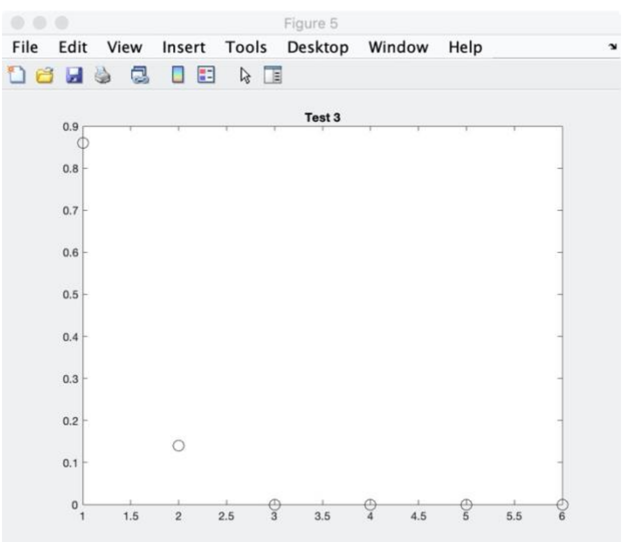


Figure 5
File Edit View Insert Tools Desktop Window Help
Test 3

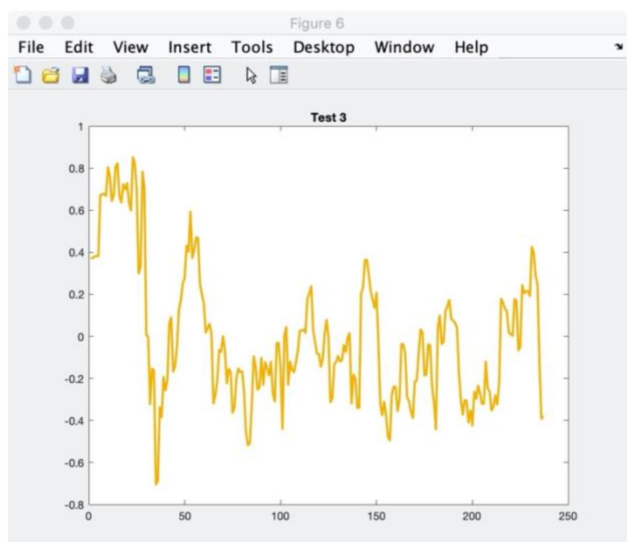
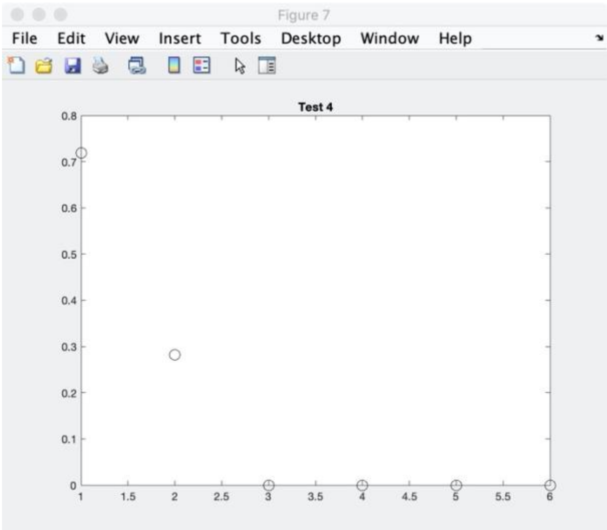
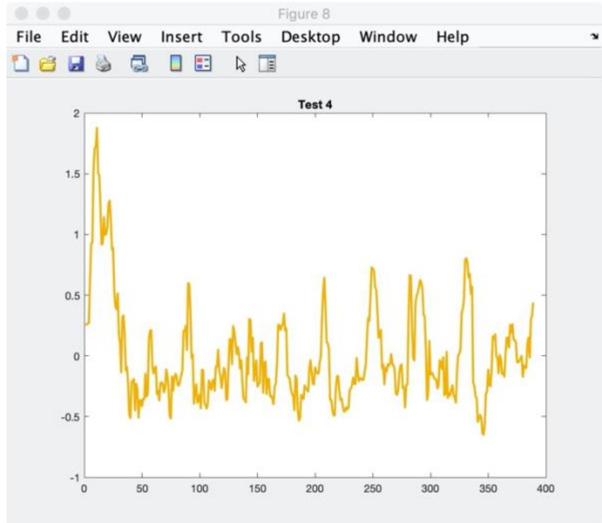


Figure 6
File Edit View Insert Tools Desktop Window Help
Test 3

Test 4





Summary and Conclusions

From the results, I could see that the first case gives the clearest results. The second case is the pretty different from the other three because there is noise. The third one and the fourth one are very similar because they only has one different part, that is, test 4 includes rotation of the plane. The results from all 4 cases are useful and provides valuable information.

Appendix A

sz = size(A) returns a row vector whose elements are the lengths of the corresponding dimensions of A.
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.
Y = double(X) converts the values in X to double precision.
k = find(X) returns a vector containing the linear indices of each nonzero element in array X.
B = repmat(A,r1,...,rN) specifies a list of scalars, r1,..,rN, that describes how copies of A are arranged in each dimension.

Appendix B

```
%% clc;clear all
load('cam1_1.mat')
load('cam2_1.mat')
load('cam3_1.mat')
numFrames1_1 = size(vidFrames1_1,4);
numFrames2_1 = size(vidFrames2_1,4);
numFrames3_1 = size(vidFrames3_1,4);
data1 = [];
for i = 1:numFrames1_1
    filter = zeros(480, 640);
    filter(150:250, 350:450) = 1;
    X = vidFrames1_1(:,:,:,i);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data1 = [data1; mean(X), mean(Y)];
end

data2 = [];
filter = zeros(480, 640);
filter(170:430, 400:500) = 1;
for j = 1:numFrames2_1
    X = vidFrames2_1(:,:,:,j);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data2 = [data2; mean(X), mean(Y)];
end

data3 = [];
filter = zeros(480, 640);
filter(200:350, 235:485) = 1;
for k = 1:numFrames3_1
    X = vidFrames3_1(:,:,:,k);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data3 = [data3; mean(X), mean(Y)];
end

[M,I] = min(data1(1:20,2));
```

```
data1 = data1(I:end,:);
[M,I] = min(data2(1:20,2));
data2 = data2(I:end,:);
[M,I] = min(data3(1:20,2));
data3 = data3(I:end,:);
data1 = data1(1:length(data3),:);
data2 = data2(1:length(data3),:);
alldata = [data1';,data2';data3']
[m,n] = size(alldata);
avg = mean(alldata,2);
all = alldata-repmat(avg,1,n);
[u,s,v]=svd(all/sqrt(n-1));
sig = diag(s);
figure(1)
plot(1:6, sig.^2/sum(sig.^2),'ko', 'Markersize', 10)
title("Test 1")
figure(2)
plot(1:214,all(2,:),1:214,all(4,:),1:214,all(6,:), 'Linewidth',2)
title("Test 1")

%% clc;clear all
load('cam1_2.mat')
load('cam2_2.mat')
load('cam3_2.mat')
numFrames1_2 = size(vidFrames1_2,4);
numFrames2_2 = size(vidFrames2_2,4);
numFrames3_2 = size(vidFrames3_2,4);
data1 = [];
for i = 1:numFrames1_2
    filter = zeros(480, 640);
    filter(150:250, 350:450) = 1;
    X = vidFrames1_2(:,:,:,i);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data1 = [data1; mean(X), mean(Y)];
end

data2 = [];
filter = zeros(480, 640);
filter(170:430, 400:500) = 1;
for j = 1:numFrames2_2
    X = vidFrames2_2(:,:,:,j);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data2 = [data2; mean(X), mean(Y)];
end
```

```
data3 = [];
filter = zeros(480, 640);
filter(200:350, 235:485) = 1;
for k = 1:numFrames3_2
    X = vidFrames3_2(:,:,:,k);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data3 = [data3; mean(X), mean(Y)];
end

[M,I] = min(data1(1:20,2));
data1 = data1(I:end,:);
[M,I] = min(data2(1:20,2));
data2 = data2(I:end,:);
[M,I] = min(data3(1:20,2));
data3 = data3(I:end,:);
data2 = data1(1:length(data1),:);
data3 = data2(1:length(data1),:);
alldata = [data1';,data2';data3']
[m,n] = size(alldata);
avg = mean(alldata,2);
all = alldata-repmat(avg,1,n);
[u,s,v]=svd(all/sqrt(n-1));
sig = diag(s);
figure(3)
plot(1:6, sig.^2/sum(sig.^2),'ko', 'Markersize', 10)
title("Test 2")
figure(4)
plot(1:297,all(2,:),1:297,all(4,:),1:297,all(6,:), 'Linewidth',2)
title("Test 2")
%% clc;clear all
load('cam1_3.mat')
load('cam2_3.mat')
load('cam3_3.mat')
numFrames1_3 = size(vidFrames1_3,4);
numFrames2_3 = size(vidFrames2_3,4);
numFrames3_3 = size(vidFrames3_3,4);
data1 = [];
for i = 1:numFrames1_3
    filter = zeros(480, 640);
    filter(150:250, 350:450) = 1;
    X = vidFrames1_3(:,:,:,i);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data1 = [data1; mean(X), mean(Y)];
```

```
end

data2 = [];
filter = zeros(480, 640);
filter(170:430, 400:500) = 1;
for j = 1:numFrames2_3
    X = vidFrames2_3(:,:,:,j);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data2 = [data2; mean(X), mean(Y)];
end

data3 = [];
filter = zeros(480, 640);
filter(200:350, 235:485) = 1;
for k = 1:numFrames3_3
    X = vidFrames3_3(:,:,:,k);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data3 = [data3; mean(X), mean(Y)];
end

[M,I] = min(data1(1:20,2));
data1 = data1(I:end,:);
[M,I] = min(data2(1:20,2));
data2 = data2(I:end,:);
[M,I] = min(data3(1:20,2));
data3 = data3(I:end,:);
data2 = data1(1:length(data1),:);
data3 = data2(1:length(data1),:);
alldata = [data1';,data2';data3']
[m,n] = size(alldata);
avg = mean(alldata,2);
all = alldata-repmat(avg,1,n);
[u,s,v]=svd(all/sqrt(n-1));
sig = diag(s);
figure(5)
plot(1:6, sig.^2/sum(sig.^2),'ko', 'Markersize', 10)
title("Test 3")
figure(6)
plot(1:237,all(2,:),1:237,all(4,:),1:237,all(6,:), 'Linewidth',2)
title("Test 3")
%% clc;clear all
load('cam1_4.mat')
load('cam2_4.mat')
load('cam3_4.mat')
```

```
numFrames1_4 = size(vidFrames1_4,4);
numFrames2_4 = size(vidFrames2_4,4);
numFrames3_4 = size(vidFrames3_4,4);
data1 = [];
for i = 1:numFrames1_4
    filter = zeros(480, 640);
    filter(150:250, 350:450) = 1;
    X = vidFrames1_4(:,:,:,i);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data1 = [data1; mean(X), mean(Y)];
end

data2 = [];
filter = zeros(480, 640);
filter(170:430, 400:500) = 1;
for j = 1:numFrames2_4
    X = vidFrames2_4(:,:,:,j);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data2 = [data2; mean(X), mean(Y)];
end

data3 = [];
filter = zeros(480, 640);
filter(200:350, 235:485) = 1;
for k = 1:numFrames3_4
    X = vidFrames3_4(:,:,:,k);
    Xg = rgb2gray(X);
    X2 = double(X);
    Xg2 = double(Xg);
    Xf = Xg2.*filter;
    thres = Xf > 10;
    ind = find(thres);
    [Y, X] = ind2sub(size(thres), ind);
    data3 = [data3; mean(X), mean(Y)];
end

[M,I] = min(data1(1:20,2));
data1 = data1(I:end,:);
[M,I] = min(data2(1:20,2));
data2 = data2(I:end,:);
[M,I] = min(data3(1:20,2));
data3 = data3(I:end,:);
data2 = data1(1:length(data1),:);
data3 = data2(1:length(data1),:);
```

```
alldata = [data1';,data2';data3']
[m,n] = size(alldata);
avg = mean(alldata,2);
all = alldata-repmat(avg,1,n);
[u,s,v]=svd(all/sqrt(n-1));
sig = diag(s);
figure(7)
plot(1:6, sig.^2/sum(sig.^2),'ko', 'Markersize', 10)
title("Test 4")
figure(8)
plot(1:389,all(2,:),1:389,all(4,:),1:389,all(6,:), 'Linewidth',2)
title("Test 4")
```